ADAPTIVE CONTROL OF ROBOT MOTION

by

Weiping Li

S.B., Beijing University of Aeronautics and Astronautics (1982)

S.M., Massachusetts Institute of Technology (1986)

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 1990

Signature of Author _____
Department of Aeronautics and Astronautics
May 20, 1990

Certified by _____
Professor Jean-Jacques E. Slotine, Thesis Supervisor
Associate Professor of Mechanical Engineering

Certified by _____
Professor Wallace E. Vander Velde
Professor of Aeronautics and Astronautics

Certified by _____
Doctor J. Kenneth Salisbury
Research Scientist, A. I. Laboratory

Accepted by _____
Professor Harold Y. Wachman
Chairman, Departmental Graduate Committee

ADAPTIVE CONTROL OF ROBOT MOTION

by

Weiping Li

Submitted to the Department of Aeronautics and Astronautics on May 20, 1990 in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Abstract

In many robot manipulation tasks, there is considerable uncertainty in the inertial parameters of the manipulated objects. Such uncertainty can cause significant motion inaccuracy or even control instability for fixed-parameter controllers. Adaptive control has been advocated as a promising approach to maintain good performance in the presence of parameter uncertainty, but the nonlinear and multi-input multi-output nature of robot dynamics makes proper adaptive control design difficult. In this thesis, an effective adaptive controller is developed based on the exploitation of the physical properties of the robot dynamics and the use of a Lyapunov-like analysis.

The basic adaptive robot controller is composed of a control law in the form of feedforward plus PD, and an adaptation law driven by motion tracking error. The adaptive control system is shown to be globally stable and the tracking errors of joint motion are shown to converge to zero. The proposed adaptive controller is experimentally demonstrated on a two-link high speed robot, with excellent performance achieved. A new method, called composite adaptation, is also developed to improve the tracking-error-based adaptive controller by incorporating the torque prediction error into the adaptation.

The adaptive controller, originally developed for free robot motion, is extended to constrained motion, with the global stability and convergence properties retained. This is achieved by modeling the interacting robots and environments involved in constrained motion as an integrated mechanism described by a set of generalized coordinates and then applying the adaptive control method on the dynamics of the integrated mechanism.

Thesis Committee:

Prof. Wallace E. Vander Velde, Chairman
Prof. Jean-Jacques E. Slotine
Dr. J. Kenneth Salisbury

Thesis Supervisor: Professor Jean-Jacques E. Slotine
Title: Associate Professor of Mechanical Engineering

Acknowledgments

I would like to express my deepest appreciation to my thesis supervisor, Professor Jean-Jacques Slotine, for his advice, trust and genuine friendship during the past four years. His unflagging excitement in dynamical systems and unbounded energy in everyday work have been inspirations to me.

I would like to thank Professor Wallace Vander Velde for serving as Chairman of my thesis committee and for his genuine kindness in these years. I am grateful to Dr. Ken Salisbury for serving in my thesis committee and for introducing me into the world of robotics with his fascinating robot hand demonstration. I would also like to thank Professor T.T. Pian for influencing me with his precise thinking and scholastic attitude when I was doing my Master degree under his supervision.

I have benefited greatly from interactions with many friends and colleagues at MIT, both intellectually and socially. I will not forget my friends Yangming Xu and Zhiming Bao for their pleasent companionship in all these years, unforgettable friendship during both good times and bad times, and stimulating discussions about all aspects of life. I would like to express my sincere appreciation to my office-mates Hyun Seok Yang, Gunter Niemeyer, Barbara Hove, Mohamed Khemira, Max Mendel, Zia Vafa for being very helpful and supportive. I want to thank many people around here for their invaluable friendship, particularly, Cheng Wang, Yayan Lu, Jiang Huang, John Keen, Vic Churpoon, Segun Ige, Kuangmang Xu, Guohua Jiang, Zhexiang Li, Alain Curodeau, Feng Zhao, Yongyi Wang, Huosi Sun, Yan Wang, Mohamed Khemira, Yaping Liu, Hongbo Xu, Wenhua Jiang, Wei Ling, Mang Xie. I also want to thank my friends in the fifth floor kitchen of Ashdown House, Alain Curodeau, Hojoon Park, Eric K.Y. Leung, Rajesh Dontula, Albert S.H. Wong, Kian Boon Tay, Yi-Chung Shao, and Sandeep Bahl, who have made my daily cooking such an enjoyable experience. Finally, I would like to thank my family for their understanding and emotional support during these years.

This research is sponsored in part by a grant from the Sloan fund, and in part by a grant from the Doherty Foundation.

Contents

List of Figures

Chapter 1
Introduction

Many manipulation tasks require accurate tracking of prespecified motion trajectories. The well-known computed torque (inverse dynamics) method can be used to control the robot motion in such tasks, provided that an accurate model of the robot system dynamics is available. However, in many practical applications, an accurate model of robot system is not available because there is considerable uncertainty in the inertial parameters of the load or the tool manipulated by the robot hand. Such parameter uncertainty can lead to significant performance degradation or even control instability for robots under computed-torque control. This problem is especially severe for high-performance robots, such as the increasingly popular direct-drive robots.

The objective of this thesis is to *develop an adaptive control method which allows stable, accurate and fast robot motion to be achieved in the presence of parameter uncertainty*. Specifically, we shall develop an adaptive tracking control

method for robot free motion, analytically study its theoretical properties such as stability and convergence, experimentally examine its performance and extend the method to robot constrained motion.

In the first section of this chapter, we provide some background for the work in the thesis. Section 1.2 gives a brief review of the work of other researchers on adaptive robot control. Section 1.3 summarizes the contribution of the thesis and section 1.4 provides an outline of the thesis.

1.1 Background

A discussion of some background topics allows us to gain a better understanding and assessment of the results in the thesis.

1.1.1 Free Motion and Constrained Motion

Robot tasks can be classified by the kind of robot motion involved, either free motion or constrained motion. *Free motion* occurs if there are no constraints on the motion of the robot end-effector. Examples of free motion tasks include pick-and-place of objects, laser cutting and spot welding. *Constrained motion* occurs when there are constraints on robot end-effectors during task execution. Examples of constrained motion tasks include a robot pushing a cart or turning a crank, and multiple robots manipulating a common object. In this thesis, we shall study the adaptive control problem in both types of robot motions.

The issue of *parameter uncertainty* arises in both free motion tasks and constrained motion tasks. In free motion, the parameter uncertainty arises when a robot grasps a load whose inertial parameters are not supplied by the computer database or the human operator (Figure 1.1). Even though the load parameters can

be determined in principle by shaking the object for a while and collecting the input-output data, and then estimating the parameters using a parameter estimator, it is not efficient because it takes time to do the shaking and estimation. It is much more desirable to pick up the load and move it away immediately, which, of course, requires a controller capable of handling parameter uncertainty. In constrained motion tasks, the inertial parameters of the environment, such as the rotational inertia of a crank to be turned by the robot or the mass of a cart to be pushed by a robot, may be unknown to the robot control system. For multi-robot manipulations, the reason for using multiple robots is usually due to the large size and large mass of the manipulated objects. For such objects, the centers of mass and moments of inertia are hard to determine accurately and parameter uncertainty can easily arise.

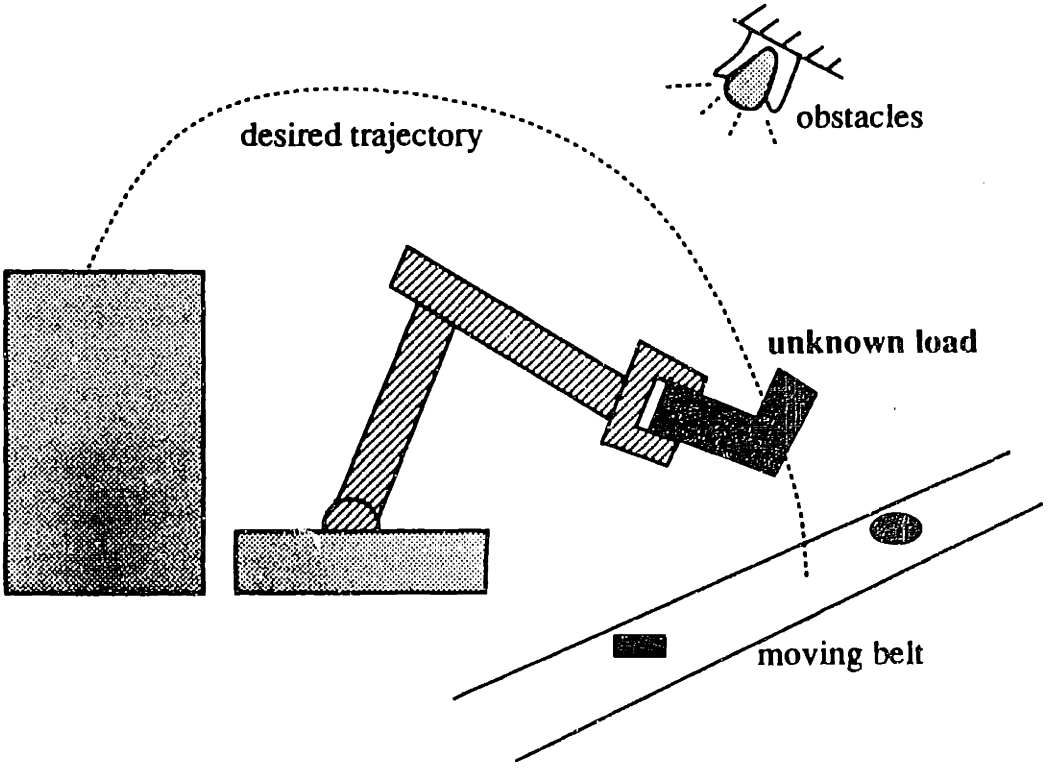


Figure 1-1: Tracking Control in the Presence of Parameter Uncertainty

1.1.2 Adaptive Control and Robust Control

Parameter uncertainty can cause severe problems for model-based controllers like the computed-torque controller [Khosla and Kanade, 1986]. The accuracy of the computed-torque controller is decreased by the presence of parameter uncertainty. In fact, motion accuracy quickly decreases as the speed of robot motion is increased, because the uncertain, and uncompensated, inertial, Coriolis and centripetal forces associated with the parameter uncertainty are dependent on the motion speed in a quadratic fashion. A more serious problem, instability, may occur if the speed of robot motion is high or/and the amount of parameter uncertainty is sufficiently large.

Two approaches have been advocated for dealing with the parameter uncertainty problem. One is robust control, and the other is adaptive control. A *robust controller* is usually a model-based controller including a high gain term for suppressing the effect of parameter uncertainty. An *adaptive controller* is generally composed of a model-based control law with adjustable parameters, and an adaptation law for tuning these parameters based on measured signals (Figure 1.2).

Two features of adaptive control make it more popular than the robust control approach. The first is its learning ability: a robust controller repeats its errors in its operations, while an adaptive controller improves its accuracy by extracting parameter information from system signals.The second is its ease of use. A robust controller usually requires the bounds of parameter uncertainty be known for stable design, while an adaptive controller usually requires little or no *a priori* parameter information. If an unexpected amount of parameter uncertainty is

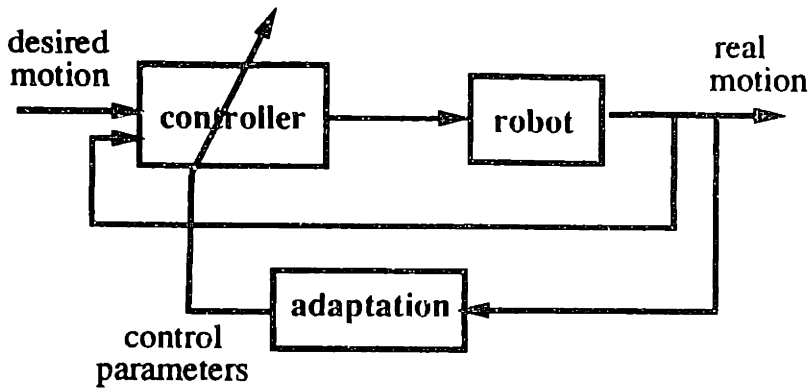


Figure 1-2: Adaptive Robot Control System

present in a task, a robust controller may go unstable, while a well-designed adaptive controller shows larger motion errors only in the initial adaptation process, having little performance degradation in the long run.

1.1.3 Indirect-drive and Direct-drive Robots

Most current industrial robots belong to the category of indirect robots, which use gears or belts to drive the links. The gears and belts not only amplify the effective torque delivered to the links by the motors located at the robot base, but also reduce the nonlinear dynamic effects of load reflected at the joints by a large factor of $1/r^2$ or $1/r$, where r is the gear ratio. Since the gear ratio r is normally a large number (usually between 20-200), the dynamic forces associated with the load are usually dominated by the inertia forces of the motor shafts. This means that the parameter uncertainty associated with the load and links has relatively little impact to robot dynamics and control. Very often, simple controllers like PID controllers can effectively control such robots in tasks of relatively low speeds.

However, there is an increasing interest in direct-drive robots which have high-

torque motors directly coupled to the robot joints. Such robots are appropriate for tasks requiring high speed and/or high accuracy because, unlike indirect drive robots, they have high mechanical stiffness, negligible backlash and low friction. Some direct-drive robots have been put into practical industrial use, such as the Adept-1 and Adept-2 robots which are manufactured by Adept Technologies and commonly used in precise assembly of electronic components. However, the performance of direct-drive robots is very sensitive to load parameter uncertainty because of the lack of gear reduction [Asada and Takeyama, 1983]. Unless the parameter sensitivity problem can be effectively handled, the potential of direct-drive robots in terms of motion speed and motion accuracy cannot be well exploited.

Though the adaptive control approach can be used for both direct-drive and indirect-drive robots in the same way, it is most useful for direct drive robots or robots with small transmission ratios such as cable-driven manipulator arms. The usefulness of adaptive controllers for indirect-drive robots is limited by the relative insensitivity of indirect-drive robots to load uncertainty and the difficulty in modeling the friction and flexibility at the gear or belt transmissions.

1.1.4 Adaptive Control Literature

The existing adaptive control literature mainly deals with linear, time-invariant, single-input single output plants [Astrom and Wittenmark, 1989; Narendra and Annaswamy, 1989]. There are few results for the adaptive control of linear multi-input multi-output plants. There are essentially no results on adaptive control of nonlinear multi-input multi-output plants. Since robot dynamics is nonlinear and multi-input multi-output in nature, the existing adaptive control literature offers no meaningful solution for the problem of adaptive robot control. Fortunately, robot

dynamics have some unique properties, such as the positive definiteness of inertia matrix and the possibility of linear parametrization, which allows us to develop a theoretically elegant adaptive controller in this thesis.

There are mainly two approaches in the adaptive control literature. One approach is the so called model-reference adaptive control (MRAC), and the other is the so called self-tuning (ST) regulators. In MRAC system, a reference model is used to provide the ideal response of the adaptive control system, and the adaptation law adjusts the controller parameters in such a way that the tracking error between the plant output and the model reference output is reduced to zero. Self-tuning control, on the other hand, is simply obtained by coupling a model-based control law and an on-line estimator (which seeks for parameters that fit the plant input and output data). The adaptive controller to be developed in this thesis basically falls into the category of the MRAC approach. But it is different from the standard MRAC systems in that no reference model is actually used.

1.1.5 Desired Properties of Adaptive Robot Controllers

Whether an adaptive robot controller is good is determined by a number of factors. The major factors can be briefly described as follows:

- **stability**: because an adaptive robot control system is nonlinear, we can address the stability issue in a local sense or a global sense.

- **convergence**: There are two types of convergence issues in adaptive robot control, the convergence of the tracking errors and the convergence of the estimated (or adjusted) parameters.

- **accuracy**: Accuracy is one of the most important indicators of controller performance. In order to demonstrate the usefulness of an adaptive controller, one

must show that it leads to better accuracy than non-adaptive controllers.

• **robustness**: Since there are many non-parametric uncertainties which exist in real robot, but are neglected in dynamic modeling and control design, the adaptive controller must be reasonably robust to them, so that it can work in practice, not just in simulations based on the idealized robot model.

• **computational efficiency**: Since robots under adaptive control are supposed to perform high speed motion, the control system should have sufficiently high sampling rate. This implies that the adaptive controller should be computationally reasonable. Specifically, the inversion of the robot inertia matrix (a 6×6 matrix for a 6-DOF robot) should be avoided because its computation with a frequency of several hundred hertz is quite excessive for present-day microprocessors.

Since these requirements are often conflicting, a good compromise should be sought in developing a practically effective adaptive robot controller.

1.2 Review of Adaptive Robot Control Literature

The subject of adaptive robot control has a short history but a rich literature. One of the first studies on adaptive robot control was made by [Dubowsky and DesForges, 1979]. Their method was based on the model-reference adaptive control approach, with a steepest descent adaptation law for adjusting the controller parameters. However, the research work assumed the robot dynamics to be linear, time-invariant, and decoupled, so that the adaptive control methods in adaptive control literature could be applied. Another early study was made by [Tomizuka and Horowitz, 1983]. In their scheme, they assumed the robot inertia to be time-invariant, factored the additional nonlinear terms into a time-invariant quadratic form of the joint velocities and considered the gravity/friction torques to

be known.

In the next few years, a large number of researchers [*e.g.* Takegaki and Arimoto, 1981; Leninger and Wang, 1982; Guo and Koivo 1984; Koditschek, 1984; Lee and Lee, 1984, Landau, 1985; Sundareshan and Koenig; Choi *et al*, 1986] made important contributions by proposing adaptive robot controllers from various perspectives (discrete-time or continuous time, MRAC or self-tuning, centralized or decentralized). Such adaptive controllers were all based on approximations of one kind or another on the robot dynamics model, such as local linearization of robot dynamics, time-invariance of inertia matrix and decoupling of joint motion. As a result, only local stability could be established for such adaptive controllers. Since our study in the thesis does not directly rely on the results in these researchers, they are not reviewed in detail here. Interested readers are referred to the relatively complete surveys by [Landau, 1985; Hsia, 1986].

Around the mid-1980's, some researchers started to design adaptive controllers based on the nonlinear and multi-input multi-output model of robot dynamics. The results belong to one of the two general categories: adaptive high-gain suppression of the nonlinearities and adaptive full-dynamics cancellation. The first type includes the notable contributions by [Balestrino, DeMaria and Sciavicco, 1984; Nicosia and Tomei, 1984; Singh 1985; Lim and Eslami, 1985, Seraji, 1987]. These adaptive controllers are similar in that they employ tracking error feedback with adaptive gains which grow until it is large enough to suppress the non-linearities. One drawback of "high-gain" feedback is the possibility of actuator saturation, whose effects are hard to predict. Another major drawback for many of these schemes is that the control law may chatter, which may lead to the undesirable excitation of unmodeled dynamics.

The second type, the adaptive full-dynamics cancellation schemes, is motivated by the desire of developing adaptive versions of the popular computed-torque method which perfoms well if robot parameters are accurately known. The distinguishing feature of these adaptive controllers is the utilization of the linear parametrization property of the robot dynamics. The first work in this direction was presented by [Craig, *et al*, 1986]. The resulting adaptive scheme contains a computed-torque controller and a MRAC adaptation law for adjusting the controller parameters. The most important contribution of this work lies in its introduction of the linear parametrization property of robot dynamics into adaptive robot control design. The adaptive controller, though having the nice computed-torque control law, has some severe problems. One problem is the need to compute the inversion of the estimated inertia matrix. This is undesirable because not only the invertibility of this matrix is hard to guarantee, but also the computational burden is too excessive for high sampling rate implementation. Another problem is the requirement of the joint acceleration measurement in the implementation of the adaptation law. Because joint accelerations usually cannot be directly measured by sensors, they have to be obtained by the numerical differentiation of the joint tachometer signals. The noise in the obtained accelerations tends to cause inaccuracy and drift in the estimated parameters [Armstrong, 1987].

Another notable adaptive controller in the adaptive computed torque category was developed in [Middleton and Goodwin, 1986]. The controller was obtained by a self-tuning approach, *i.e.*, directly coupling a least-square type parameter estimator with a computed torque controller. The adaptive controller, though possessing the desirable theoretical property of global tracking convergence, also suffers the computational drawback of requiring the inversion of estimated inertia

matrix. One notable contribution in this work is that it avoids the requirement of joint acceleration measurement by filtering the joint torque before using it for parameter estimation.

Adaptive robot control is still an actively studied area, with recent work represented by [Slotine and Li, 1986, 1987a,b,cd, 1988, 1989; Koditschek 1987; Hsu *et al*, 1987; Li and Slotine, 1987, 1988, 1989a,b; Sadegh and Horowitz 1987; Bayard and Wen 1987; Canudas *et al*, 1988; Spong, 1990].

1.3 Contributions

As seen in the above review, the existing work on adaptive control either relies on simplifying approximations on robot dynamics or requires excessive computation. The basic contribution of this thesis is *the development of an adaptive control method for robot motion* which avoids both of these problems. Specifically, the following results have been obtained:

- *an adaptive controller for robot free motion is derived based on physical properties of robot dynamics and a Lyapunov-like analysis.* Compared with the previous adaptive controllers by [Craig, *et al*, 1986; Middleton and Goodwin, 1986]], it is computationally efficient because the controller does not require the inversion of the estimated inertia matrix and can be implemented by a modified version of the recursive Newton-Euler method.

- *The theoretical and practical properties of this adaptive controller are extensively examined.* It is shown to have global stability and global tracking error convergence. The condition for convergence of estimated parameters is found to be a sufficient richness or persistent excitation condition on the desired motion trajectories. The adaptive controller is experimentally implemented on a two-DOF

high speed robot to study its stability, accuracy and robustness, with good results obtained.

• *Alternative techniques of parameter adaptation and parameter estimation are proposed.* A novel technique of using simultaneously the motion tracking errors and torque prediction errors for adaptation, called composite adaptation, is proposed, leading to improved adaptive control performance. Two techniques are also proposed for improving on-line parameter estimation of load parameters, avoiding the possible "gain explosion" problem in the exponentially-forgetting least-square method.

• *The adaptive controllers developed for free motion control are extended to control of constrained motion.* This is achieved by using a so called integrated modeling approach for the constrained robot system. The desirable properties such as global stability and global tracking convergence are retained. The integrated modeling concept is believed to be also useful for non-adaptive constrained motion control.

1.4 Organization of the Thesis

This thesis addresses a diverse variety of topics. The adopted form of presentation is motivated by two considerations: readability and insights. First, the results are presented in such a way that they can be understood by robotics researchers with little or no knowledge of adaptive control literature. Secondly, a bottom-up style of development (from one DOF joint space adaptive control to multiple-DOF control in free and constrained motions), instead of top-down approach, has been used to foster insights into the adaptive robot control systems. This means that we start from adaptive control of a single-link robot, then extend

the results to multi-link robots and to constrained motion control.

Chapter 2 prepares us for the development of the adaptive robot control systems by providing the unique properties of the robot dynamic model and the mathematical tools for later analysis. In section 2.4, an adaptive controller is developed for a simple one-DOF robot (essentially a mass) to highlight the concepts and techniques which will be involved in later design for general robots. The procedure used to derive this adaptive controller is formalized into a so called Lyapunov-like design method to facilitate the search of appropriate adaptive controllers in the next chapters.

Chapter 3 provides the most important results of the thesis, the derivation of an adaptive joint-space controller for robot free motion, and the proof of its tracking convergence. Some simulations with a 2-DOF manipulator are included to illustrate the performance of the adaptive controller. At the end of the chapter, a Cartesian-space version of the adaptive controller is derived. Chapter 4 provides further understanding of the algorithm by studying the conditions for the convergence of the estimated parameters.

Chapter 5 experimentally examines the performance of the adaptive controller on a 2-DOF manipulator. The importance of the experimental implementation lies in the fact that the experimental results reflect the attainable performance of the adaptive controller in practical applications, since the non-parametric uncertainties (such as motor dynamics, link flexibility, *etc*) neglected in the adaptive control design and computer simulation are present in the hardware implementations. Chapter 6 discusses various on-line parameter estimators for t estimating the load parameters. Chapter 7 proposes and analyzes a novel technique, called composite adaptation, for improving the adaptive control performance.

Chapter 8 extends the previous adaptive controller to robot control in constrained motion. This is made possible by using a so-called integrated approach for robot modeling. With such a modeling perspective, one can actually treat the constrained motion control and free motion control in a unified approach and essentially all results from free motion control, adaptive or not, are all applicable to constrained motion control.

Chapter 2
Preliminaries

The development of the adaptive control method in this thesis is based on the unique physical properties of robot dynamic model, and on a mathematical tool called Barbalat's lemma. It is the objective of this chapter to briefly review these materials. Because the next chapter, devoted to the adaptive control of robots with arbitrary number of joints, is mathematically complicated, the adaptive control design for a simple one-link robot is discussed at the end of this chapter as a simple illustration of the concepts and techniques to be used in the next chapter.

2.1 Dynamic Model of Robot Manipulators

Let us briefly derive the dynamic model of a n-joint manipulator. In the following derivation and throughout the thesis, we make the fundamental assumption that *the links and the joints of the robot are rigid.* The robots can have revolute joints, translational joints or a mixture of them.

Given an open-chain manipulator with n joints, its position can be described by the positions of its n joints, *i.e.*, the n joint positions constitute the generalized coordinates of the manipulator dynamics, normally referred to as the joint coordinates. Let the n joint positions $q_1(t)$, $q_2(t)$, ..., $q_n(t)$ be contained in the $n\times 1$ vector $\mathbf{q}$. Then the total kinetic energy of the manipulator can be expressed as

$$T = \frac{1}{2}\dot{\mathbf{q}}^T \mathbf{H}(\mathbf{q})\,\dot{\mathbf{q}}$$

where $\dot{\mathbf{q}}$ is the joint velocity vector and $\mathbf{H}(\mathbf{q})$ the $n\times n$ inertia matrix of the robot. Note that the inertia matrix of the manipulator depends on the joint position $\mathbf{q}$, as is consistent with our physical insights (a stretched arm has a larger inertia than a folded arm). The potential energy G associated with the robot links is a function of the manipulator position $\mathbf{q}$, which can be denoted by

$$G = G(\mathbf{q})$$

The Lagragian of this system is thus

$$L = T - G = \frac{1}{2}\,\dot{\mathbf{q}}^T \mathbf{H}(\mathbf{q})\,\dot{\mathbf{q}} - G(\mathbf{q})$$

Now using Lagrangian equations,

$$\frac{dL}{dt} - \frac{\partial L}{\partial \mathbf{q}} = \boldsymbol{\tau}$$

we can obtain the dynamic equations of the manipulator

$$\mathbf{H}\,\ddot{\mathbf{q}} + \mathbf{b}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau} \qquad (2\text{–}1)$$

where $\boldsymbol{\tau}$ is the joint actuator torque vector, $\mathbf{g}(\mathbf{q})$ is the gravitational torque vector, *i.e.*,

$$\mathbf{g}(\mathbf{q}) = \frac{\partial G}{\partial \mathbf{q}}$$

and the $n\times1$ vector **b** contains the Coriolis and centripetal torques of the robot links, defined by

$$\mathbf{b}(\mathbf{q}, \dot{\mathbf{q}}) = \dot{\mathbf{H}}\dot{\mathbf{q}} - \frac{\partial T}{\partial \mathbf{q}}$$

Now let us examine the vector $\mathbf{b}(\mathbf{q}, \dot{\mathbf{q}})$ more closely. First, we note that each component of **b** is a quadratic function of the joint velocity components $\dot{q}_i$ (i = 1, 2 , ..., n). This can be seen be noting that the ith component of **b** is

$$b_i(\mathbf{q}, \dot{\mathbf{q}}) = \sum_{j=1}^{j=n} \dot{H}_{ij}\dot{q}_j - \frac{1}{2}\dot{\mathbf{q}}^T \frac{\partial \mathbf{H}}{\partial q_i}\dot{\mathbf{q}} = \sum_{j=1}^{j=n}\sum_{k=1}^{k=n} h_{ijk}\dot{q}_j\dot{q}_k \qquad (2\text{–}2)$$

where the coefficiens h_{ijk} are functions of only the joint positions and verify

$$h_{ijk} = \frac{\partial H_{ij}}{\partial q_k} - \frac{1}{2}\frac{\partial H_{jk}}{\partial q_i}$$

Secondly, we note that the quadratic form in (2-2) implies that b_i can be written as the product of a $1\times n$ vector $\mathbf{c}_i(\mathbf{q}, \dot{\mathbf{q}})$ and the velocity vector $\dot{\mathbf{q}}$ *i.e.*, $b_i = \mathbf{c}_i\dot{\mathbf{q}}$. Therefore, the vector $\mathbf{b}(\mathbf{q}, \dot{\mathbf{q}})$ can be written as the product of a matrix **C** and the velocity vector $\dot{\mathbf{q}}$, *i.e.*,

$$\mathbf{b}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} \qquad (2\text{–}3)$$

where **C** is the matrix obtained by stacking up the row vectors $\mathbf{c}_i$.

Thirdly, we note that there are many choices of **C** which can achieve the kind of decomposition in (2-3). For example, a quadratic function $b_1 = \dot{q}_1^2 + 2\dot{q}_1\dot{q}_2 + \dot{q}_2^2$ can correspond to either $\mathbf{c}_1 = [\dot{q}_1, (\dot{q}_1 + 2\dot{q}_2)]$ or $\mathbf{c}_1 = [(\dot{q}_1 + \dot{q}_2), (\dot{q}_1 + \dot{q}_2)]$. A particularly interesting choice of the matrix **C** for

(2-3) is given by

$$C_{ij} = \frac{1}{2}\sum_{k=1}^{n}\frac{\partial H_{ij}}{\partial q_k}\dot{q}_k + \frac{1}{2}\sum_{k=1}^{n}\left(\frac{\partial H_{ik}}{\partial q_j} - \frac{\partial H_{jk}}{\partial q_i}\right)\dot{q}_k$$

$$= \frac{1}{2}\dot{H}_{ij} + \frac{1}{2}\sum_{k=1}^{n}\left(\frac{\partial H_{ik}}{\partial q_j} - \frac{\partial H_{jk}}{\partial q_i}\right)\dot{q}_k \qquad (2\text{–}4)$$

This choice of C satisfies (2-3) because, by reindexing, we can rewrite b in (2-3) as

$$b_i = \frac{1}{2}\sum_{j=1}^{n}\sum_{k=1}^{n}\frac{\partial H_{ij}}{\partial q_k}\dot{q}_j\dot{q}_k + \frac{1}{2}\sum_{k=1}^{n}\sum_{j=1}^{n}\left(\frac{\partial H_{ik}}{\partial q_j} - \frac{\partial H_{jk}}{\partial q_i}\right)\dot{q}_k\dot{q}_j$$

The motivation of this choice is the existence of a nice property associated with it, as will be discussed soon. In this thesis, we will concentrate on the following form of robot dynamic model

$$\mathbf{H(q)}\ddot{\mathbf{q}} + \mathbf{C(q,\dot{q})}\dot{\mathbf{q}} + \mathbf{g(q)} = \boldsymbol{\tau} \qquad (2\text{–}5)$$

where the matrix C is defined by (2-4).

In the above equation, friction forces are neglected. But they can be easily incorporated into the above model, simply as an extra term on the left side of the above equation. Usually friction forces are quite difficult to model accurately. Possible models of the friction are Coulomb friction, viscous friction and stiction, or their combinations. Coulomb friction has the form of $f_{ci} = -\alpha_{ci}sgn(\dot{q})$, where f_{ci} is the Coulomb friction force on the ith joint, α_{ci} is the coefficient of the friction, and sgn[] is the sign function. Though the Coulomb coefficient is, according to classic mechanics theory, dependent on the contact force between the joint and the link, and thus on the robot configuration q and other things, it is usually assumed to be constant in manipulator control for simplicity. Viscous

friction is simply a force linear in terms of joint velocity. Stiction is a force which occurs at zero velocity and disappears at nonzero velocity. In practice, which friction model to use depends on the friction situation for the particular robot. If the friction forces are modeled by Coulumb and/or viscous forms, the adaptive control design follows an identical procedure as will be seen. Therefore, for simplicity of notation, we will assume the friction to be absent in the rest of this chapter.

2.2 Properties of Robot Dynamic Model

Now let us discuss a few important properties of the robot model in (2-5). These properties will be instrumental in the development in the later chapters.

A. The Uniform Positive Definiteness of Inertia Matrix H

The first crucial property associated with the above dynamic model is that *the inertia matrix* $\mathbf{H}(\mathbf{q})$ *is uniformly positive definite*, *i.e.*, for all $\mathbf{q}$ in robot workspace,

$$\mathbf{H}(\mathbf{q}) \geq \alpha \mathbf{I} \tag{2-6}$$

where α is a positive constant and $\mathbf{I}$ the identity matrix with the same dimension as $\mathbf{H}$.

The positive definiteness of $\mathbf{H}$ is an inherent physical property of the mechanical system. It stems from the physical fact that the kinetic energy of a manipulator $(1/2)\dot{\mathbf{q}}^T\mathbf{H}(\mathbf{q})\dot{\mathbf{q}}$ must be larger than zero for *any* joint position $\mathbf{q}$ and *any* joint velocity $\dot{\mathbf{q}}$. The positive definiteness of $\mathbf{H}$ can thus be simply proven using the above physical fact and the following mathematical theorem: a symmetric matrix $\mathbf{M}$ is positive definite if and only if its quadratic form $\mathbf{x}^T\mathbf{M}\mathbf{x}$ is positive for *any* nonzero vector $\mathbf{x}$.

The uniformity of the positive definiteness indicates that the manipulator must have a *non-zero* minimal inertia in its workspace. This part of the property is not so obvious. To show this, we can use the technique of contradiction. Assume that such a nonzero α does not exist. Then, there must be a position in the workspace where the inertia matrix has a zero eigenvalue because the points in the workspace constitute a closed set. Let x_0 be the eigenvector associated with this zero eigenvalue. Then, if the robot arm moves with a unit velocity $x_0/\|x_0\|$, its kinetic energy is zero. Since this is physically impossible, a positive α must exist so that (2-6) is satisfied.

C. The Linear Parametrization of the Manipulator Dynamics

Another property crucial to our later adaptive control design is that *the robot dynamics equation is linearly parametrizable.* The linear parametrizability is proven in [Khosla and Kanade, 1985; An, *et al.*, 1985], and it means that each of the individual terms of the left hand side of (2-5), and therefore the whole robot dynamics, is linear in terms of a suitably selected set of equivalent inertial parameters. That is, we can decompose the dynamic forces into

$$\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} = \mathbf{Y}_H(\mathbf{q},\ddot{\mathbf{q}})\mathbf{a} \qquad (2\text{–}7a)$$

$$\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} = \mathbf{Y}_C(\mathbf{q},\dot{\mathbf{q}})\mathbf{a} \qquad (2\text{–}7b)$$

$$\mathbf{g}(\mathbf{q}) = \mathbf{Y}_G(\mathbf{q})\mathbf{a} \qquad (2\text{–}7c)$$

where **a** is a vector of constant equivalent inertial parameters and $\mathbf{Y}_H$, $\mathbf{Y}_C$ and $\mathbf{Y}_G$ are matrices which are dependent on the robot motion quantities but independent of the paramters in **a**. The property can also be proven by showing that the inertia matrix **H** can be made linear in terms of a set of constant parameters.

Usually 10 equivalent parameters are associated with a rigid-body object. One parameter corresponds to the mass of the object, three to the product of the mass and the coordinates of the center of the mass, and six to the moments of inertia.

D. the Skew-Symmetry of the matrix $(\dot{\mathbf{H}} - 2\mathbf{C})$

Furthermore, earlier researchers [Takegaki and Arimoto, 1981; Koditschek, 1984] pointed out that the matrices H and C in (2-5) are not independent. Actually, with the matrix C defined by (2-4), a third property of the robot dynamics is that the matrix $(\dot{\mathbf{H}} - 2\mathbf{C})$ is skew-symmetric. This can be shown by noting that

$$\dot{H}_{ij} - 2C_{ij} = \sum_{k=1}^{n} \left[\frac{\partial H_{jk}}{\partial q_i} - \frac{\partial H_{ik}}{\partial q_j}\right] \dot{q}_k$$

and therefore,

$$\dot{H}_{ij} - 2C_{ij} = -[\dot{H}_{ji} - 2C_{ji}]$$

or simply written as

$$\dot{\mathbf{H}} - 2\mathbf{C} = -(\dot{\mathbf{H}} - 2\mathbf{C}^T) \tag{2-8}$$

The skew-symmetry property implies that for any $n\times 1$ vector x

$$\mathbf{x}^T (\dot{\mathbf{H}} - 2\mathbf{C})\, \mathbf{x} = 0 \tag{2-9}$$

This can be shown by noting that (2-8) implies that

$$\mathbf{x}^T[\dot{\mathbf{H}} - 2\mathbf{C}]\mathbf{x} = -\mathbf{x}^T[\dot{\mathbf{H}} - 2\mathbf{C}^T]\mathbf{x} = -\mathbf{x}^T[\dot{\mathbf{H}} - 2\mathbf{C}]\mathbf{x}$$

The skew-symmetry property is not fundamental for the development of the globally convergent adaptive controller, but using it makes the resulting adaptive controller computationally-efficient, as can be seen later.

D. a 2-link Manipulator Example

To illustrate the above properties of robot dynamics, let us consider the simple two-link manipulator shown in Figure 2.1.

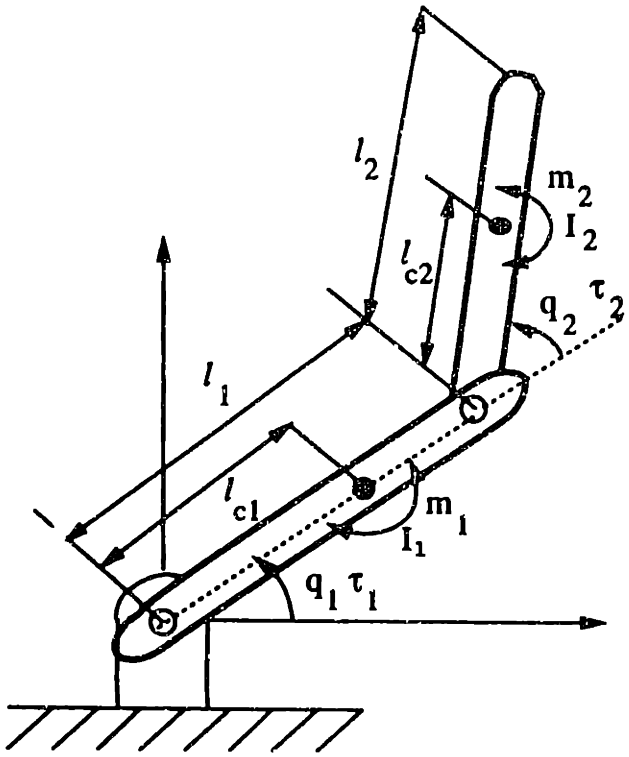


Figure 2-1: A two-link manipulator

With a motor at each joint, the dynamics of this manipulator can be obtained from Lagragian equations as

$$\begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} + \begin{bmatrix} -h\dot{q}_2 & -h\dot{q}_1 - h\dot{q}_2 \\ h\dot{q}_1 & 0 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} + \begin{bmatrix} g_1 \\ g_2 \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \qquad (2\text{–}10)$$

where

$$H_{11} = m_1 l_{c_1}^2 + I_1 + m_2[l_1^2 + l_{c_2}^2 + 2 l_1 l_{c_2} cos q_2] + I_2$$

$$H_{22} = m_2 l_{c_2}^2 + I_2$$

$$H_{12} = H_{21} = m_2 l_1 l_{c_2} cos q_2 + m_2 l_{c_2}^2 + I_2$$

$$h = m_2 l_1 l_{c_2} sin q_2$$

$$g_1 = m_1 l_{c_1} g cos q_1 + m_2 g[l_{c_2} cos(q_1 + q_2) + l_1 cos q_1]$$

$$g_2 = m_2 l_{c_2} g cos(q_1 + q_2)$$

Let us discuss the property of linear parametrization first. Obviously, the robot dynamics is not linear in terms of the physical inertial parameters, as reflected in the terms like $m_1 l_{c1}^2$. However, if we let a be the vector composed of the following elements,

$$a_1 = m_2$$

$$a_2 = m_2 l_{c_2}$$

$$a_3 = I_1 + m_1 l_{c_1}^2$$

$$a_4 = I_2 + m_2 l_{c_2}^2$$

one can show that each term on the left-hand side of (2-10) is linear in terms of the equivalent inertia parameters in a. Specifically

$$H_{11} = a_3 + a_4 + a_1 l_1^2 + 2 a_2 l_1 cos q_2$$

$$H_{22} = a_4$$

$$H_{12} = H_{21} = a_2 l_1 cos q_2 + a_4$$

Note that l_1 and l_2 are kinematic parameters and such parameters are assumed

to be known in adaptive robot control. The above expressions indicate that the inertia torque terms are linear in terms of a. It is simple to do the same for the other terms.

The four equivalent parameters in a also have physical interpretations. The first parameter is the mass of the second link. The second parameter is the product of the mass and the location of the mass center. The third parameter is the inertia of the first link around the first joint and the fourth parameter is the inertia of the second link around the second joint. Note that, in the above example, the number of equivalent parameters is 4 while the number of original physical inertial parameters is 6 (three parameters for each link, *i.e.*, mass, center of mass, moment of inertia). This is actually true in general for robots of arbitrary number of links. This is to say that there are fewer equivalent parameters than physical parameters and therefore the physical parameters cannot be determined from the equivalent parameters, *i.e.*, they are unidentifiable. But this does not cause any problems in control. The reason is that, in computing control inputs, we simply use the equivalent parameters or their estimated values without any concern about what the physical inertia parameters are.

Now let us check the uniform positive definiteness of H. The determinant of H is

$$H_{11}H_{22} - H_{12}^2 = a_3 a_4 + a_1 a_4 l_1^2 - a_2^2 a_1^2 cos^2 q_2$$

Since

$$\begin{aligned} & a_1 a_4 l_1^2 - a_2^2 l_1^2 cos\, q_2 \\ = & m_2 l_2 l_1^2 + m_2^2 l_{c_2}^2 l_1^2 (1 - cos\, q_2) \end{aligned}$$

we have

$$H_{11}H_{22} - H_{12}^2 \geq a_3 a_4$$

i.e., the determinant of the 2×2 inertia matrix is strictly positive. This and the positive nature of H_{11} indicate the uniform positive definiteness of **H**.

Finally, we note that the matrix $\dot{\mathbf{H}} - 2\mathbf{C}$ is given by

$$\dot{\mathbf{H}} - 2\mathbf{C} = \begin{bmatrix} -2\dot{q}_2 & -h\dot{q}_2 \\ -h\dot{q}_2 & 0 \end{bmatrix} + \begin{bmatrix} 2\dot{q}_2 & 2h\dot{q}_1 + h\dot{q}_2 \\ -2h\dot{q}_1 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 2h\dot{q}_1 + h\dot{q}_2 \\ -2h\dot{q}_1 - h\dot{q}_2 & 0 \end{bmatrix}$$

This is obviously a skew-symmetric matrix.

2.3 Mathematical Tools

In this section, we shall describe the important mathematical tool in this thesis, Barbalat's lemma. Its application to adaptive control design is illustrated in the next section.

2.3.1 Barbalat's lemma

Barbalat's lemma is a mathematical result concerning the asymptotic properties of scalar functions. It was introduced into the area of control by Popov in his book on hyperstability [Popov, 1976]. He stated Barbalat's lemma as follows:

Lemma 2.1 (Barbalat's Lemma): *If a real function f(t) is uniformly continuous for all* $t \geq 0$ *and if the limit of the integral* $\int_0^t f(r)\,dr$ *as t tends to infinity exists and is a finite number, then*

$$\lim_{t \to \infty} f(t) \to 0$$

By uniform continuity, we mean that for any positive number $\varepsilon > 0$, there exists $\delta > 0$, such that for all t_1 and t_2 satisfying $|t_2 - t_1| < \delta$,

$$|f(t_2) - f(t_1)| < \varepsilon$$

This lemma can be given geometrical interpretation in terms of the plane with t as horizontal coordinate and f(t) as vertical coordinate, as shown in Figure 2.2. It means that if the curve f(t) is reasonably smooth (implied by the uniform continuity of f) and the total area between the f(t) curve and the positive horizontal axis is finite, the curve must converge to zero. Note that the function f(t) does not have to be of a constant sign.

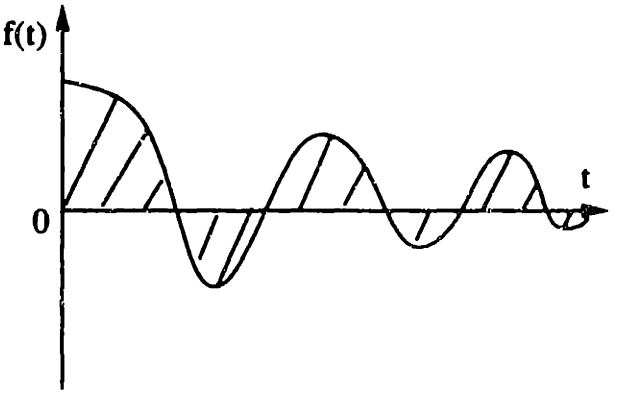


Figure 2-2: Geometrical Interpretation of Barbalat's Lemma

Because of the importance of this lemma in this thesis, let us provide a brief proof for it.

Proof of Barbalat's lemma: This lemma can be proven using the method of

contradiction. The proof is based on the following property of a convergent function: if a function g(t) converges to a finite limit, then the difference $g(t+\eta)-g(t)$, with η being a fixed positive constant must converge to zero as $t\to\infty$. In the following proof, we denote

$$\int_0^t f(r)\,dr = g(t)$$

We shall show that there exists a positive number η such that $g(t+\eta)-g(t)$ does not converge to zero. Note that

$$g(t+\eta)-g(t)=\int_t^{t+\eta} f(r)\,dr$$

Now assume that $f(t)$ does not approach zero as $t\to\infty$. This implies that there exists at least one positive number R_o such that for $\forall T>0$, $\exists\, t>T$, $|f(t)|\geq R_o$. Based on this, one can find an infinite sequence of t's in the following way: Suppose we first choose $T_1 = 1$ and we find t_1 which satisfies $t_1 > T$ and

$$|f(t_1)|\geq R_0$$

By choosing $T_2 = 2 + t_1$, we shall be able to find t_2 such that $t_2 \geq T_2$ and

$$|f(t_2)|\geq R_0$$

Continuing this process, we can find t_3, t_4, ... It is important to note that $t_i\to\infty$ as $i\to\infty$.

Since $f(t)$ is assumed to be *uniformly* continuous, $\exists\,\eta>0$, such that for any t' and t'' satisfying $|t'-t''|<\eta$

$$|f(t')-f(t'')|<\frac{R_o}{2}$$

This implies that for any t within the η-neigborhood of t_i (*i.e.*, such that

$|t - t_i| < \eta$)

$$f(t) > \frac{R_o}{2}$$

Hence, for all t_i ,

$$g(t_i + \eta) - g(t_i - \eta) = \int_{t_i-\eta}^{t_i+\eta} f(t)\,dt \geq \frac{R_o}{2} 2\eta > R_o\eta$$

Since $t_i \rightarrow \infty$ as $t \rightarrow \infty$, this implies that g(t) does not converge to a limit, a contradiction to the assumption in the lemma statement. □

Two facts about the uniform continuity of functions are very useful and summarized into the following lemma.

Lemma 2.2 (Uniform Continuity Lemma): *A function g(t) is uniformly continuous if its derivative $\dot{g}(t)$ is bounded. The product of two uniformly continuous and bounded functions is uniformly continuous.*

Proof: To prove the first part, let us use the middle-value theorem as follows

$$g(t_2) - g(t_1) = \dot{g}(\zeta)(t_2 - t_1)$$

where t_2 and t_1 are arbitrary values and ζ is a value between them. Boundedness of the derivative and the definition of uniform continuity immediately indicate the uniform continuity of the function f(t).

To show that the product of uniformly continuous and bounded functions f(t) and g(t) is also uniformly continuous, let us note

$$f(t_2)g(t_2) - f(t_1)g(t_1) = g(t_2)[f(t_2) - f(t_1)] + f(t_1)[g(t_2) - g(t_1)]$$

Let the bounds of f(t) and g(t) be M_f and M_g. For any $\varepsilon>0$, there must exist a

positive constant δ_f such that for any t_2 and t_1 satisfying $|t_2 - t_1| < \delta_f$

$$|f(t_2) - f(t_1)| < \frac{\varepsilon}{2M_f}$$

since f(t) is uniformly continuous. Similarly, there exists a positive constant δ_g such that $|t_2 - t_1| < \delta_g$ implies that

$$|g(t_2) - g(t_1)| < \frac{\varepsilon}{2M_g}$$

If the smaller of δ_f and δ_g is taken to be δ, then for any t_1 and t_2 satisfying $|t_2 - t_1| < \delta$

$$|f(t_2)g(t_2) - f(t_1)g(t_1)| < |g(t_2)||f(t_2) - f(t_1)| + |f(t_1)||g(t_2 - g(t_1)| < \varepsilon$$

This indicates the uniform continuity of the function $f(t)\,g(t)$. □

The intuitive meaning of the lemma is clear: a function with bounded slope is smooth and the product of two smooth and bounded functions must be smooth.

2.3.2 Useful Corollaries of Barbalat's lemma

It is often inconvenient to use the Barbalat's lemma for control *design*. The following corollary of the Barbalat's is much more convenient to use because it allows us to use a design procedure analogous to that of Lyapunov's direct method. This corollary is sufficient for most problems solvable by Barbalat's lemma, including the ones in this thesis.

Corollary 2.1 (Lyapunov-like Corollary): *If a real function $V(t)$ ($t > 0$) satisfies the following conditions*

$$V(t) \geq 0;$$

$\dot{V}(t) \leq 0$;

$\dot{V}(t)$ *is uniformly continuous;*

then $\dot{V}(t) \rightarrow 0$, as $t \rightarrow 0$.

One easily proves this corollary by regarding $\dot{V}$(t) as the function f(t) in the statement of Barbalat's lemma and noting that the monotonic decrease of the lower bounded function V(t) implies V(t) convergence to a limit smaller than V(0). Since

$$\int_0^t \dot{V}(r)\, dr = V(t) - V(0)$$

the convergence of V(t) implies that the integral tends to a finite limit.

Geometrically, this corollary has a very simple interpretation, as shown in Figure 2.3: the slope of a lower bounded and monotonically decreasing curve must converge to zero if the slope is smooth.

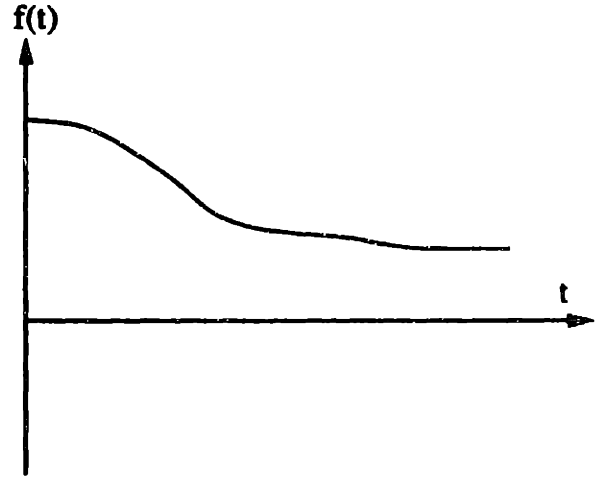


Figure 2-3: A Non-Negative, Decreasing and Smooth Function

The seemingly superfluous condition of uniform continuity is necessary to draw the convergence conclusion, as reflected in the counterexample in Figure 2.4. The function V in Figure 2.4(a) is non-negative and its derivative shown in Figure 2.4(b) is seen to be non-positive, but the slope $\dot{V}$ does not go to zero. This occurs because $\dot{V}$ is not uniformly continuous, *i.e.*, $\dot{V}$ is not smooth.

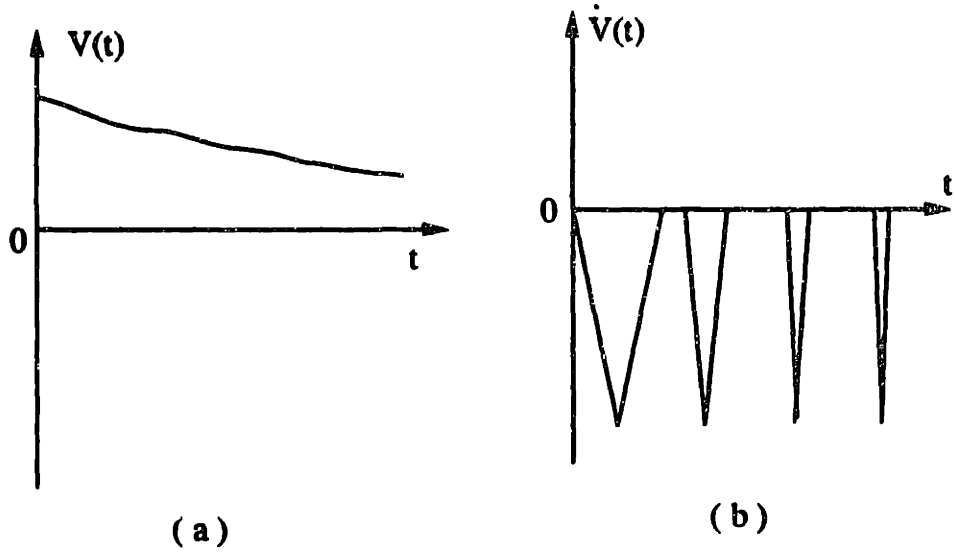


Figure 2-4: A function with not-uniformly-continuous derivative

Sometimes the following corollary of Barbalat's lemma is also useful.

Corollary 2.2 (L^2 Corollary): *If the real function $f(t)$ is L^2, i.e.,*

$$\int_0^\infty f^2(r)\,dr < \infty$$

and f^2 is uniformly continuous, then $f(t) \to 0$ as $t \to \infty$.

This corollary can be easily proven by regarding f^2 as the function f in Barbalat's lemma. Of course, one can state similar results for functions belonging to L^p.

2.4 Adaptive Control of a One-DOF Robot

In this section, we discuss the adaptive control design for a one-DOF robot. This is done to achieve two purposes. First, it illustrates how to use Barbalat's lemma for adaptive control design. Secondly, it illustrates the major concepts involved in the adaptive control design for general robots in the next chapter.

2.4.1 Tracking Control in the Presence of Parameter Uncertainty

The dynamics of the one-link robot shown in Figure 2.5 can be written as follows

$$m\ddot{q} = \tau \tag{2-11}$$

where m is the inertia of the link, q is the joint position and τ the joint torque. Obviously this equation also represents the dynamics of a simple mass.

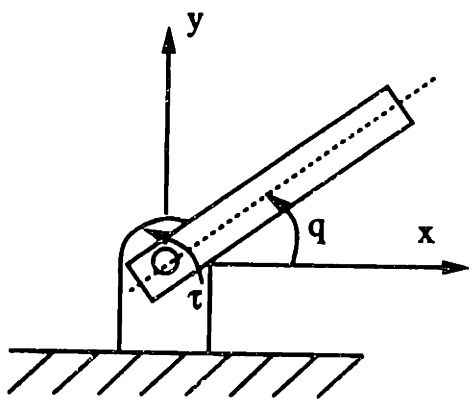


Figure 2-5: A One-Link manipulator

For this one-link robot, the motion tracking problem can be stated as follows:

Given the desired position $q_d(t)$, desired velocity $\dot{q}_d(t)$ and desired acceleration $\ddot{q}_d(t)$, and given the measurement of joint position $q(t)$ and velocity $\dot{q}(t)$, design a controller so that the tracking error $q(t) - q_d(t)$ is small.

If the inertia parameter m is known *a priori*, a computed-torque controller in the form

$$\tau(t) = m\,[\ddot{q}_d - 2\lambda\dot{\tilde{q}} - \lambda^2\tilde{q}] \tag{2-12}$$

where $\tilde{q} = q(t) - q_d(t)$, can be used, with λ being a positive number. The closed-loop dynamics

$$\ddot{\tilde{q}} + 2\lambda\dot{\tilde{q}} + \lambda^2\tilde{q} = 0$$

represents a well-damped dynamics. If initial position and velocity errors are zero, *i.e.*, $\tilde{q}(0) = 0$ and $\dot{\tilde{q}}(0) = 0$,

$$\tilde{q}(t) = \dot{\tilde{q}}(t) = 0 \qquad \forall t \geq 0$$

i.e., the robot has perfect tracking of the desired motion. If the initial tracking errors are not exactly zero, the later tracking errors will not be identically zero, but will be exponentially convergent to zero as $t \to \infty$.

If m is unknown, the computed-torque controller cannot be used in principle. But we can guess a value for m, and plug it into the control law (2-10). Let the guessed parameter be $\hat{m}$. Then the control law is

$$\tau(t) = \hat{m}\,[\ddot{q}_d - 2\lambda\dot{\tilde{q}} - \lambda^2\tilde{q}]$$

and the closed-loop dynamics is

$$m\,\ddot{\tilde{q}} + 2\hat{m}\lambda\dot{\tilde{q}} + \hat{m}\lambda^2\tilde{q} = \tilde{m}\ddot{q}_d$$

where $\tilde{m}$ is the difference between the true inertia m and its guessed value. Provided that $\hat{m}$ is positive, this dynamics is stable. But one notes from the closed-loop dynamics that tracking error *always* exists due to the parameter error $\tilde{m}$. The tracking error $\tilde{q}(t)$ depends on the "external input" $\tilde{m}\ddot{q}_d$, *i.e.*, it depends on the amount of parameter uncertainty and the magnitude of desired acceleration. For robots with more links, one can similarly show that tracking errors are dependent on the amount of parameter uncertainty and the magnitudes of desired velocity and acceleration. The major difference is that instability, a much more severe problem, can possibly occur as a result of parameter error due to the nonlinear nature of multi-joint robot dynamics.

2.4.2 Adaptive Control Design

Now let us design an adaptive controller to control the one-link robot, so that the tracking error due to parameter uncertainty can be gradually eliminated. The basic idea of the adaptive controller is to adjust the parameter $\hat{m}$ on-line instead of sticking to the initially guessed value. The adaptive controller is thus composed of a control law for computing torque for the joint input τ and an adaptation law for adjusting the parameter $\hat{m}$.

To obtain a control law and an adaptation law, let us follow an approach similar to the Lypaunov method of control design, *i.e*, defining a scalar error function and then choosing the controller to make this function decrease. We start by considering the following function

$$V(t) = \frac{1}{2}ms^2 + \frac{1}{2\gamma}\tilde{m}^2 \qquad (2\text{--}13)$$

where γ is a positive constant and s is a combination of the position tracking error and velocity tracking error, defined by

$$s = \dot{\tilde{q}} + \lambda \tilde{q} \tag{2-14}$$

with λ being a positive constant. The quantity s, as a linear combination of position error and velocity error, can be regarded as a compact measure of the tracking accuracy. The function V can be interpreted as a measure of the total error in the adaptive control system, with the first term in (2-14) reflecting the tracking error and the second term reflecting the parameter estimation error.

The variation rate of this "error function" is

$$\dot{V} = m s \dot{s} + \frac{1}{\gamma} \tilde{m} \dot{\tilde{m}}$$

$$= s(\tau - m \ddot{q}_r) + \frac{1}{\gamma} \dot{\hat{m}} \tilde{m}$$

where $\ddot{q}_r = \ddot{q}_d - \lambda \dot{\tilde{q}}$ and the dynamics (2-10) has been used.

This indicates that the variation of the error function depends on the control input τ and the parameter adaptation rate $\dot{\hat{m}}$. If the control law is chosen to be

$$\tau = \hat{m} \ddot{q}_r - k s \tag{2-15}$$

with k being a positive number, then

$$\dot{V} = -k s^2 + \tilde{m} [\ddot{q}_r s + \frac{1}{\gamma} \dot{\hat{m}}]$$

If, furthermore, the parameter variation is chosen to be

$$\dot{\hat{m}} = -\gamma \ddot{q}_r s \tag{2-16}$$

then the variation rate of the error function is

$$\dot{V} = -k s^2 \leq 0$$

This means that $V(t) \leq V(0)$, and, accordingly,

$$s^2(t) \le \frac{2V(0)}{m}$$

$$\tilde{m}^2 \le 2\gamma V(0)$$

i.e., s and $\tilde{m}$ are bounded. The boundedness of s implies the boundedness of the tracking error $\tilde{q}$ and $\dot{\tilde{q}}$, since (2-14) defines a stable first-order filter with s as its input. Therefore, the adaptive controller defined by (2-15) and (2-16) guarantee that the tracking errors $\tilde{q}$ and $\dot{\tilde{q}}$ and the parameter error $\tilde{m}$ are all bounded.

The structure of the adaptive one-link control system is shown in Figure 2.6.

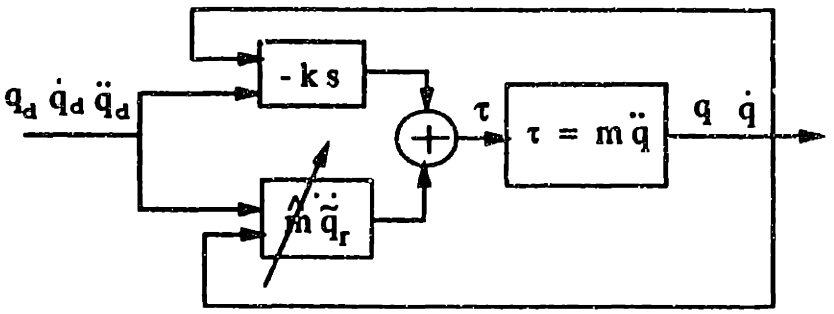


Figure 2-6: Adaptive Control of One Link

2.4.3 Tracking Convergence

We can now further analyze the adaptive control system defined by (2-15), (2-16), (2-14), and (2-11), using Barbalat's lemma or, more conveniently, corollary 2.1 in section 2.2. To start with, note that the function V defined by (2-13) statisfies the first two conditions in corollary 2.1. This implies that the convergence of $\dot{V}$, and accordingly of s, to zero, can be shown by proving the

uniform continuity of $\dot{V}$.

The uniform continuity of $\dot{V}$ can be shown by proving the boundedness of $\ddot{V}$, according to lemma 2.2. Since

$$\ddot{V} = -2s\dot{s}$$

the boundedness of $\ddot{V}$ can be established by the boundedness of $\dot{s}$. To show this boundedness, let us note that the closed-loop dynamics of the tracking error is defined from (2-15) and (2-11) and can be written as

$$m\dot{s} + ks = \tilde{m}\ddot{q}_r \tag{2-17}$$

This defines an interesting stable filter relationship between s (reflecting tracking errors) and parameter error $\tilde{m}$, as shown in Figure 2.7, where p is the Laplace operator. If the desired trajectories q_d, $\dot{q}_d$ and $\ddot{q}_d$ are bounded, the already shown boundedness of $q, \dot{q}$ and $\tilde{m}$ imply the boundedness of $\tilde{m}\ddot{q}_r$, the right-hand side of the above equation. This boundedness and the stability of the filter imply the boundedness of s and $\dot{s}$.

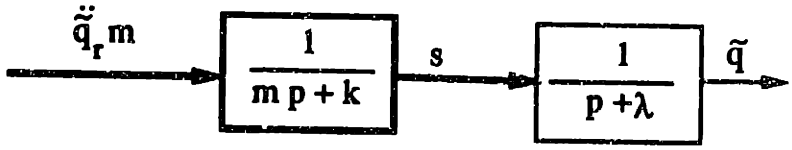


Figure 2-7: Signals in One-Link Adaptive Control System

Thus we have shown that s converges to zero. Since (2-14) defines an exponentially stable filter, one easily shows that the convergence of s implies

$$\tilde{q}(t) \to 0 \quad \text{and} \quad \dot{\tilde{q}} \to 0$$

as $t \to \infty$.

In summary, the adaptive controller defined by (2-15) and (2-16) guarantees the convergence of tracking error to zero without any *a priori* knowledge of the link inertia m.

2.5 A Lyapunov-like Method for Adaptive Control Design

One notes that Barbalat's lemma, though usable for analysis if an adaptive controller has been designed, does not provide much clue as to how to construct an adaptive controller. This indicates the need for a more or less formal procedure of searching for adaptive controllers. Generalizing the procedure used to derive the above adaptive controller, we now obtain a semi-formal method of designing adaptive controllers based on the Corollary 2.1.

Let all the tracking errors (including position and velocity errors) in the adaptive control system be denoted by the vector $\tilde{x}$, parameter estimation error by $\tilde{a}$ and control input by τ. The method is composed of three steps:

step 1: Choosing a positive semi-definite or positive definite error function $V[\tilde{x}, \tilde{a}]$;

step 2: Obtaining a control law τ and an adaptation law $\dot{\hat{a}}$ so that $\dot{V}$ is non-positive;

step 3: showing that $\dot{V} \to 0$ by proving the uniformly continuity of $\dot{V}$.

This method is semi-formal because it does not always guarantee the successful discovery of an adaptive controller. Since our objective is to make the tracking

error $\tilde{\mathbf{x}}$ converge to zero, the third step for asymptotic convergence is meaningful only when $\dot{V}$ is a function of the tracking errors.

This design procedure is quite similar to the design method based on Lyapunov's direct method. But it differs from Lyapunov's method in that the function V here does not have to be *strictly* positive definite, thus possibly not qualifying as a Lyapunov function candidate. In view of the similarity and difference, we call the above design procedure a *Lyapunov-like procedure*. The semi-positive error function V is called a Lyapaunov-like function candidate. If V is indeed shown to be monotonically decreasing along the trajectories of the adaptive control system, it is called a *Lyapunov-like function*. Note that the existence of a Lyapunov function for a system implies the stability of the system, but the existence of a Lyapunov-like function does not necessarily.

It is important to stress that the concepts of Lyapunov-like method, Lyapunov-like function, Lyapunov-like function candidate, and, indeed, the above procedure itself are introduced simply for the convenience of design. In later chapters, we shall see that such a procedure does lead us to useful adaptive robot controllers for both free motion and constrained motion.

Chapter 3
Adaptive Free Motion

In this chapter, we present the most important results in this thesis: *the design of an adaptive controller for robot free motion and the analytical proof of its tracking error convergence.* Insights into the behavior of the adaptive control system are gained through careful examinations of the system structure and extensive simulations on a two-link manipulator. Other issues related to this adaptive free-motion controller, such as parameter convergence, experimental implementation and further extensions, are studied in Chapters 4-5.

The robots to be considered in this chapter may have an arbitrary number of joints (thus the robots can be either non-redundant or redundant) and the joints can be revolute or translational. Though the development of the adaptive controller is much more complicated mathematically than the one-DOF case in section 2.4, the major concepts and results are the same. It follows the same three-step procedure used before. Specifically, we first choose a Lyapunov-like function for the

adaptive control system, then find a control law and an adaptation law such that this error function keeps decreasing, and finally use Barbalat's lemma to conclude asymptotic properties for the adaptive control system.

This chapter is structured as follows. Section 3.1 presents a precise definition of the adaptive control design problem. Section 3.2 derives an adaptive joint-space controller and discusses its basic structure. The global convergence of tracking errors for this adaptive controller is shown in section 3.3 using Barbalat's lemma. Section 3.4 provides the simulation results of a two-link robot under the proposed adaptive controller. Some discussions and remarks concerning the development and application of the adaptive controller are made in section 3.5. Section 3.6 presents a number of interesting variations of the developed adaptive controller. Section 3.7 derives a Cartesian-space version of this adaptive joint-space controller.

3.1 Statement of the Adaptive Control Problem

Intuitively, the objective of the adaptive control design is to provide a controller which can guarantee stable and accurate tracking control in the presence of load/link uncertainty. But the development of the adaptive controller requires a precise statement of this objective, as given below:

Objective of adaptive control design: *given the desired position* $\mathbf{q}_d(t)$, *desired velocity* $\dot{\mathbf{q}}_d(t)$, *desired acceleration* $\ddot{\mathbf{q}}_d(t)$, *and the measurements of the joint position* $\mathbf{q}$ *and velocity* $\dot{\mathbf{q}}$, *and with some or all of the manipulator/load parameters being unknown, design a control law containing variable parameters for the actuator torques* $\boldsymbol{\tau}$, *and an adaptation law for adjusting the variable parameters* $\hat{\mathbf{a}}$, *such that the joint position tracking errors asymptotically converge*

to zero, i.e., $\mathbf{q}(t) - \mathbf{q}_d(t) \rightarrow 0$ as $t \rightarrow \infty$.

In the rest of this thesis, we shall use the word "trajectory" to mean the time history of a motion quantity. The term "desired trajectories" will be used to mean the desired position $\mathbf{q}_d(t)$, the desired velocity $\dot{\mathbf{q}}_d(t)$ and desired acceleration $\ddot{\mathbf{q}}_d(t)$. Furthermore, we shall use the term "estimated parameters" to refer to the adjustable parameters in the control law. Note that this actually represents a subtle abuse of the word "estimated" because the values of these parameters are supplied by a parameter adaptation law, not by a parameter estimator (a parameter estimator intends to reduce data-fitting error, while a parameter adaptation law intends to reduce the tracking error).

The following comments can help clarify the above statement of the design objective:

- *the desired trajectories*, including $\mathbf{q}_d$, $\dot{\mathbf{q}}_d$ and $\ddot{\mathbf{q}}_d$, are assumed to be available from the robot trajectory planning system.

- *the measurements of joint position* $\mathbf{q}$ *and velocity* $\dot{\mathbf{q}}$ are assumed in this control design statement. This is not a restrictive assumption in view of the fact that most industrial robots are equipped with encoders for measuring joint position and tachometers for joint velocities. In the absence of tachometers, one can numerically differentiate the encoder signals to obtain joint velocity if the encoders have reasonably high number of bits.

- *the joint acceleration* $\ddot{\mathbf{q}}$ is not assumed in the control design. The reason is that joint accelerations normally cannot be measured directly and, unlike encoder signals, noisy tachometer signals usually do not allow numerical differentiation.

- *Non-parametric uncertainties*, such as link and joint flexibility, motor

dynamics and measurement noise are assumed absent in the control design. This assumption allows the possibility of asymptotic convergence to zero of the joint tracking errors.

3.2 Derivation of the Adaptive Controller

In this section, we derive an adaptive controller which achieves the specified objective. Since the derivation is essentially the same whether some or all the parameters are unknown, we shall consider the case of all parameters being unknown.

3.2.1 Definition of Some Quantities

Before deriving the adaptive controller, we define some useful quantities so as to make the derivation of the adaptive control law easy to follow.

The position tracking error $\tilde{\mathbf{q}}$ is defined to be the difference between the manipulator joint position $\mathbf{q}$ and its desired value $\mathbf{q}_d$, *i.e.*,

$$\tilde{\mathbf{q}}(t) = \mathbf{q}(t) - \mathbf{q}_d(t)$$

and the velocity tracking error $\dot{\tilde{\mathbf{q}}}$ is its derivative. An important auxiliary signal $\dot{\mathbf{q}}_r$, called "reference velocity", is formed by modifying the desired velocity $\dot{\mathbf{q}}_d$ using the position error $\tilde{\mathbf{q}}$,

$$\dot{\mathbf{q}}_r = \dot{\mathbf{q}}_d - \Lambda\tilde{\mathbf{q}} \tag{3-1}$$

with Λ being a positive definite matrix. Intuitively, the second term on the right-hand side represents a sort of feedback, which intends to increase $\dot{\mathbf{q}}_r$ if the actual trajectory $\mathbf{q}$ lags behind $\mathbf{q}_d$ and vice versa. Correspondingly, the "reference acceleration" is

$$\ddot{\mathbf{q}}_r(t) = \ddot{\mathbf{q}}_d(t) - \Lambda \dot{\tilde{\mathbf{q}}}(t) \tag{3-2}$$

It is important to remark that $\ddot{\mathbf{q}}_r$ depends on the desired trajectories $\ddot{\mathbf{q}}_d$ and $\dot{\mathbf{q}}_d$ and joint measurements $\mathbf{q}$ and $\dot{\mathbf{q}}$, but not on the joint acceleration $\ddot{\mathbf{q}}$. We also point out that, though Λ is often chosen to be a diagonal matrix, it can be a full matrix.

We also define a quantity $\mathbf{s}$ to represent the error between the joint velocity $\dot{\mathbf{q}}$ and the reference velocity $\dot{\mathbf{q}}_r$, *i.e.*,

$$\mathbf{s}(t) = \dot{\mathbf{q}}(t) - \dot{\mathbf{q}}_r(t) \tag{3-3}$$

Let $\mathbf{a}$ be a constant m-dimensional vector containing the unknown elements in the suitably selected set of equivalent dynamic parameters. Then, according to the linear parametrization property of the robot dynamics, the matrices $\mathbf{H}$, $\mathbf{C}$ and $\mathbf{g}$ are all linear in terms of the elements in $\mathbf{a}$. Let $\hat{\mathbf{a}}(t)$ be the estimated value of $\mathbf{a}$ in our adaptive system. Then the matrices $\hat{\mathbf{H}}$, $\hat{\mathbf{C}}$, and $\hat{\mathbf{g}}$, which are obtained from the matrices $\mathbf{H}$, $\mathbf{C}$ and $\mathbf{g}$ by substituting the estimated $\hat{\mathbf{a}}$ for the actual $\mathbf{a}$, must also be linear in terms of the elements of $\hat{\mathbf{a}}$. Let $\tilde{\mathbf{H}} = \hat{\mathbf{H}} - \mathbf{H}$, $\tilde{\mathbf{C}} = \hat{\mathbf{C}} - \mathbf{C}$, $\tilde{\mathbf{g}} = \hat{\mathbf{g}} - \mathbf{g}$. This implies that the following relation exists

$$\tilde{\mathbf{H}}\ddot{\mathbf{q}}_r + \tilde{\mathbf{C}}\dot{\mathbf{q}}_r + \tilde{\mathbf{g}} = \mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}_r, \dot{\mathbf{q}}_r)\,\tilde{\mathbf{a}} \tag{3-4}$$

where $\tilde{\mathbf{a}} = \hat{\mathbf{a}} - \mathbf{a}$ is the parameter estimation error vector, $\mathbf{Y}$ is a $n \times m$ matrix function which is dependent on the desired trajectory, measured position and measured velocity, but independent of the joint acceleration and the inertial parameters.

In the rest of this chapter, we shall often infer properties of the position tracking error $\tilde{\mathbf{q}}$ and velocity tracking error $\dot{\tilde{\mathbf{q}}}$ from the properties of $\mathbf{s}$. Now let us briefly discuss such relations between $\mathbf{s}$ and $\tilde{\mathbf{q}}$ and $\dot{\tilde{\mathbf{q}}}$. In doing so, we note that, with $\dot{\mathbf{q}}_r$ defined by (3-1), $\mathbf{s}$ is related to $\tilde{\mathbf{q}}$ by

$$s = \dot{\tilde{q}} + \Lambda \tilde{q} \tag{3-5}$$

It is seen that s is actually the linear combination of the position tracking error $\tilde{q}$ and velocity tracking error $\dot{\tilde{q}}$. Alternatively, this means that the tracking error $\tilde{q}$ can be interpreted as the filtered version of s through a first order filter (shown in Figure 3-1). The following lemma summarizes the relations between s, $\tilde{q}$, and $\dot{\tilde{q}}$.

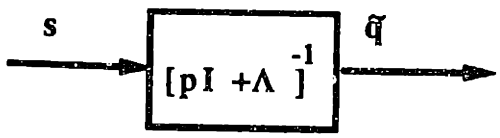


Figure 3-1: Filter Relation Between s and $\tilde{q}$

Lemma 3.1 (filter lemma): *With* s *and* $\tilde{q}$ *related by* $s = \dot{\tilde{q}} + \Lambda \tilde{q}$, *the boundedness of* s *implies the boundedness of* $\tilde{q}$ *and* $\dot{\tilde{q}}$, *the asymptotic convergence of* s *to zero guarantees that of* $\tilde{q}$ *and* $\dot{\tilde{q}}$, *and the exponential convergence of* s *implies that of* $\tilde{q}$ *and* $\dot{\tilde{q}}$.

In the case of $n = 1$, the matrix Λ is a positive scalar which may be simply rewritten as λ. The filter is $1/(p + \lambda)$ and the above relations are intuitive clear. Lemma 3.1 simply indicates that the relations for the scalar case can be extended to the general case of Λ being a full positive definite matrix.

Proof: The basic idea of the proof is to note that equation (3-5) can be rewritten

$$\dot{\tilde{q}} = -\Lambda \tilde{q} + s \tag{3-6}$$

Its solution $\tilde{q}$ is

$$\tilde{q}(t) = s(0)e^{-\Lambda t} + \int_0^t e^{-\Lambda(t-r)}\, s(r)\, dr \qquad (3\text{–}7)$$

which is composed of an exponentially decaying term due to the initial condition $\tilde{q}(0)$, and a convolution term due to the "input" s. Using relation (3-7), and the definitions of boundedness, asymptotic convergence and exponential convergence, one can straightforwardly show such relations between $\tilde{q}$ and s. The corresponding properties of $\dot{\tilde{q}}$ can then be inferred from (3-6). □

The boundedness relation between the signal vectors s, $\tilde{q}$ and $\dot{\tilde{q}}$ is particularly interesting and worth some discussion. We now show that the boundedness of s implies the boundedness of $\tilde{q}$ and $\dot{\tilde{q}}$. Assume that s is known to be bounded by a positive number ε, *i.e*

$$\|s\| \le \varepsilon$$

Then, after the exponentially decaying term in (3-7) (whose effects disappear quickly) is ignored, the position tracking error satisfies

$$\|\tilde{q}(t)\| \le \int_0^t \| e^{-\Lambda(t-r)} \| \, \|s(r)\| \, dr$$

$$\le \int_0^t e^{-\lambda_1(t-r)}\, \varepsilon \, dr = \frac{\varepsilon}{\lambda_1}(1 - e^{-\lambda_1 t})$$

where λ_1 is the smallest eigenvalue of Λ. Thus,

$$\|\tilde{q}(t)\| \le \phi$$

where

$$\phi = \frac{\varepsilon}{\lambda_1}$$

Therefore, for a tracking task, if we want the position tracking error to be bounded by a prespecified bound ϕ, it can be translated as a bound $\lambda_1 \phi$ on s with

λ_1 specified in the A matrix. From (3-6), we can conclude that the velocity tracking error is bounded by a related number, because

$$\|\dot{\tilde{q}}\| \le \|s\| + \|\Lambda \tilde{q}\| \le \varepsilon + \lambda_1 \frac{\varepsilon}{\lambda_1} = 2\varepsilon$$

Since the magnitude of s directly determines the magnitudes of the position and velocity tracking errors, s *can be regarded as an alternative representation of tracking accuracy.* This is similar to the representation in sliding mode control [Slotine, 1984].

3.2.2 The Choice of the Lyapunov-like Function Candidate

The first step in deriving the adaptive robot controller involves choosing a Lypaunov-like functic ı V. However, the choice of an appropriate Lyapunov-like function is usually difficult because there is no generally applicable way of generating such a function. The choice is particularly difficult for nonlinear multi-input systems like the robot dynamics. In our design problem, physical insights and trial-and-errors have played a critical role in the successful formulation of an elegant Lyapunov-like function.

The following Lyapunov-like function will be considered

$$V(t) = \frac{1}{2}[s^T H s + \tilde{a}^T P_0^{-1} \tilde{a}] \qquad (3\text{–}8)$$

where P_0 is assumed to be a constant positive definite matrix. Intuitively, this function is an "error function", with the first term reflecting the error in tracking motion and the second term reflecting the error in parameter estimation.

Some interesting features are noticeable in this function. First, the whole function V has the appearance of quadratic forms. However, unlike the one-link

case in (2-13), it is not really a quadratic form because the matrix $H(q)$ is not a constant matrix. H usually contains a lot of sinusoidal or cosine functions of the joint angles. The first term, with the position-dependent inertia matrix in the middle, has been obtained from the clue of kinetic energy, and the second term motivated by the standard practice of using quadratic forms in adaptive control design. The unique form of the first term allows us to use the inherent Lagrangian structure of the robot dynamics and leads to nice properties of the resulting adaptive controller, as will be seen later.

It is seen that the above function is not a positive definite function for the adaptive control system because nonzero position error $\tilde{q}$ and velocity error $\dot{\tilde{q}}$ which satisfy

$$s = \dot{\tilde{q}} + \Lambda \tilde{q} = 0$$

can lead to $V = 0$. We stress that it is legitimate to use this semi-positive definite function instead of strictly positive definite functions because Barbalat's lemma, instead of Lyapunov theorems, will be used in our analysis. It is also interesting to note that the position tracking error $\tilde{q}$ and velocity tracking error $\dot{\tilde{q}}$, which are the quantities of real concern, do not appear explicitly in the Lyapunov-like function candidate. Instead, the function is quadratic in terms of s, the compact measure of the tracking error. Because of lemma 3.1, the tracking errors $\tilde{q}$ and $\dot{\tilde{q}}$ will converge to zero if s can be shown to converge to zero.

3.2.3 Obtaining the Control and Adaptation Laws

Now we come to the second step of the adaptive controller derivation, *i.e.*, choose a control law and an adaptation law such that $\dot{V} \leq 0$. The differentiation of the error function V is

$$\dot{V} = \mathbf{s}^T \mathbf{H} \dot{\mathbf{s}} + (1/2)\mathbf{s}^T \dot{\mathbf{H}} \mathbf{s} + \tilde{\mathbf{a}}^T \mathbf{P}_0^{-1} \dot{\tilde{\mathbf{a}}} \tag{3-9}$$

Noting that

$$\mathbf{H}\dot{\mathbf{s}} = \mathbf{H}\ddot{\mathbf{q}} - \mathbf{H}\ddot{\mathbf{q}}_r \tag{3-10}$$

and using the manipulator dynamics, we obtain

$$\dot{V} = \mathbf{s}^T [\, \boldsymbol{\tau} - \mathbf{C}\dot{\mathbf{q}} - \mathbf{g} - \mathbf{H}\ddot{\mathbf{q}}_r] + (1/2)\mathbf{s}^T \dot{\mathbf{H}} \mathbf{s} + \tilde{\mathbf{a}}^T \mathbf{P}_0^{-1} \dot{\tilde{\mathbf{a}}}$$

In order to use the skew-symmetry property (2-8) of the robot dynamics, let us add the zero term ($\mathbf{s}^T\mathbf{C}\mathbf{s} - \mathbf{s}^T\mathbf{C}\mathbf{s}$) to the right-hand side of the above

$$\dot{V} = \mathbf{s}^T [\boldsymbol{\tau} - \mathbf{C}\dot{\mathbf{q}} - \mathbf{g} - \mathbf{H}\ddot{\mathbf{q}}_r + \mathbf{C}\mathbf{s}] + (1/2)\mathbf{s}^T [\dot{\mathbf{H}} - 2\mathbf{C}]\mathbf{s} + \tilde{\mathbf{a}}^T \mathbf{P}_0^{-1} \dot{\tilde{\mathbf{a}}}$$

The skew-symmetry property and the constant nature of a lead to

$$\dot{V} = \mathbf{s}^T [\boldsymbol{\tau} - \mathbf{C}\dot{\mathbf{q}}_r - \mathbf{g} - \mathbf{H}\ddot{\mathbf{q}}_r] + \tilde{\mathbf{a}}^T \mathbf{P}_0^{-1} \dot{\hat{\mathbf{a}}}$$

where the term associated with $\dot{\mathbf{H}}$ has been eliminated. Based on this expression, let us take a control law in the following form,

$$\boldsymbol{\tau} = \hat{\mathbf{H}}\ddot{\mathbf{q}}_r + \hat{\mathbf{C}}\dot{\mathbf{q}}_r + \hat{\mathbf{g}} - \mathbf{K}_d\mathbf{s} \tag{3-11}$$

where $\mathbf{K}_d$ is a uniformly positive definite and uniformly bounded matrix and the estimated parameters $\hat{\mathbf{a}}$ involved in $\hat{\mathbf{H}}$, $\hat{\mathbf{C}}$, and $\hat{\mathbf{g}}$ are to be generated by the adaptation law. Substitution of (3-11) into $\dot{V}$ leads to

$$\dot{\mathbf{V}} = \mathbf{s}^T(\tilde{\mathbf{H}}\ddot{\mathbf{q}}_r + \tilde{\mathbf{C}}\dot{\mathbf{q}}_r + \tilde{\mathbf{g}} - \mathbf{K}_d\mathbf{s}) + \tilde{\mathbf{a}}^T \mathbf{P}_0^{-1} \dot{\hat{\mathbf{a}}}$$

$$= -\mathbf{s}^T \mathbf{K}_d \mathbf{s} + \tilde{\mathbf{a}}^T [\mathbf{P}_0^{-1} \dot{\hat{\mathbf{a}}} + \mathbf{Y}^T \mathbf{s}]$$

where $\mathbf{Y}$ has been defined in (3-4).

In order to get rid of the second term on the right-hand side of the above equation, the following adaptation law is chosen

$$\dot{\hat{\mathbf{a}}} = -\mathbf{P}_0 \mathbf{Y}^T(\mathbf{q}, \dot{\mathbf{q}}, \dot{\mathbf{q}}_r, \ddot{\mathbf{q}}_r)\mathbf{s} \tag{3-12}$$

This leads to

$$\dot{V}(t) = -\mathbf{s}^T \mathbf{K}_d \mathbf{s} \leq 0 \tag{3-13}$$

Note that (3-13) implies that V will decrease as long as s is not zero. Intuitively, this and the lower boundedness of V implies that s should tend to zero. However, the rigorous proof of this convergence requires the uniform continuity of $\dot{V}$, as indicated in Barbalat's lemma. The detailed convergence proof is provided in the next section.

In the above design, $\mathbf{K}_d$ can be chosen to be any positive definite constant matrix. Time-varying $\mathbf{K}_d$ may also be used, but it must be uniformly positive definite (so that $\dot{V}$ is smaller than or equal to 0) and uniformly upper bounded (required in the tracking convergence proof given later). This implies that it has to be lower bounded and upper bounded by constant positive definite matrices, *i.e.*,

$$k_0 \mathbf{I} \leq \mathbf{K}_d(t) \leq k_1 \mathbf{I}$$

with k_0 and k_1 being positive constants. In practice, $\mathbf{K}_d$ is often chosen to be constant or even diagonal for simplicity. But the possibility of $\mathbf{K}_d$ being time-varying will sometimes be of theoretical interest.

3.2.4 Closed-loop Dynamics and Global Stability

The closed-loop dynamics of the adaptive control system is of order $3n$, and given by the n 2nd-order equations of plant dynamics (2-5) and n first-order equations of parameter adaptation (3-12). By noting that the control law (3-11) can be rewritten as

$$\tau = \mathbf{Y}\hat{\mathbf{a}} - \mathbf{K}_d \mathbf{s} \tag{3-14}$$

and then substituting it into (2-5), we obtain

$$\mathbf{H}\dot{\mathbf{s}} + (\mathbf{K}_d + \mathbf{C})\mathbf{s} = \mathbf{Y}\tilde{\mathbf{a}} \tag{3-15}$$

This is a very convenient form which will be frequently used later. The complete dynamics of the adaptive control system are thus defined by (3-15), (3-5) and (3-12).

For the *nonlinear* adaptive control system, with states being s, $\tilde{\mathbf{q}}$ and $\tilde{\mathbf{a}}$ (or equivalently $\dot{\tilde{\mathbf{q}}}$, $\tilde{\mathbf{q}}$ and $\tilde{\mathbf{a}}$), we can discuss its stability in local and global sense. First, let us discuss the local stability. The definition of local stability can be found in a number of texts [see *e.g.*, Slotine and Li, 1990]. A system is called stable (or locally stable in the sense of Lyapunov) if its states can remain arbitrarily close to the origin by starting sufficiently close to the origin.

We can now show that this adaptive robot controller locally stable. Expression (3-13) implies that V(t) is non-increasing and

$$V(t) \le V(0)$$

Due to the definition of V(t) in (3-8), the uniform positive definiteness of H, and positive definiteness of $\mathbf{P}_0$, the above equation implies

$$\alpha \|\mathbf{s}\|^2 + \beta \|\tilde{\mathbf{a}}\|^2 \le V(t) \le V(0)$$

where α is given in (2-6) and β is the smallest eigenvalue of $\mathbf{P}_0^{-1}$. Therefore the tracking error measure s and and the parameter estimation error $\tilde{\mathbf{a}}$ are uniformly bounded, *i.e.*,

$$\|\mathbf{s}\| < \frac{V(0)}{\alpha} \qquad \|\tilde{\mathbf{a}}\| < \frac{V(0)}{\beta}$$

Due to lemma 3.1, the position error $\tilde{q}$ and velocity error $\dot{\tilde{q}}$ are also bounded.

One notes that the bounds of the states are dependent on s(0), the initial error in motion, and $\tilde{a}(0)$, the initial error in parameter estimation. Thus, the states $\tilde{q}(t)$, $\dot{\tilde{q}}(t)$, and $\tilde{a}(t)$ of the adaptive control system can be made arbitrarily small by starting from sufficiently small initial values. This indicates the local stability of the adaptive controller.

Now let us discuss the global stability. In the nonlinear control literature, only the definition of global asymptotic stability is given. There exists no formal definition of the concept of global stability. In our study here, we use the concept of global stability to mean that the kind of well-behavedness characterized by local stability actually extends to arbitrary initial values of $\tilde{q}$, $\dot{\tilde{q}}$, and $\tilde{a}$. In view of the fact the previous discussion is not restricted to small initial conditions, we conclude that our adaptive controller is *globally stable*.

3.2.5 The Structure of the Adaptive Controller

The structure of the adaptive control system is shown in Figure 3-2. The controller consists of two parts. The first part, denoted by F.F. and given by (3-11), is a special form of full-dynamics *feedforward* compensation, with three terms corresponding to inertial, centripetal and Coriolis, and gravitational torques. This part, based on the estimated parameters, attempts to provide the joint torques necessary to make the desired motions. From the forms of $\dot{q}_r$ and $\ddot{q}_r$, as given in (3-1) and (3-2), it is seen that this seemingly feedforward part also contains additional feedback actions. The second part actually represents PD feedback, since

$$-K_d s = -K_d \dot{\tilde{q}} - K_d \Lambda \tilde{q} \qquad (3\text{-}16)$$

Intuitively, it intends to regulate the joint trajectories about the desired trajectories, given the incomplete dynamics compensation provided by the first part (incomplete until the parameters converge).

The required inputs to the controller are the desired joint position q_d, velocity $\dot{q}_d$, and acceleration $\ddot{q}_d$. The required measurements are the joint position q and velocity $\dot{q}$.

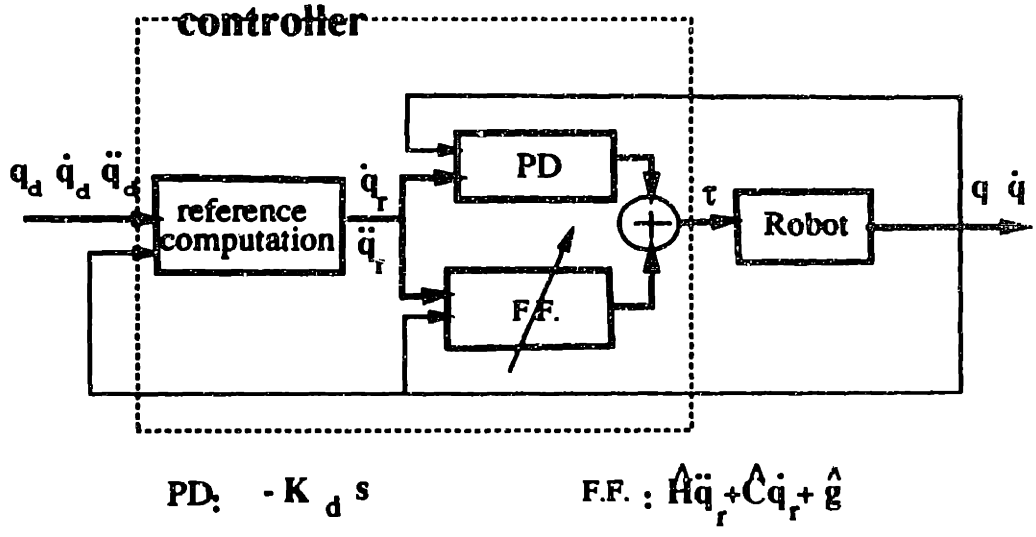


Figure 3-2: Structure of the Adaptive Robot Control System

The structure of the adaptation law is sketched in Figure 3-3 The parameter adaptation is driven by the tracking error s in a direction determined by the matrix Y which reflects the system dynamics . As long as s is not zero, the parameters adaptation will continue. If the tracking errors become zero, the parameter adaptation stops. This indicates that the parameter adaptation is not directly concerned about finding the true parameters. What the adaptation law is concerned about is the convergence of the tracking error. However, this does not imply that

the parameters from the adaptation law have no connection to true parameters, since the tracking errors are related to parameter errors. A more detailed discussion of this point is provided in chapter 4.

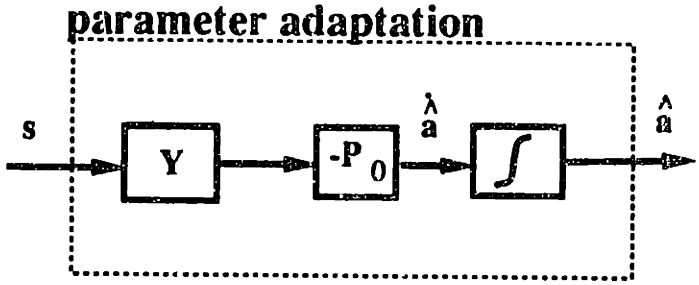


Figure 3-3: the parameter adaptation law

Note that the adaptation law can also be expressed as

$$\dot{\hat{\mathbf{a}}} = -\mathbf{P}_0 \frac{\partial \tau}{\partial \hat{\mathbf{a}}} \mathbf{s} \tag{3-17}$$

This form is reminiscent of the intuitive "MIT rule" [Astrom, 1989] in model-reference adaptive control, with the matrix $\mathbf{P}_0$ playing the role of adaptation gain and the output sensitivity function replaced by the torque sensitivity function. For adaptive controllers based on the MIT rules, stability can be guaranteed only for small adaptation gains. In the present adaptive controller, the magnitude of Γ does not affect the global stability of the system (as long as unmodeled dynamics are not excited), but it directly conditions the speed of adaptation, and therefore the system's performance.

In adaptive control, there are two major types of adaptive controllers: the model-reference adaptive controllers and the self-tuning regulators. One might be curious about which category this adaptive robot controller belongs to. The answer

is that, in a strict sense, it belongs to neither. Obviously, it is not a self-tuning regulator. It is not a model-reference adaptive controller either because it does not contain a reference model. However, in a generalized sense, it can be regarded as a model-reference adaptive controller with a unity model.

It is interesting to ponder why we do not need a reference model in our adaptive control system. To understand this, we have to go to the motivations for using a reference model in model-reference adaptive control. There are actually two motives in using reference models. First, a reference model specifies the desired performance of the control system. Since in many control applications, the externally provided reference input, such as the position measurement of an aircraft provided to a radar tracking system, does not directly specify the desired response of the control system (the control system response cannot possibly be the same as the external reference signal because of its inherent dynamics), a well-chosen reference model is needed to filter this signal to provide the ideal response of the control system. Secondly, a reference model, as a filter of the external input, provides the derivatives of the model output which allows perfect tracking of the reference signal. In our robot control problems, both motivations of the reference model are not necessary. On one hand, the desired performance of the control system has already been specified by the robot trajectory planner. There is no need of specifying them again. On the other hand, the first and second order derivatives of the desired position are also available and perfect tracking is possible without the need of introducing a reference model.

Most researchers [*e.g.*, Horowitz, 1981; Craig *et al*, 1986] in adaptive robot control included reference models in their adaptive controllers, *i.e.*, filter the desired trajectories between passing them to the adaptive control systems. Usually it is done simply to follow the tradition in the model-reference adaptive control

literature. In our adaptive controller here, even though it is straightforward to incorporate a reference model by simply filtering the desired trajectories before they are tracked, such a practice is not advocated. The reason is that such filtering causes distortion of the desired motion signals and causes motion errors in the execution of the practical tasks. For example, if a robot is required to draw a circle, a control system with a reference model can at best draw a "filtered circle".

3.3 Proving Global Tracking Convergence

In this section, we apply Barbalat's lemma (or Corollary 2.1) to show the global convergence of the tracking errors in the adaptive robot control system. By *global* tracking convergence, we mean that starting with *arbitrary* values of initial errors in $\tilde{q}(0)$, $\dot{\tilde{q}}(0)$ $\tilde{a}(0)$, the tracking errors $\tilde{q}(t)$ and $\dot{\tilde{q}}(t)$ will converge to zero asymptotically. The result is summarized in the following theorem.

Theorem 3.1: (global tracking convergence theorem) *Under the adaptive controller (3-14) and (3-12), the position tracking error $\tilde{q}(t)$ and velocity tracking error $\dot{\tilde{q}}(t)$ of the robot joints are guaranteed to be globally convergent to zero, and the estimated parameters guaranteed bounded, provided that the desired trajectories q_d, $\dot{q}_d$ and $\ddot{q}_d$ are bounded functions.*

Proof: The boundedness of $\tilde{a}$, $\tilde{q}$ and $\dot{\tilde{q}}$ have already been proven in subsection 3.2.4. To prove the global convergence of $\tilde{q}$ and $\dot{\tilde{q}}$, we only have to prove the convergence of s, according to lemma 3.1. The convergence of s can be proven by showing that $\dot{V} \rightarrow 0$, based on (3-13). Since V is a non-negative function and $\dot{V}$ has been shown to be non-positive, it follows from Corollary 2.1 that we only have to show that $\dot{V}$ is uniformly continuous in order to conclude the convergence of $\dot{V}$ to zero. Now let us prove the uniform continuity of $\dot{V}$ by showing that its

derivative $\ddot{V}$ is bounded, which can be demonstrated using a long chain of boundedness arguments.

From (3-13), one obtains that

$$\ddot{V}(t) = -2\, s^T K_d \dot{s}$$

Since s is bounded, the boundedness of $\ddot{V}$ can be guaranteed by the boundedness of $\dot{s}$. $\dot{s}$ can be determined from (3-15) as

$$\dot{s} = H^{-1}[\,-(K_d+C)s + Y\tilde{a}\,] \tag{3-18}$$

Therefore, the boundedness of $\dot{s}$ can be asserted by showing the boundedness of every term on the right-hand side of the equation.

The inverse of the inertia matrix, H^{-1}, is upper bounded, *i.e.*,

$$H^{-1} \le \frac{1}{\alpha} I$$

because of (2-6). The matrix K_d has been chosen to be bounded in the design of the adaptive controller, and s and $\tilde{a}$ have been shown to be bounded. Therefore, we only have to show the boundedness of $C(q, \dot{q})$ and $Y(q, \dot{q}, \dot{q}_r, \ddot{q}_r)$. Their boundedness is guaranteed by the already shown boundedness of $\tilde{q}$ and $\tilde{q}$, as we explain in the following.

Since q_d, $\dot{q}_d$ and $\ddot{q}_d$ are assumed to be bounded, q, $\dot{q}$, $\dot{q}_r$ and $\ddot{q}_r$ are also bounded, from their definitions. From the derivation of the Lagrangian equations, one can observe that the matrix C contains only bounded terms like products of the sine (and cosine) functions and the joint velocities. One can also conclude the boundedness of the matrix $Y(q, \dot{q}, \dot{q}_r, \ddot{q}_r)$ by examining its composition.

In summary the convergence proof is achieved by the following chain of

arguments: $\dot{s}$ bounded $\rightarrow \ddot{V}$ bounded $\rightarrow \dot{V}$ uniformly continuous $\rightarrow \dot{V}$ convergent $\rightarrow s$ convergent $\rightarrow \tilde{q}$ and $\dot{\tilde{q}}$ convergent.

□

Note that parameter error $\tilde{a}$ has been shown to be bounded, but has not been shown to be convergent to zero. In fact, additional conditions are required for the convergence of the estimated parameter. Such conditions will be investigated in chapter 4.

3.4 Simulation With a 2-DOF manipulator

Because the adaptive robot control system represents a highly nonlinear and complex dynamic system, theoretical analysis can only provide limited understanding about its behavior. Therefore, computer simulation represents a very important and useful tool in examining its behavior. This section presents the simulation of the proposed adaptive controller on a 2-DOF robot. There are a number of specific objectives in presenting the simulation results:

1. to verify the proven global stability and tracking convergence;
2. to gain insights into the adaptive control system;
3. to examine the effects of various factors on adaptive control;

The simulations are done on the robot model shown in Figure 3-4. We now discuss various aspects of this simulation study.

3.4.1 The Robot Dynamics

The manipulator shown in Figure 3-4 has a somewhat strange appearance. It is composed of four-bars but has only 2 degrees of freedom. It is assumed to be located in the horizontal plane and thus there is no gravity effect in the dynamics. This robot actually represents the model of the semi-direct-drive manipulator in the Whitaker college of MIT which we used for experimentally implementing the adaptive controller developed above. It is *semi*-direct drive because its second link is driven by a four bar link algorithm instead of gears or direct motor drives. Simulation results on this manipulator are presented because they will help us to gain insights into the experimental results in chapter 5.

The dynamics of this manipulator is quite similar to that of the manipulator in Figure 2.1. The appropriate generalized coordinates of the robot are the joint angles q_1 and q_2. After reparametrization by the equivalent parameters, the dynamic equation of this manipulator can be obtained to be

$$a_1 \ddot{q}_1 + (a_3 c_{21} + a_4 s_{21}) \ddot{q}_2 - (a_3 s_{21} - a_4 c_{21}) \dot{q}_2^2 = \tau_1 \qquad (3\text{-}19a)$$

$$(a_3 c_{21} + a_4 s_{21}) \ddot{q}_1 + a_2 \ddot{q}_2 + (a_3 s_{21} - a_4 c_{21}) \dot{q}_1^2 = \tau_2 \qquad (3\text{-}19b)$$

where $c_{21} = cos(q_2 - q_1)$, $s_{21} = sin(q_2 - q_1)$. It is clearly linear in terms of the four equivalent parameters a_1, a_2, a_3, a_4, which are related to the physical parameters of the links in Figure 3.4 through

$$a_1 = J_1 + J_a + m_2 l_1^2 \qquad (3\text{-}20a)$$

$$a_2 = J_2 + J_b + m_a r_a^2 \qquad (3\text{-}20b)$$

$$a_3 = m_2 h_2 l_1 cos\delta - m_a h_a r_a \qquad (3\text{-}20c)$$

$$a_4 = m_2 h_2 l_1 \sin\delta \qquad \text{(3-20d)}$$

Note from Figure 3-4 that the robot has "grasped" a load in its hand. In the above equations the load has been conveniently regarded part of the second link. This implies that if the inertial parameters of the load is uncertain, the physical parameters associated with the second link, *i.e.*, m_2, h_2, δ and J_2 are unknown. Correspondingly, the equivalent parameters a_2, a_3 and a_4 are unknown.

It is useful to point out that, though the knowledge of robot link parameters is not essential for adaptive control, such knowledge is helpful to reduce the parameter uncertainty and, thus, the initial tracking errors. For example, for the above robot, if the parameters of the load are unknown but those of the robot links known, the first equivalent parameter a_1 is known because it only depends on the physical parameters of the robot itself. Though the last three parameters are unknown due to the uncertainty in load parameters, the knowledge of robot parameters can be used to provide the useful initial estimates of the parameters a_2, a_3 and a_4.

The C matrix for this manipulator is

$$\mathbf{C} = \begin{bmatrix} 0 & a_4 c_{21}\dot{q}_2 - a_3 s_{21}\dot{q}_2 \\ a_3 s_{21}\dot{q}_1 - a_4 c_{21}\dot{q}_1 & 0 \end{bmatrix}$$

In our simulations throughout this thesis, the parameters of the model (3-19) are assumed to have the following values

$$a_1 = 0.15\ (Kg.m^2) \quad a_2 = 0.04\ (Kg.m^2) \quad a_3 = 0.03\ (Kg.m^2) \quad a_4 = 0.025\ (Kg.m^2)$$

These values are chosen because they are the approximate values of parameters

of the manipulator in the later experimentation.

3.4.2 The Adaptive Control Design

Since the structure of the adaptive controller has been determined in the previous sections, the design here only involves the choice of the design parameters which are contained in the three matrices $\mathbf{K}_d$, Λ, $\mathbf{P}_0$. All of these matrices have to be positive definite according to the previous design and analysis. For simplicity, they are chosen in the following simulations to be the product of positive numbers and unity matrices, *i.e.*,

$$\mathbf{K}_d = k_d \mathbf{I}$$

$$\Lambda = \hat{\lambda} \mathbf{I}$$

$$\mathbf{P}_0 = \gamma \mathbf{I}$$

The explicit form of the control law $\tau = \hat{\mathbf{H}}\,\ddot{\mathbf{q}}_r + \hat{\mathbf{C}}\,\dot{\mathbf{q}}_r - \mathbf{K}_d \mathbf{s}$ in terms of the parameter vector a is

$$\tau_1 = Y_{11} a_1 + Y_{13} a_3 + Y_{14} a_4 - k_d s_1 \qquad (3\text{-}21a)$$

$$\tau_2 = Y_{22} a_2 + Y_{23} a_3 + Y_{24} a_4 - k_d s_2 \qquad (3\text{-}21b)$$

where

$$Y_{11} = \ddot{q}_{r1}$$

$$Y_{13} = c_{21} \ddot{q}_{r2} - s_{21} \dot{q}_2 \dot{q}_{r2}$$

$$Y_{14} = s_{21} \ddot{q}_{r2} + c_{21} \dot{q}_2 \dot{q}_{r2}$$

$$Y_{22} = \ddot{q}_{r2}$$

$$Y_{23} = c_{21}\ddot{q}_{r1} + s_{21}\dot{q}_{r1}\dot{q}_1$$

$$Y_{24} = s_{21}\ddot{q}_{r1} - c_{21}\dot{q}_{r1}\dot{q}_1$$

and

$$\dot{q}_{r1} = \dot{q}_{d1} - \lambda\tilde{q}_1$$

$$\dot{q}_{r2} = \dot{q}_{d2} - \lambda\tilde{q}_2$$

$$s_1 = \dot{\tilde{q}}_1 + \lambda\tilde{q}_1$$

$$s_2 = \dot{\tilde{q}}_2 + \lambda\tilde{q}_2$$

The adaptation law can be explicitly written as

$$\dot{\hat{a}}_1 = -\gamma Y_{11}s_1 \qquad (3\text{-}22a)$$

$$\dot{\hat{a}}_2 = -\gamma Y_{22}s_2 \qquad (3\text{-}22b)$$

$$\dot{\hat{a}}_3 = -\gamma(Y_{13}s_1 + Y_{23}s_2) \qquad (3\text{-}22c)$$

$$\dot{\hat{a}}_4 = -\gamma(Y_{14}s_1 + Y_{24}s_2) \qquad (3\text{-}22d)$$

3.4.3 Intuitive Interpretation of the Adaptation Law

The above equations have strong intuitive appeal. The rate of adaptation is linearly proportional to the tracking error measure s. The estimate $\hat{a}_1$ is driven only by the tracking error of the first joint. This is, according to (3-17), due to the fact that a_1 appears only in the first joint motion equation (3-19a) or the torque input equation (3-21a); similarly for $\hat{a}_2$. On the other hand, $\hat{a}_3$ is driven by a combination of the first joint error and the second joint error, because a_3 appear in both (3-19a) and (3-19b); and similarly for $\hat{a}_4$.

Let us take a closer look at (3-22a) and see how the adaptation mechanism extracts information about parameter uncertainty from the tracking error measure s. Consider for instance the case when Y_{11} is positive. If s_1 is negative, meaning a lag behind the desired motion in the first joint, adaptation law (3-22a) increases the estimate $\hat{a}_1$. The reason for this increase is that the adaptation mechanism interprets the lag as indicating an insufficient joint torque τ_1 , and since $(\partial\tau_1/\partial\hat{a}_1) = Y_{11} > 0$, this is in turn interpreted as an insufficiency in $\hat{a}_1$. On the other hand, this increase in $\hat{a}_1$ leads to an increase in the torque τ_1 , as seen from (3-21a), and, accordingly, to a reduction in the lag behind the desired motion. The adaptation mechanism of $\hat{a}_2$ can be explained using the same argument. The variations of $\hat{a}_3$ and $\hat{a}_4$ can be explained similarly, with the difference that these parameter estimates are driven by weighted averages of s_1 and s_2 . The weighting factors for $\hat{a}_3$, for instance, are Y_{13} and Y_{23} ; their magnitudes reflect the extent of the contribution $\hat{a}_3$ to τ_1 and τ_2 , a: ' the corresponding lags in the joints.

Thus, the adaptation law generates the parameter estimates in an intuitively reasonable or "intelligent" manner.

3.4.4 Results from Adaptive and PD Controllers

Let the desired position trajectories to be followed be given by the following functions

$$\begin{bmatrix} q_{d1}(t) \\ q_{d2}(t) \end{bmatrix} = = \begin{bmatrix} \frac{\pi}{4} + 2(1 - cos3t) \\ \frac{\pi}{6} + (1 - cos5t) \end{bmatrix} \qquad (3\text{–}23)$$

There is one sinusoid component at each joint. The desired positions are plotted in Figure 3-5. The initial values of the positions and velocities are zero. But note

that the initial desired accelerations are not zero.

Let us consider the control of the manipulator under the above adaptive controller. To provide a reference point for the adaptive control performance, a PD controller is also simulated. The PD controller has the following form

$$\tau_1 = -k_d\dot{\tilde{q}}_1 - k_d\lambda\tilde{q}_2$$

$$\tau_2 = -k_d\dot{\tilde{q}}_2 - k_d\lambda\tilde{q}_2$$

This means that the PD controller is identically the same as the PD part of the adaptive controller. Thus the difference in the performance of the adaptive controller and PD controller is attributable to the adaptive feedforward part of the adaptive controller.

For both the adaptive controller and the PD controller, the values of the design parameters are taken to be

$$\gamma = 0.03\ (Kg.m^2.sec^2) \qquad k_d = 3.\ (N.m.sec) \qquad \lambda = 10.(N.m)$$

The initial position and velocity tracking errors are taken to be zero. To highlight the effects of the adaptation, we take the initial values of the estimated parameters to be all zero, *i.e.*,

$$\hat{a}_1(0) = \hat{a}_2(0) = \hat{a}_3(0) = \hat{a}_4(0) = 0$$

This implying that there is no *a priori* knowledge in the parameters of the links and the load, and thus the adaptive controller is initially the same as the PD controller. In understanding the following results, It is helpful to remember that a PD controller $\tau = -\mathbf{K}_d\dot{\tilde{\mathbf{q}}} - \mathbf{K}_p\tilde{\mathbf{q}}$ is physically equivalent to a set of joint dampers and springs with coefficients given by the gain matrices $\mathbf{K}_d$ and $\mathbf{K}_p$.

The results of the manipulator under the adaptive controller are plotted in Figures 3-6 and the results under PD control are shown in Figures 3-7. It is noted from Figure 3-6(a) that the position tracking errors and velocity tracking errors (the slopes of the position error curves) of the adaptive controller are both asymptotically convergent, as predicted by the analysis. The comparison of the position tracking error of the adaptive and PD controllers, shown in Figure 3-6(a) ad 3-7(a), are quite interesting. We note that in the initial period the tracking errors of both controllers show the same tendency of growing negatively (implying the robot motion lags behind the desired motion). This tendency is due to the passive nature of the PD control (mimicking joint dampers and springs). This tendency is reversed for the adaptive controller after a short time, with errors reaching only -0.7^0 for the first joint and -3.1^0 for the second joint, while it continues for the PD controller until the errors reach -2^0 and -6.5^0 respectively. The reason for the reversal of the tracking error growth and subsequent decay of the tracking errors in Figure 3.6(a) is the increasing effects of the adaptive feedforward compensation. The tracking errors of the PD controllers stop growing negatively and roll back because the desired joint accelerations change directions. The tracking errors of the PD controller do not converge because PD controller, mimicking dampers and springs, cannot fully compensate the dynamic forces during robot motion.

The estimated parameters of the adaptive controller for this particular trajectories, shown in Figure 3-6(c) and Figure 3-6(d), are seen to converge to the true parameters $\mathbf{a} = \{0.15, 0.04, 0.03, 0.025\}^T$ (Kg.m^2). It is seen that they change fast when there are large tracking errors and change slowly when tracking errors become small, as expected from the form of the adaptation law. It is important to note that the joint torques of the adaptive controller, shown in Figure 3-6(b) has

comparable magnitudes as those of the PD controller shown in Figure 3-7(b). This implies that the adaptive controller gains the better tracking motion not by stronger control authority but rather by "smarter" use of the input. Note that the τ_1 in Figure 3-6(b) has a somewhat sharp peak (larger than 4 N.m) at about 0.1 second. This is the only time when the joint torques of the adaptive controller are visibly larger than those of the PD controller. This peak occurs because the system has accelerated (noting that the initial desired accelerations are not zero) before the proper estimates of the parameters are found by the adaptation law. For desired trajectories which have zero initial desired accelerations, this effect is much relieved, as will be seen later for a different trajectory.

3.4.5 Effects of Design Parameters

Though the stability and convergence of the adaptive controller is guaranteed for any positive values of the design parameter matrices $\mathbf{K}_d$, Λ and Γ, the quantitative behavior, such as the speed of convergence and magnitude of the maximum tracking errors, of the adaptive controller is dependent on magnitudes of these matrices. In this section, let us use simulation results to study the effects of the design parameter choices. In the following, we simulate the adaptive controller on the desired trajectories of sinusoid type shown in Figure 3-5, with different values of the design parameters. Thus the results in Figure 3.6 provide a reference point for comparison.

A. the adaptation gain

It is expected that larger adaptation gains correspond to better adaptive control, because it implies faster adaptation. To verify this, the gain matrix in subsection 3.4.4 ($\mathbf{P} = 0.03\,\mathbf{I}$) is increased to

$$P = 0.1\,I$$

with the rest of the adaptive controller being unchanged. The results are plotted in Figure 3-8. It is observed that the tracking errors are now considerably smaller than those in Figure 3.6. The estimated parameters rise more quickly, and converge faster. We also note that the results in Figure 3-8 are more oscillatory than those in Figure 3-6.

However, this relation of increasing convergence with increasing gain is true only to certain point. With further increase of the adaptation gain, the convergence of the adaptive controller is found to be worse, as shown in Figure 3.9. The explanation is that the the adaptation is of a gradient nature, and too large gain leads to overshooting in the gradient search of the ideal parameters.

Therefore, it is important to use proper adaptation gain in using the adaptive controller. This is sometimes difficult to do because the optimal gain value may depend on the trajectories. This problem will later be avoided in an improved version of the adaptation law, called composite adaptation law, where the gain can be pushed much higher without incurring severe parameter oscillation. The improved adaptation law is useful for enhancing tracking performance and handling parameter variations.

B. the gain matrices K_d

Next, we examine the effects of the K_d matrix on the performance of the adaptive controller. If K_d is increased with all other design parameters fixed, one naturally expect the adaptive control performance to improve, because the action of the PD part in the controller is strengthened. Note that in increasing K_d, the gain of proportional feedback term ($K_d \Lambda \tilde{q}$) is also increased. To see the effect of

increasing the D (derivative) part alone, let us maintain the proportional feedback part of the controller unchanged while K_d is varied. This implies that Λ is varied inversely. Now, we consider the following values of K_d and Λ,

$$K_d = 6.I$$

$$\Lambda = 5.I$$

with the adaptation gain still being $\Gamma = 0.03\,I$ and the initial parameters still zero. The above implies that the gain of the derivative part is doubled compared with that in subsection 3.4.4.

The simulation results are shown in Figure 3-10. Compared with the results in Figure 3.6, it is noted that the initial tracking errors are considerably smaller. The first joint torque has a smaller initial peak corresponding to the larger initial errors. It is interesting to note that the estimated parameters are more sluggish. A plausible explanation for this sluggishness is that the increased PD action leads to the reduced error in the joint position and thus a reduced need for adaptation.

Simulations with a reduced K_d show converse effects on the adaptive control, indicating the consistency of the above relations between K_d and the adaptive control results.

C. the gain matrix A

Finally, let us examine the effects of the gain matrix Λ. A larger value of λ than that in subsection 3.4.4 is simulated,

$$\Lambda = 20I$$

The results of the adaptive control are given in Figure 3-11. Compared with

Figure 3.6, one notes that the tracking error is better but the estimated parameters are somewhat worse. This decrease in the quality of the estimated parameter is explained as before for the case of increased K_d.

It is found that smaller A leads to converse effects on the adaptive control. Thus we can conclude that larger A leads to better tracking error but worse parameter error.

3.4.6 Effects of Desired Trajectories

One of the factors affecting the behavior of the proposed adaptive controller is the features of the desired motion trajectories. First, the convergence of the estimated parameters in the adaptive control system depends on the complexity of the desired trajectories. Secondly, the tracking errors of the system are dependent on the speed and acceleration of the desired trajectories. For a given amount of parameter uncertainty, desired trajectories with higher velocity and/or accelerations lead to smaller tracking errors. This is reasonable intuitively, and can also be seen from examining the closed-loop dynamics (3-15) whose left-hand side is related to the desired velocities and accelerations. In this subsection, we shall illustrate the relations between the desired trajectories and the adaptive control behavior, particularly the relation between the trajectory complexity and parameter convergence.

A. polynomial+rest desired trajectories

We now consider a desired motion different from that for the previous simulations. The desired position trajectories are plotted in Figure 3-12. Physically, it represents a back-forth motion between the point A and B shown in Figure 3.13. The four-second motion is composed of four parts: one-second

motion from point A to point B, one-second rest at point B, one-second motion back to point A, and one-second rest at point A.

This motion more or less emulates the industrial robot motions in pick-and-place tasks. The resting periods are introduced to let the robot settle down around the final positions. For this motion, the point A corresponds to the joint angles (20^0, 70^0) and point B to (100^0, 140^0). In the first-second motion from A to B. the desired position histories are defined by the following first-order polynomials

$$\begin{bmatrix} q_{d1}(t) \\ q_{d2}(t) \end{bmatrix} = \begin{bmatrix} q_{A1} + \Delta q_1 (10 - 15t + 6t^2 t^3) \\ q_{A2} + \Delta q_2 (10 - 15t + 6t^2 t^3) \end{bmatrix} \tag{3-24}$$

where $\Delta q_1 = (q_{B1} - q_{A1})$, $\Delta q_2 = (q_{B2} - q_{A2})$, and

$$\begin{bmatrix} q_{A1} \\ q_{A2} \end{bmatrix} = \begin{bmatrix} 20^0 \\ 70^0 \end{bmatrix} \qquad \begin{bmatrix} q_{B1} \\ q_{B2} \end{bmatrix} = \begin{bmatrix} 100^0 \\ 140^0 \end{bmatrix}$$

Expression (3-24) implies that the desired velocities and desired accelerations are zero at point A and point B are both zero. In the third-second motion, the desired position histories are similarly defined by fifth-order polynomials.

B. the control results

The results of the adaptive controller on this set of desired trajectories are plotted in Figure 3-14. From 3-14(a), the tracking errors of the adaptive controller are seen to display convergence tendency. But, from the 4-seconds simulation results, it is not obvious at all whether the estimated parameters converge to the true parameters $a = \{0.15, 0.04, 0.03, 0.025\}$. To see how the estimated parameters look like after a long time, we let the robot repeat the motion specified

by Figure 3.11 for ten times. The position tracking errors and estimated parameters during the last 4-seconds are shown in Figures 3-15. It is seen that the tracking errors have converged to small values but the estimated parameters are far away from the true parameters.

To see the comparison of performance, the PD controller is also run on this set of desired trajectories. The position tracking errors and joint torques are shown in Figure 3-16. Comparing 3-13(a) and 3-15(a), one notes that the difference in the tracking errors in the first second of motion is not very much for the tow controllers, unlike the quick improvement shown in Figure 3-6(a) corresponding to the fast sinusoid trajectory. The reason is that the desired trajectories here are relatively slow and it takes longer time for the adaptation to have substantial effects on the tracking errors. Another interesting observation is that the joint torques shown in Figure 3-14(b) and (3-16(b) are very similar. The initial peaks seen in Figure 3-6(b) are not present in Figure 3-14(b). This absence of torque peaks is attributable to the fact that the initial desired accelerations here are zero (which allows the adaptation law time to find reasonable parameters before the motion speed and acceleration reach considerable magnitudes).

C. Polynomial trajectories

One might suspect that the lack of parameter convergence is due to the resting periods in the above desired motion. To see that this is not true, the adaptive controller is simulated on the desired trajectories shown in Figure 3-17, which corresponds to motions back and forth between point A and B without any resting periods. The results of the adaptive controller are shown in Figure 3-18. They are seen to be quite similar to the results in Figure 3-14, though the tracking errors converge faster due to the lack of the pauses. The adaptive controller is also run on

this set of desired trajectories for ten times. The results in the last four seconds display the features of having small tracking errors but large parameter errors, similarly to the case of the polynomial+rest trajectories. It is interesting to note that the desired position histories in Figure 3-17 look quite similar to those in Figure 3-5, but the joint torques and estimated parameters look very different.

The main lesson to be learned from the simulation results in this subsection is that for certain trajectories, it is possible for the adaptive controller to *control the motion accurately with inaccurate estimated parameters*. The issue of parameter convergence will be studied in in detail chapter 4.

3.4.7 Effects of Non-Parametric Uncertainties

In the previous simulations, we have assumed the absence of non-parametric uncertainties. Now let us examine the performance and stability of the adaptive controller when non-parametric uncertainties are present in the plant.

A. Effects of Disturbances

A common disturbance in robot control is the often neglected Coulumb friction. Suppose there is an unmodeled Coulumb friction of 0.3 N.m in each joint, *i.e.*, with disturbance torques of

$$d_1(t) = -0.3\, sgn(\dot{q}_1) \quad (N.m)$$

$$d_2(t) = -0.3\, sgn(\dot{q}_2) \quad (N.m)$$

The results of the adaptive control on the sinusoidal-type trajectories in Figure 3.5 is shown in Figure 3.19. It is seen that the tracking errors now converge to within 1 degree for each joint, instead of to zero. The estimated parameters now

oscillate around the true parameters.

The sinusoidal-type trajectories used in the above simulation are expected to be persistently exciting and lead to robust performance (to be explained further in Chapter 4). It is therefore of interest to see the adaptive control performance on the less exciting polynomial-type trajectories. This is done in Figure 3.20 which shows the results when the polynomial+rest trajectories in Figure 3.12 are used. It is seen that the adaptive controller still performs well, with little parameter drift observed.

B. Effects of Motor Dynamics

Now let us assume that the joint actuators have first order filter dynamics of bandwidths 30 (rad/sec), *i.e.*,

$$\tau_1 = \frac{30}{p+30} u_1$$

$$\tau_2 = \frac{30}{p+30} u_2$$

where u_1 and u_2 are the joint torque commands given by the computer, and τ_1 and τ_2 are the joint torques delivered by the actuators (note that no other non-parametric uncertainties included). Note that now u_1 and u_2 are computed by the adaptive controller by (3-11).

The results of the adaptive controller for the sinusoidal-type trajectories are shown in Figure 3.21 and the results for the polynomial-rest trajectories in Figure 3.22. It is seen that the adaptive controller performs well in both cases.

C. Effects of Other Non-parametric Uncertainties

The effects of a number of other non-parametric uncertainties have also been examined by simulations. The adaptive controller has been found to perform well in general. However, when measurement noise is present and the trajectories are not persistently exciting (on polynomial-type trajectories, for example), the estimated parameters are found to drift slowly, though tracking errors remain small. This can lead to adaptive control instability if the unmodified adaptive controller runs for a long time and the estimated parameters drift to huge numbers.

It is useful to point out that the problem of parameter drift is not a significant problem in the practice of adaptive robot control for two reasons. First, in most robotic manipulations using adaptive control, the manipulations usually last only for a few seconds and parameter drift cannot pose a serious threat in such a short time. Secondly, if long-time manipulation is desired, the problem of parameter drift can be greatly relieved by incorporating a small dead-zone in the adaptation law, as verifiable by simulations. We remark that, if an unknown load has to be manipulated with a long time, it might be advantageous to use parameter estimation followed by non-adaptive control, *i.e*, shaking the load for a couple of seconds and estimating the parameters on-line. The on-line parameter estimators in chapter 6 may be used during the shaking motion.

3.5 Discussions Regarding the Adaptive Controller

To gain insights about the derivation and application of the above adaptive controller, let us now discuss a few interesting points.

Number of parameters to be adjusted: Since the link parameters do not change when a robot picks up a new load, it is possible to accurately estimate the robot link parameters beforehand and take them as known in the adaptive control.

In this way, only the ten parameters associated with manipulated load are unknown (one for mass, three for center of mass and six for moments of inertia) and need be estimated on-line.

Sampling frequency: The adaptive controller has been designed in continuous time, but its implementation has to be in discrete time because of the use of digital computation. In order to best use the limited computational power, a good strategy is to use different rates for the update of various parts of the adaptive controller. Because of its simplicity, the PD control part in the control law should be update with a rate much higher than those of the feedforward part and the estimated parameters.

One interesting question is the appropriate choice of sampling rates. A number of factors affect the values of appropriate sampling frequencies for an adaptive robot control system. One of the major factors is the speed of the desired trajectories: faster desired trajectories require the control system to respond faster, and thus require higher sampling rates. If the sampling rate is too slow compared with the desired trajectories, the control system will have large tracking errors or even instability. Another factor is the bandwidth of the motors: performance is not further improved much with the increase of sampling rates after the sampling is already faster than the motor bandwidth. The reason for this is that the motors are basically filtering devices which do not respond to high frequency components of its input. The implication of the filtering is that too fast sampling wastes computational power.

Recursive computation of the adaptive controller: It is important to point out that the adaptive controller (3-14) and (3-12) is computationally efficient. In [Walker 1988], a recursive computation technique was developed for the above

adaptive controller and applied to a closed-chain robot.

For computed-torque controller, the recursive Neuton-Euler scheme is known to provide efficient control computation. In [Niemeyer and Slotine, 1988], a modified version of the Neuton-Euler scheme is developed to carry out the computation of the adaptive controller proposed above. The required computational effort of the control law (3-14) is roughly the same as that required for the recursive Neuton-Euler implementation of a computed-torque controller. With this recursive computational method, the implementation of the proposed adaptive controller on 6-d.o.f. robots is within the computational capacity of current microprocessors. If fact, the adaptive controller has been implemented on a 4-link high-performance robot at the Artificial Intelligence Laboratory of MIT, with good experimental results achieved. What is most interesting about this implementation is that a total of *40 parameters* are assumed to be unknown, and three microprocessors operating in parallel were able to accomplish the required control and adaptation computation. In practice, the parameters to be estimated are only the ten parameters of the load, and therefore, the computational burden will be substantially smaller than that in the 40-parameter experiments.

Unique features of the the adaptive controller development: Compared with what is generally done in adaptive control literature, the development of this adaptive controller has some unique features. First, the state-space model of the robot dynamics is not used. If we rewrite the robot dynamics in terms state-space representation, *i.e.*,

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_1 \\ H^{-1}(\tau - C\dot{q} - g) \end{bmatrix}$$

with $\dot{x}_1 = q$, $x_2 = \dot{q}$, not only the dynamics becomes much more complicated,

the lack of insights with this representation would have probably rendered the development of the globally tracking-convergent adaptive controller impossible. Secondly, the physical properties of the dynamic systems are carefully employed. In view of the fact that current research in adaptive control tends to consider mathematically classified plants, the successful development of the above adaptive controller for the complex robot dynamics seems to suggest a new direction of adaptive control research: study particular classes of physical systems instead.

Design based on a Lyapunov Function: Though the above derivation has been based on Barbalat's lemma and a Lyapunov-like function, it is possible to derive a somewhat restricted version of the previous adaptive controller based on Lyapunov's direct method and Lyapunov functions, as we now show. Consider the following

$$V = \frac{1}{2}\left[\mathbf{s}^T \mathbf{H} \mathbf{s} + 2\tilde{\mathbf{q}}^T \mathbf{\Lambda} \mathbf{K}_d \tilde{\mathbf{q}} + \tilde{\mathbf{a}}^T \mathbf{\Gamma}^{-1} \tilde{\mathbf{a}} \right] \tag{3-25}$$

where $\mathbf{K}_d$ and $\mathbf{\Lambda}$ are positive definite and *diagonal* matrices. Note that this is a strictly positive definite function in terms of the closed-loop system states and thus is qualified to serve as a Lyapunov function candidate. The same forms of control law and adaptation law as given by (3-14) and (3-12) lead to the following derivative along system trajectories

$$\begin{aligned} \dot{V} &= -\mathbf{s}^T \mathbf{K}_d \mathbf{s} + 2\tilde{\mathbf{q}}^T \mathbf{\Lambda} \mathbf{K}_d \tilde{\mathbf{q}} \\ &= -\dot{\tilde{\mathbf{q}}}^T \mathbf{K}_d \dot{\tilde{\mathbf{q}}} - \tilde{\mathbf{q}}^T \mathbf{\Lambda}^T \mathbf{K}_d \mathbf{\Lambda} \tilde{\mathbf{q}} \end{aligned}$$

This non-positive nature of $\dot{V}$ indicates that (3-25) provides a Lyapunov function for the adaptive control system and thus guarantees the global stability and tracking convergence of the adaptive system.

As will be seen later, it might be desirable to allow $\mathbf{K}_d$ to be a full matrix. This

can be achieved if $\mathbf{\Lambda}$ is restricted to the product of a positive number and a unit matrix, *i.e.*, $\mathbf{\Lambda} = \lambda \mathbf{I}$. In this way, the Lyapunov function candidate is

$$V = \frac{1}{2}[\, s^T \mathbf{H} s + 2\lambda \tilde{\mathbf{q}}^T \mathbf{K}_d \tilde{\mathbf{q}} + \tilde{\mathbf{a}}^T \mathbf{\Gamma}^{-1} \tilde{\mathbf{a}}]$$

and the corresponding derivative is

$$\dot{V} = -\dot{\tilde{\mathbf{q}}}^T \mathbf{K}_d \dot{\tilde{\mathbf{q}}} - \lambda^2 \tilde{\mathbf{q}}^T \mathbf{K}_d \tilde{\mathbf{q}}$$

This also guarantees the convergence of the tracking errors.

Note, however, that when both $\mathbf{K}_d$ and $\mathbf{\Lambda}$ are positive definite *full* matrices, the function V in (3-25) may not be positive definite (both $\mathbf{K}_d \mathbf{\Lambda}$ may not be positive definite) and thus may not be qualified as a Lyapunov function candidate. This implies that the design in section 3.2 based on *Lyapunov-like* function leads to an adaptive controller with more flexible design parameters.

The Concept of Reference Velocity: The definition of reference velocity given in (3-1) may seem to be somewhat strange. It is motivated by our desire of making position tracking error converge to zero. A discussion of its origin is interesting. Initially, we had chosen the following form of Lyapunov-like function candidate

$$V(t) = (1/2)[\dot{\tilde{\mathbf{q}}}^T \mathbf{H} \dot{\tilde{\mathbf{q}}} + \tilde{\mathbf{q}}^T \mathbf{K}_p \tilde{\mathbf{q}}] + (1/2)\, \tilde{\mathbf{a}}^T \mathbf{\Gamma}^{-1} \tilde{\mathbf{a}} \qquad (3\text{-}26)$$

with $\mathbf{K}_p$ being a positive definite matrix, which is strictly positive definite and the first two terms has the appearance of mechanical energy. By choosing the control law to be

$$\boldsymbol{\tau} = \hat{\mathbf{H}} \ddot{\mathbf{q}}_d + \mathbf{C} \dot{\mathbf{q}}_d + \hat{\mathbf{g}} - \mathbf{K}_d \dot{\tilde{\mathbf{q}}} - \mathbf{K}_p \tilde{\mathbf{q}} \qquad (3\text{-}27)$$

the derivative of V along the system trajectories is easily found to be

$$\dot{V} = -\dot{\tilde{\mathbf{q}}}^T \mathbf{K}_d \dot{\tilde{\mathbf{q}}} \qquad (3\text{-}28)$$

The control law (3-27) has a nice form which contains the explicit PD feedback terms, but only the velocity convergence is guaranteed by (3-28), *i.e.*,

$$\dot{\mathbf{q}} - \dot{\mathbf{q}}_d \to 0$$

Position tracking convergence is not guaranteed, as was verified by simulation results. The concept of reference velocity is introduced in place of $\dot{\mathbf{q}}_d$ in (3-26) so that (3-28) leads to

$$\dot{\mathbf{q}} - \dot{\mathbf{q}}_r \to 0$$

This guarantees the convergence of both $\tilde{\mathbf{q}}$ and $\dot{\tilde{\mathbf{q}}}$. In doing so, it is also found that the second term $(1/2)\tilde{\mathbf{q}}^T \mathbf{K}_p \tilde{\mathbf{q}}$ on the right-hand side of (3-26) can be eliminated without affecting $\dot{V}$. Then we end up with the Lyapunov-like function (3-8).

Adaptation in position control problems: There are two types of control problems in robot free motion, one being the position control problems (only the initial and final positions of the robot are specified) and the other being the tracking control problems (the whole desired trajectory is specified). It is useful to point out that adaptive control is primarily concerned with the tracking control problem. Although position control problems can also benefit from adaptive control by adaptively compensating the gravity term, the tracking control problems are the real targets of adaptive control approach. The reason is that the position control can be often handled by PID controllers which do not rely on robot model and thus are insensitive with respect to parameter uncertainty. The problem with PID controllers is the overshoot around the final position. In tasks such overshoot is undesirable, one can either use a modified version of PID control or translate the position control problem into a tracking control problem. One way of translating the position control problem is to filter the desired final

positions, and let the output of the filter by tracked by an adaptive controller.

Role of $(\dot{H} - 2C)$: the above adaptive controller used the skew-symmetry property of the matrix $(\dot{H} - 2C)$ so that the term associated with $\dot{H}$ in (3-9) can be removed. However, the use of this property is not essential for the formulation of globally convergent adaptive controller for robots. If we change the definition of the matrix Y in the previous control and adaptation laws to be

$$\tilde{H}\ddot{q}_r + \tilde{C}\dot{q} + \tilde{g} + (1/2)(\dot{\hat{H}}s - \dot{H}s) = Y_c\tilde{a}$$

then the resulting adaptive controller can be easily shown to have the same global convergence property using a similar analysis. Specifically, from (3-9),

$$\dot{V} = s^T[\tau - C\dot{q} - G - H\ddot{q}_r - (1/2)\dot{H}s] + \tilde{a}^T\Gamma^{-1}\dot{\tilde{a}}$$

Using the control law

$$\tau = \hat{H}\ddot{q}_r + \hat{C}\dot{q} + \hat{g} + \frac{1}{2}\dot{\hat{H}}s - K_d s$$

and the adaptation law

$$\dot{\hat{a}} = -\Gamma Y_c^T s$$

we also obtain (3-13) and the associated properties.

Known Parameter Case: When there is no parameter uncertainty, there is no need for adaptation. The control law in (3-14) can be used with the unknown parameters $\hat{a}$ substituted for by the known parameters a. The resulting controller is not the same as the computed-torque controller. It may be regarded as an alternative to the computed-torque controller for the non-adaptive control of robots.

For this non-adaptive controller, the control system can now be shown to be exponentially convergent. In the following let us consider the choice of the gain (damping) matrix being

$$\mathbf{K}_d = \lambda \mathbf{H} \tag{3-29}$$

This choice of adjust $\mathbf{K}_d$ according to the inertia matrix $\mathbf{H}$ is intuitively appealing: larger damping for joints of larger inertia.

To show the exponential tracking convergence, let us consider the same function V as before, *i.e.*,

$$V = \frac{1}{2} \mathbf{s}^T \mathbf{H} \mathbf{s}$$

Differentiation of V yields

$$\dot{V} = -\lambda \mathbf{s}^T \mathbf{H} \mathbf{s} = -2\lambda V \tag{3-30}$$

This implies that $V(t) = V(0)e^{-2\lambda t}$. Due to the uniform positive definiteness of $\mathbf{H}$,

$$\alpha \|\mathbf{s}\|^2 \leq V(t) = V(0)e^{-2\lambda t}$$

Therefore,

$$\|\mathbf{s}\|^2 \leq \frac{V(0)}{\alpha} e^{-2\lambda t}$$

This means that s is exponentially convergent to zero with a rate λ, which, according to lemma 3.1, implies the exponential convergence of tracking errors $\tilde{\mathbf{q}}$ and $\dot{\tilde{\mathbf{q}}}$.

It is interesting to note that the control law can be rewritten as

$$\tau = H[\ddot{q}_d - 2\lambda\dot{\tilde{q}} - \lambda^2\tilde{q}] + C\dot{q}_r + g \qquad (3\text{-}31)$$

This control law is almost the same as the standard computed torque controller, except for a minor difference in the second term. This control law is interesting because it does not exactly cancels the robot dynamics due to the form of the second term in (3-31) but it can guarantee exponential convergence at the same rate. This may be attributed to the fact that the control utilizes the inherent structure of the robot dynamics.

3.6 Variations of the Adaptive Controller

A number of variations of the adaptive controller in (3-14) and (3-12) can be simply obtained. Such variations maintain the desirable stability and convergence properties of basic adaptive controller and introduce new and interesting features into the adaptive control system. Whether such variations have fundamental advantages over the the adaptive controller in section 3.2 is still to be decided by more examination, particularly experimental study.

A. Choose $K_d = \lambda\hat{H}$

The matrix K_d contains design parameters to be chosen by the user. As seen in section 3.2, the matrix can be time-varying but must be uniformly positive definite. Motivated by the choice of K_d at the end of last section, one might be interested in using

$$K_d = \lambda\hat{H} \qquad (3\text{-}32)$$

in the adaptive case. However, there is a problem with this choice: the matrix $\hat{H}$ may not be always positive definite in the course of parameter adaptation. We now show that, with a simple modification in the adaptation law, this problem can

be neatly handled.

Let us still consider the Lyapunov-like function candidate (3-8). With controller gain K_d determined by (3-32), its derivative is

$$\dot{V} = s^T[\tilde{H}\ddot{q}_r + \tilde{C}\dot{q}_r + \tilde{g} - \lambda\hat{H}s] + \tilde{a}^T\Gamma^{-1}\dot{\tilde{a}}$$

$$= s^T[-\lambda Hs + \tilde{H}(\ddot{q}_r - \lambda s) + \tilde{C}\dot{q}_r + \tilde{g}] + \tilde{a}^T\Gamma^{-1}\dot{\tilde{a}}$$

If we let Y_m denote the modified Y matrix, defined by

$$\tilde{H}(\ddot{q}_r - \lambda s) + \tilde{C}\dot{q}_r + \tilde{g} = Y_m(q, \dot{q}, \dot{q}_r, \ddot{q}_r)\,\tilde{a}$$

then

$$\dot{V} = -\lambda s^T Hs + \tilde{a}^T[\Gamma^{-1}\dot{\hat{a}} + Y_m s]$$

It is obvious that the adaptation law

$$\dot{\hat{a}} = -\Gamma Y_m{}^T s \tag{3-33}$$

leads to

$$\dot{V} = -\lambda s^T Hs \tag{3-34}$$

Since the inertia matrix H is always uniformly positive definite, the above expression guarantees the global stability and the global tracking convergence of the adaptive control system, using similar reasoning to that in section 3.3. Note that with the matrix K_d determined by (3-32), the control law is given by

$$\tau = Y_m\hat{a}$$

B. PI type adaptation

Other variations of the algorithm can be easily derived. For instance, one may

add a "proportional" term to the parameter update

$$\hat{\mathbf{a}} = \hat{\mathbf{a}}_0 - \Gamma_1 \mathbf{Y}^T \mathbf{s} \tag{3-35}$$

where Γ_1 is a (perhaps time-varying) symmetric positive definite matrix, with

$$\dot{\hat{\mathbf{a}}}_0 (t) = - \Gamma \mathbf{Y}^T \mathbf{s}$$

as before. Indeed, using

$$V(t) = \frac{1}{2} [\, \mathbf{s}^T \mathbf{H} \mathbf{s} + \tilde{\mathbf{a}}_0^T \Gamma^{-1} \tilde{\mathbf{a}}_0 \,]$$

as a Lyapunov function candidate, where $\tilde{\mathbf{a}}_0 = \hat{\mathbf{a}}_0 - \mathbf{a}$, we now get

$$\dot{V}(t) = - \mathbf{s}^T (\mathbf{K}_d + \mathbf{Y} \Gamma_1 \mathbf{Y}^T) \mathbf{s} \leq 0$$

While such modification may thus increase the convergence rate of the algorithm, it may also lead to noisier estimates, since $\hat{\mathbf{a}}$ now incorporates the term $\mathbf{Y}^T\mathbf{s}$ directly. Note that this modification of the adaptation law, originally inspired by a passivity interpretation of the adaptive controller, is found to be equivalent to replacing the gain matrix $\mathbf{K}_d$ by $\mathbf{K}_d + \mathbf{Y} \Gamma_1 \mathbf{Y}^T$.

C. Incorporating Integral Feedback

Examining the derivation in section 3.2 and the convergence analysis in section 3.3, it is noted that the convergence of s to zero, *i.e.*, $\dot{\mathbf{q}}$ to $\dot{\mathbf{q}}_r$ is guaranteed regardless of the definition of $\dot{\mathbf{q}}_r$. In the above development, the reference velocity $\dot{\mathbf{q}}_r$ is obtained by modifying the desired velocity by position tracking errors. Based on the above point, integration of position error may also be incorporated in the reference velocity,

$$\dot{\mathbf{q}}_r = \dot{\mathbf{q}}_d - \Lambda_1 \tilde{\mathbf{q}} - \Lambda_2 \int_0^t \tilde{\mathbf{q}}(r)\, dr \tag{3-36}$$

the definition of s is

$$s = \dot{q} - \dot{q}_r = \dot{\tilde{q}} + \Lambda_1 \tilde{q} + \Lambda_2 \int_0^t \tilde{q}(r)\,dr$$

One can easily show that, with the above definitions replacing $\dot{q}_r$ and s in the previous sections, the resulting adaptive controller can guarantee the global convergence of s and thus of $\dot{\tilde{q}}$, $\tilde{q}$, and $\int_0^t \tilde{q}(r)\,dr$ to zero, provided that the desired trajectories are bounded.

One notes that this adaptive controller is composed of an inverse dynamics compensation part and a PID feedback part. The inclusion of integral term in the controller may have the possibility of improving the accuracy of the controller in the presence of unmodeled low frequency disturbances. But it is not studied later because inclusion of integration has to tendency of slowing down the response of a controller.

3.7 Extension to Cartesian Space Adaptive Control

Robot control can be achieved in either joint space formulation or Cartesian space (or called operational space) formulation. There are proponents for both formulations in robotic community [see Craig, 1986; Khatib, 1987]. It is the objective of this section to show that the adaptive controller derived in section 3.2 in joint space can also be extended to Cartesian space control. This extension, besides being interesting for free-motion control tasks, might be useful in developing adaptive versions of the hybrid force-motion controller for compliant motion control.

One may define a Cartesian-space controller as a controller which achieves robot motion control by feedbacking the Cartesian-space tracking error instead of joint-space tracking errors. In this section, two ways of extending the previous

joint-space adaptive controller to Cartesian space are described, but their detailed examination is not carried out because of our emphasis on the joint space formulation. To avoid technical difficulties, we assume that the robot is non-redundant in the following.

A. Redefining the reference velocity $\dot{q}_r$

Since, as mentioned in section 3.6, $\dot{q} - \dot{q}_r$ converges to zero regardless of the definition of $\dot{q}_r$, a simple way of doing this extension is to redefine the reference velocity $\dot{q}_r$ in the basic adaptive controller in section 3.2 to be

$$\dot{q}_r = J^{-1}[\dot{x}_d - \Lambda(x - x_r)] \tag{3-37}$$

where x is the Cartesian-space position of the robot end-effector, x_d the desired Cartesian position and J(q) is Jacobian matrix between the Cartesian coordinates and the joint coordinates defined by the kinematic relation

$$\dot{x} = J\dot{q} \tag{3-38}$$

In doing so, we leave the forms of the control law (3-14) and adaptation law (3-12) unchanged. Corresponding to the above redefinition, let s still be the difference between the real joint space velocity $\dot{q}$ and the reference velocity $\dot{q}_r$, which means that

$$s = \dot{q} - \dot{q}_r = J^{-1}[\dot{\tilde{x}} + \Lambda\tilde{x}] \tag{3-39}$$

Thus s is closely related to the tracking errors in Cartesian space. Then by using the same form of Lyapunov-like function candidate as (3-8), we obtain

$$\dot{V} = -s^T K_d s = -[\dot{\tilde{x}} + \Lambda\tilde{x}]^T J^{-T} K_d J^{-1}[\dot{\tilde{x}} + \Lambda\tilde{x}] \tag{3-40}$$

Provided that J is non-singular, we obtain as before

$$\dot{\tilde{\mathbf{x}}} + \Lambda \tilde{\mathbf{x}} \rightarrow 0 \tag{3-41}$$

This implies that the Cartesian space tracking error $\tilde{\mathbf{x}} \rightarrow 0$ as $t \rightarrow \infty$. This convergence result has been indeed demonstrated in the simulation results presented in [Slotine and Li, 1986].

B. Using Cartesian-space coordinates to describe robot position

Another way of doing the extension is to start from a Cartesian-space description of robot position and generate the adaptive Cartesian-space controller in the same way as that described in section 3.2 for the adaptive joint-space controller. This means that we choose the Cartesian coordinates (*e.g.*, end-effector coordinates) $\mathbf{x}$ as the generalized coordinates for the robot dynamics. The derivation of the robot dynamics and the derivation of the adaptive controller are parallel to those for the joint space.

First, by using the kinematic relation (3-38), the kinetic energy of the robot can be expressed as

$$T = \frac{1}{2} \dot{\mathbf{x}}^T \mathbf{J}^{-1} \mathbf{H} \mathbf{J}^{-1} \dot{\mathbf{x}}$$

Using the Lagrangian equations, one obtain the dynamics of the robot in Cartesian space to be

$$\mathbf{H}_c(\mathbf{x}) \ddot{\mathbf{x}} + \mathbf{C}_c(\mathbf{x}, \dot{\mathbf{x}}) \dot{\mathbf{x}} + \mathbf{g}_c(\mathbf{x}) = \mathbf{J}^{-T} \boldsymbol{\tau} \tag{3-42}$$

where

$$\mathbf{H}_c(\mathbf{x}) = \mathbf{J}^{-T} \mathbf{H}(\mathbf{q}) \mathbf{J}^{-1}$$

$\mathbf{C}_c$ is similarly defined as $\mathbf{C}$ in (2-4) and $[\dot{\mathbf{H}}_c - 2\mathbf{C}_c]$ is similarly skew-symmetric. It is important to note that the left-hand side of (3-42) can also be

linearly parametrized, because $\mathbf{H}_c$ is linear in terms of the same inertial parameters $\mathbf{a}$ as before (noting that $\mathbf{J}$ contains only kinematic parameters which are assumed to be known).

Let us define a Cartesian input force vector $\mathbf{F}$ (an intermediate variable introduced for design purpose) to be

$$\mathbf{F} = \mathbf{J}^{-T}\boldsymbol{\tau} \tag{3-43}$$

The dynamics (3-42) are then in the same form as the joint-space dynamics (2-4) with similar properties. As a result, we can obtain the control law

$$\mathbf{F} = \hat{\mathbf{H}}_c\ddot{\mathbf{x}}_r + \hat{\mathbf{C}}_c\dot{\mathbf{x}}_r + \hat{\mathbf{g}} - \mathbf{K}_d\mathbf{s} \tag{3-44}$$

with the reference velocity now being

$$\mathbf{x}_r = \mathbf{x}_d - \boldsymbol{\Lambda}\tilde{\mathbf{x}} \tag{3-45}$$

and $\mathbf{s}$ being

$$\mathbf{s} = \dot{\mathbf{x}} - \dot{\mathbf{x}}_r = \dot{\tilde{\mathbf{x}}} + \boldsymbol{\Lambda}\tilde{\mathbf{x}} \tag{3-46}$$

The adaptation law is now

$$\dot{\hat{\mathbf{a}}} = -\boldsymbol{\Gamma}\mathbf{Y}_c^T\mathbf{s} \tag{3-47}$$

with $\mathbf{Y}_c$ defined similarly to $\mathbf{Y}$ before. By using a similar form of Lyapunov function candidate as that in (3-8), we obtain

$$\dot{V} = -\mathbf{s}^T\mathbf{K}_d\mathbf{s}$$

which implies that $\mathbf{s}\to\mathbf{0}$, $\tilde{\mathbf{x}}\to\mathbf{0}$, and $\dot{\tilde{\mathbf{x}}}\to\mathbf{0}$ as $t\to\infty$.

In implementing the adaptive controller, one computes the joint torque input by

reverting the relation (3-43), *i.e.*,

$$\tau = J^T F \tag{3-48}$$

with F defined by (3-44). The structure of the adaptive Cartesian-space controller is sketched in Figure 3-19. The PD term in the Figure is the Cartesian PD which has the form

$$-K_d s = -K_d \dot{\tilde{x}} - K_d \Lambda \tilde{x}$$

One might wonder about the relations between the two kinds of extensions. First, one note that the two schemes are different if the same gain matrix K_D is used, since the PD term in the former extension is $-K_d J^{-1}[\dot{\tilde{x}} + \Lambda \tilde{x}]$ while the PD term in the joint torque of the latter extension is $-J^T K_d[\dot{\tilde{x}} + \Lambda \tilde{x}]$. Secondly, it is easy to prove the they are identical if the gain matrix K_d in the first extension is chosen to be

$$K_{d1} = J^T K_{d2} J$$

where K_{d2} is the gain matrix in the second extension. Finally, we point out that a number of issues associated with these Cartesian-space algorithms, such as the extension to redundant manipulators, have not been addressed yet.

Comment 1: The adaptive Cartesian-space controller shown in Figure 3-19 has been implemented in the 4-DOF high performance robot at the Artificial Intelligence Laboratory of MIT, as reported by [Niemeyer and Slotine, 1989]. The computation was performed recursively and the control results were found to be desirable.

Comment 2: For manipulation tasks which require tracking of not only the end-effector position but also the end-effector orientation, the computation of the

orientation error components in $\tilde{x}$ (which is composed of end-effector position errors and orientation errors) is not a trivial issue. In computing these components, one can use the simple technique described in [Luh, 1981] and improved in [Yuan, 1988].

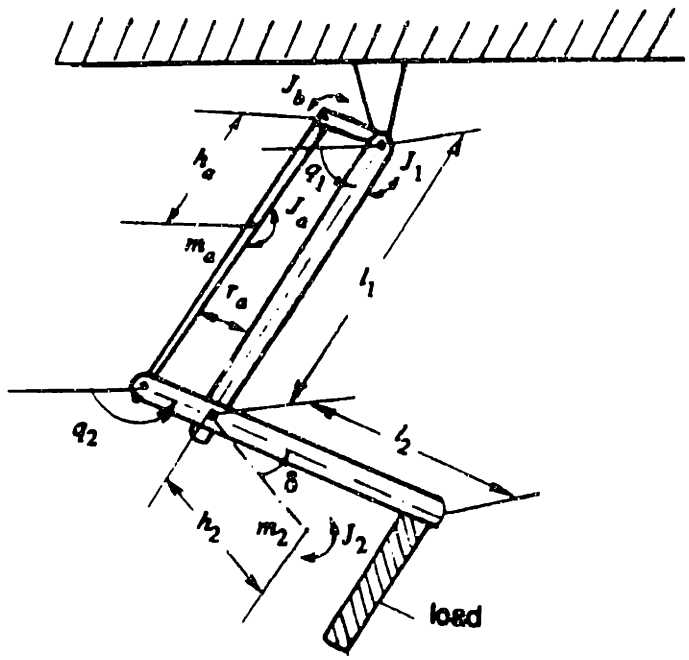


Figure 3-4: a two-link manipulator with a load

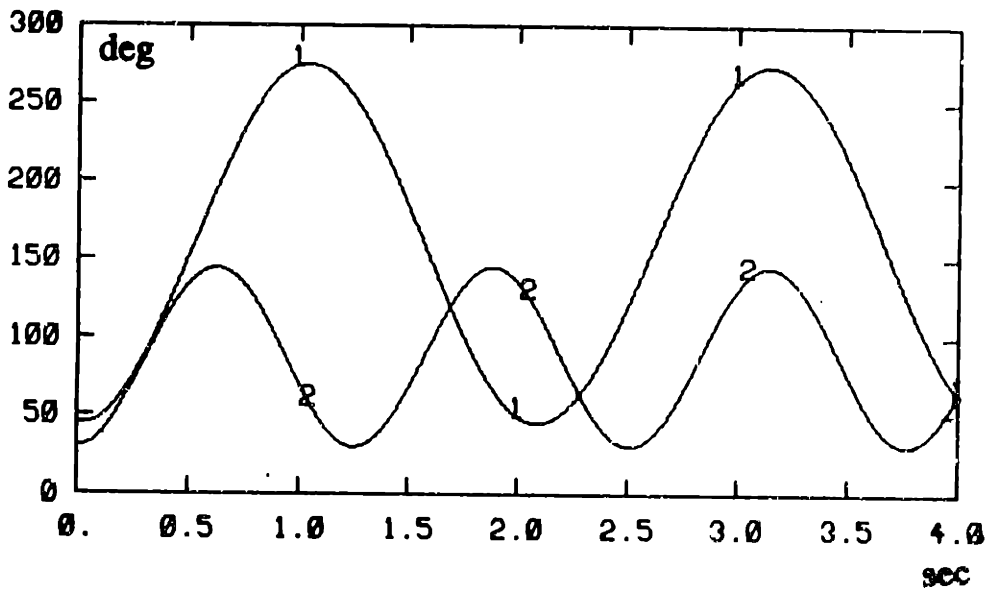


Figure 3-5: Desired motion trajectories--- sinusoid

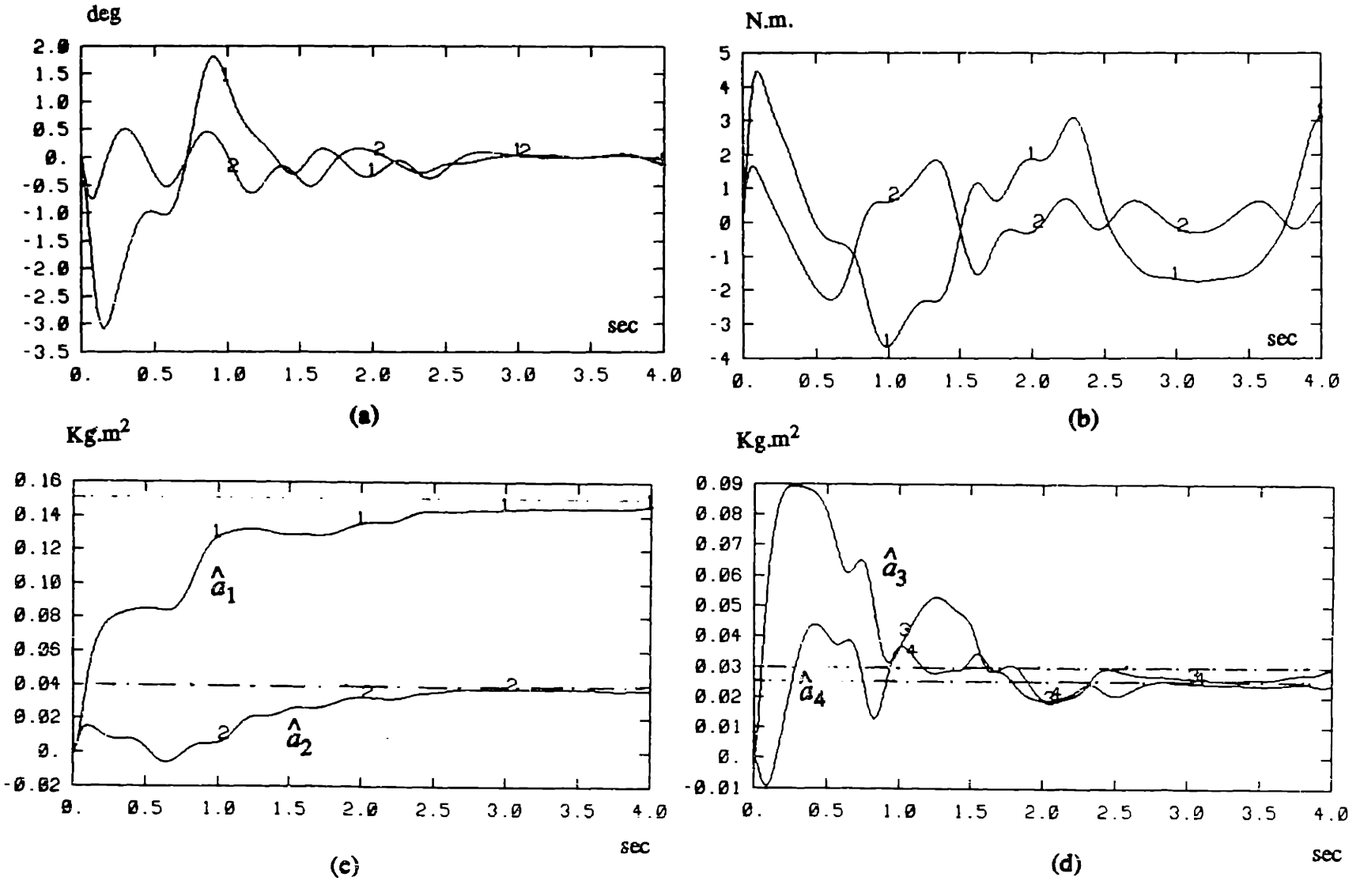
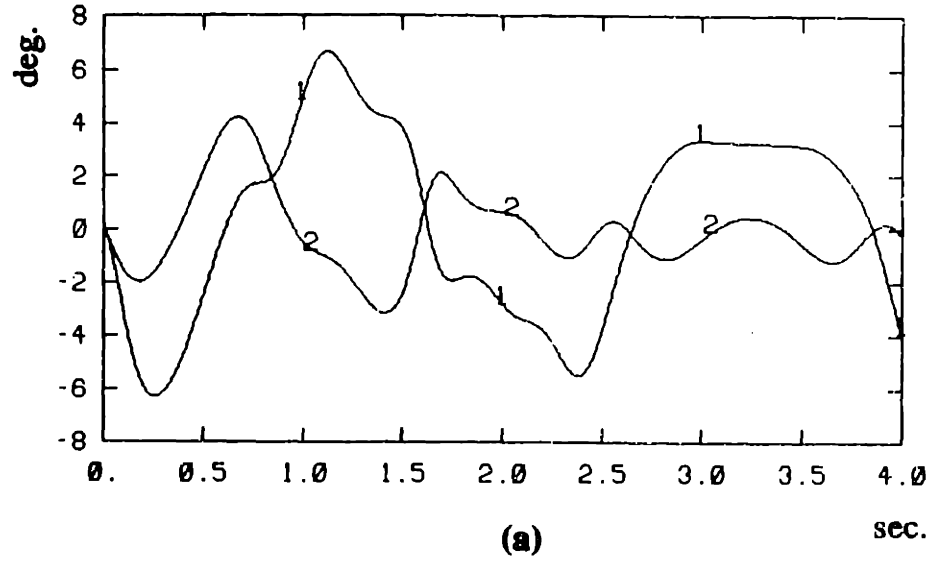
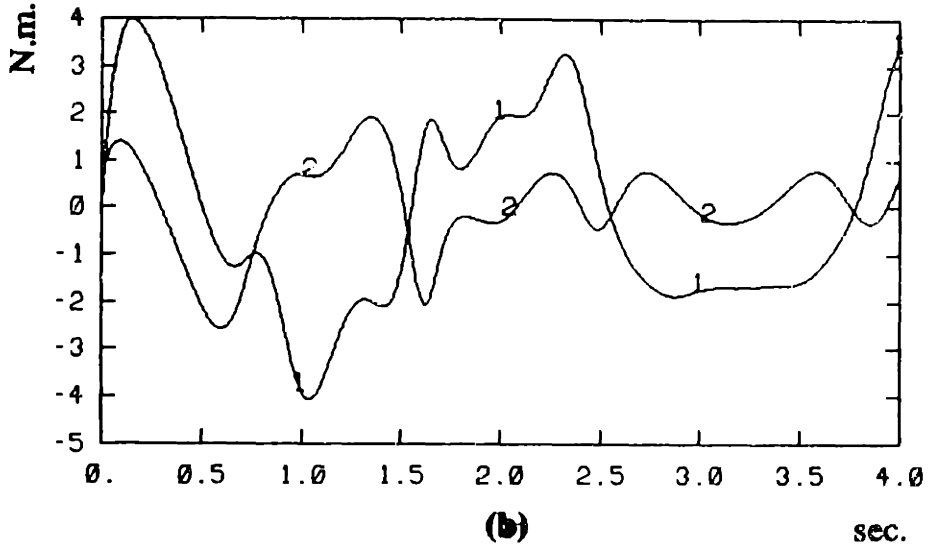


Figure 3-6: adaptive control: (a) $\tilde{q}$; (b) τ; (c) $\hat{a}_1$, $\hat{a}_2$; (d) $\hat{a}_3$, $\hat{a}_4$



(a)



(b)

Figure 3-7: PD control: (a) $\tilde{\mathbf{q}}$; (b) τ

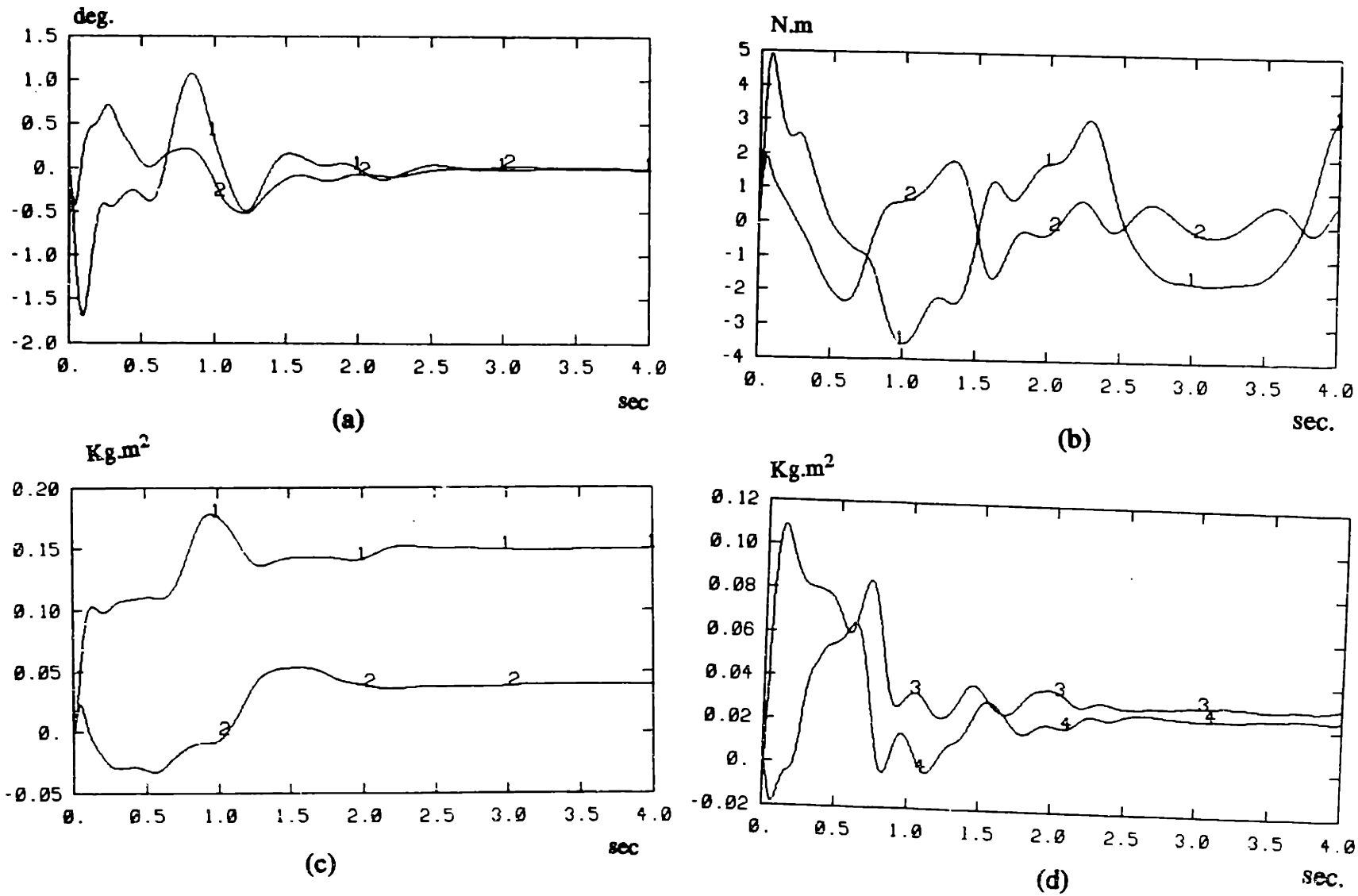


Figure 3-8: Adaptation Gain $\gamma = 0.1$: (a) $\tilde{\mathbf{q}}$; (b) τ; (c) $\hat{a}_1, \hat{a}_2$; (d) $\hat{a}_3, \hat{a}_4$

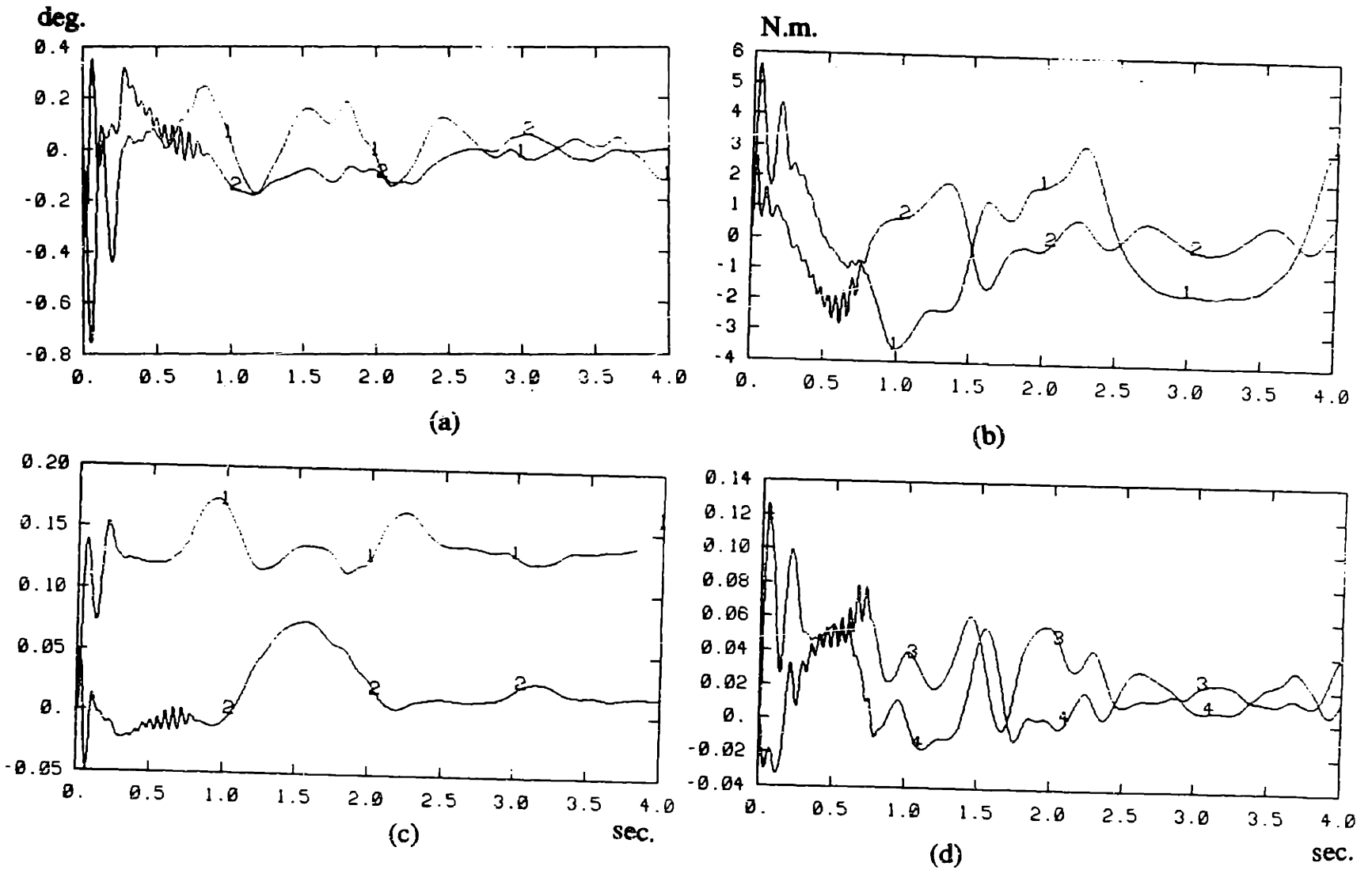


Figure 3-9: Adaptation Gain $\gamma = 0.4$: (a) $\tilde{\mathbf{q}}$; (b) τ; (c) $\hat{a}_1, \hat{a}_2$; (d) $\hat{a}_3, \hat{a}_4$

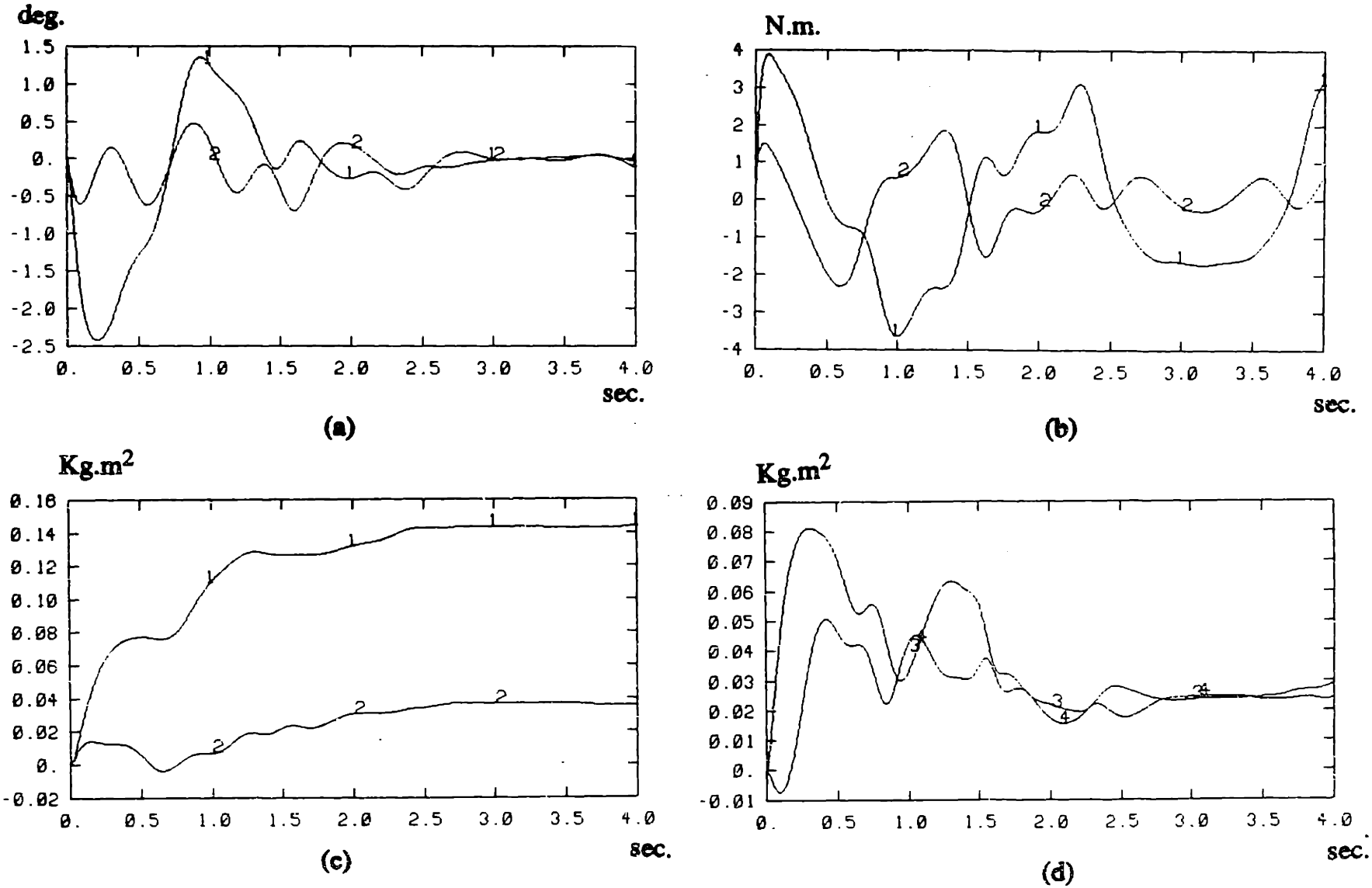


Figure 3-10: Adaptive Control k_d=6, λ=5 (*a*) $\tilde{\mathbf{q}}$; (*b*) τ; (*c*) $\hat{a}_1, \hat{a}_2$; (*d*) $\hat{a}_3, \hat{a}_4$

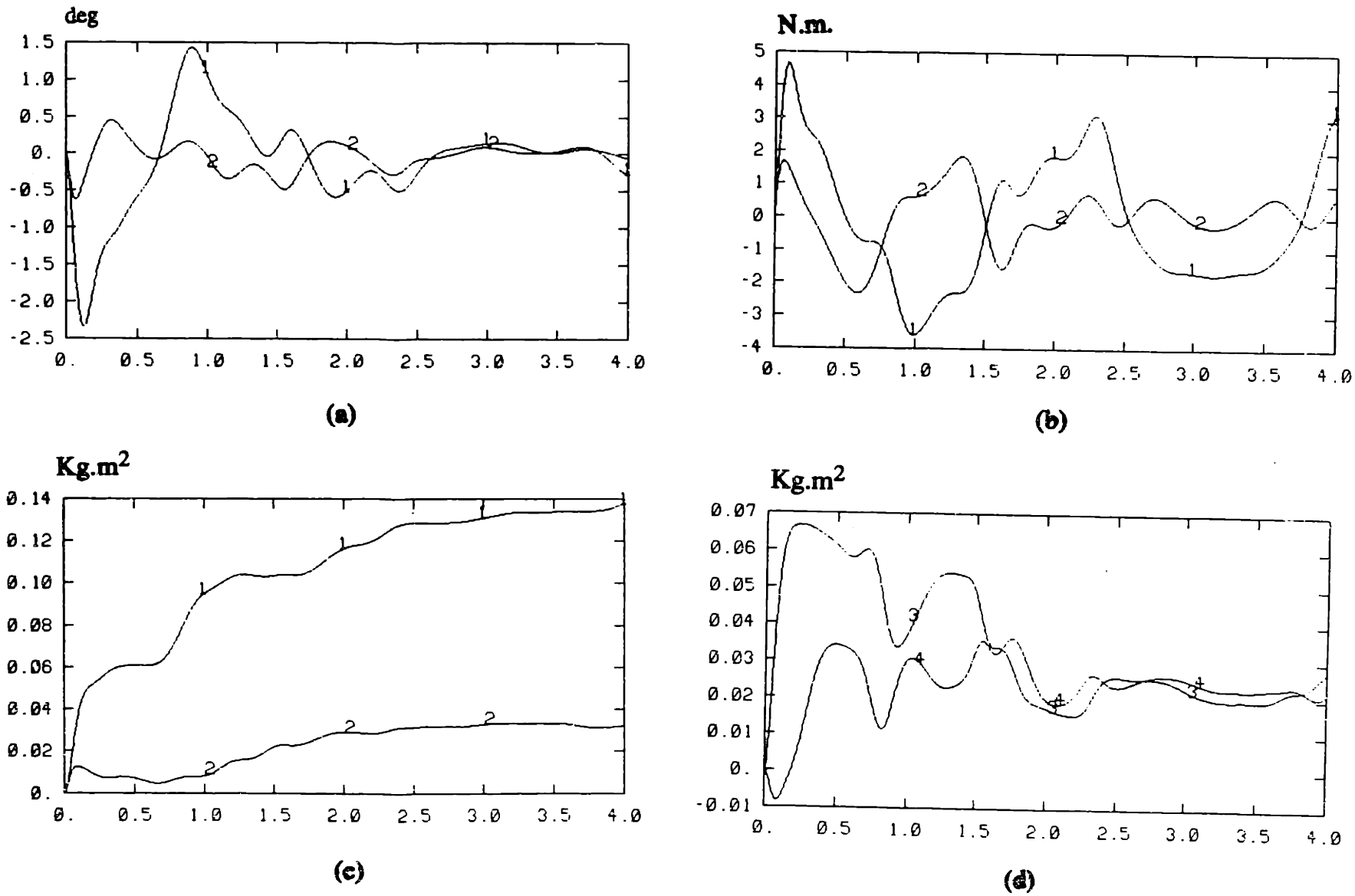


Figure 3-11: Adaptive Control $\lambda = 20$ (a) $\tilde{\mathbf{q}}$; (b) τ; (c) $\hat{a}_1$, $\hat{a}_2$; (d) $\hat{a}_3$, $\hat{a}_4$

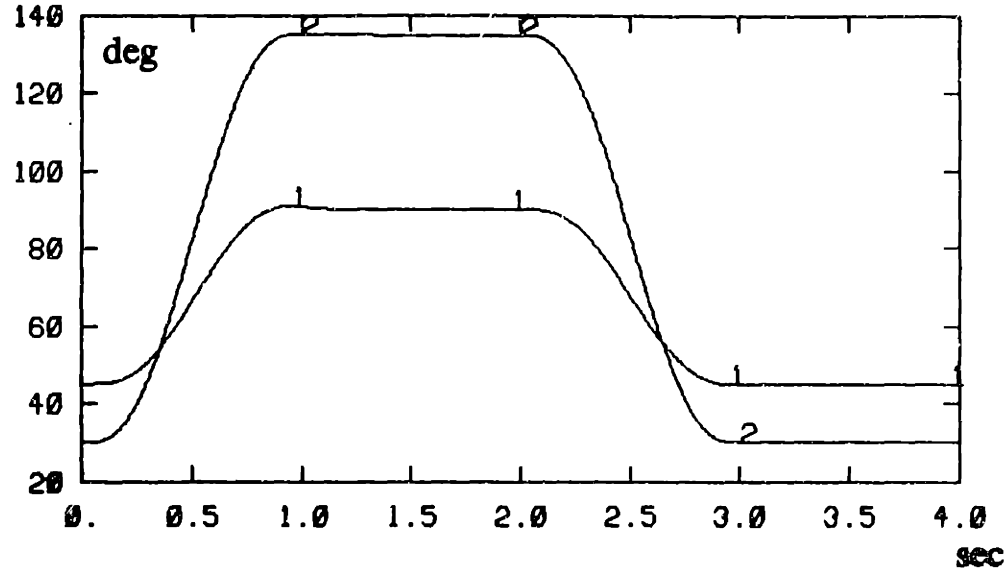


Figure 3-12: desired trajectories --- polynomials plus rest

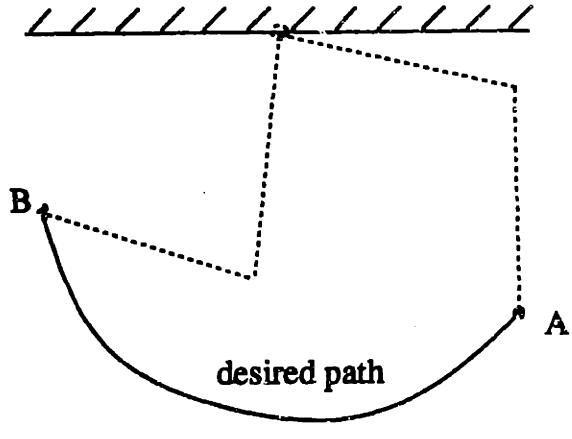


Figure 3-13: desired trajectories --- polynomials plus rests

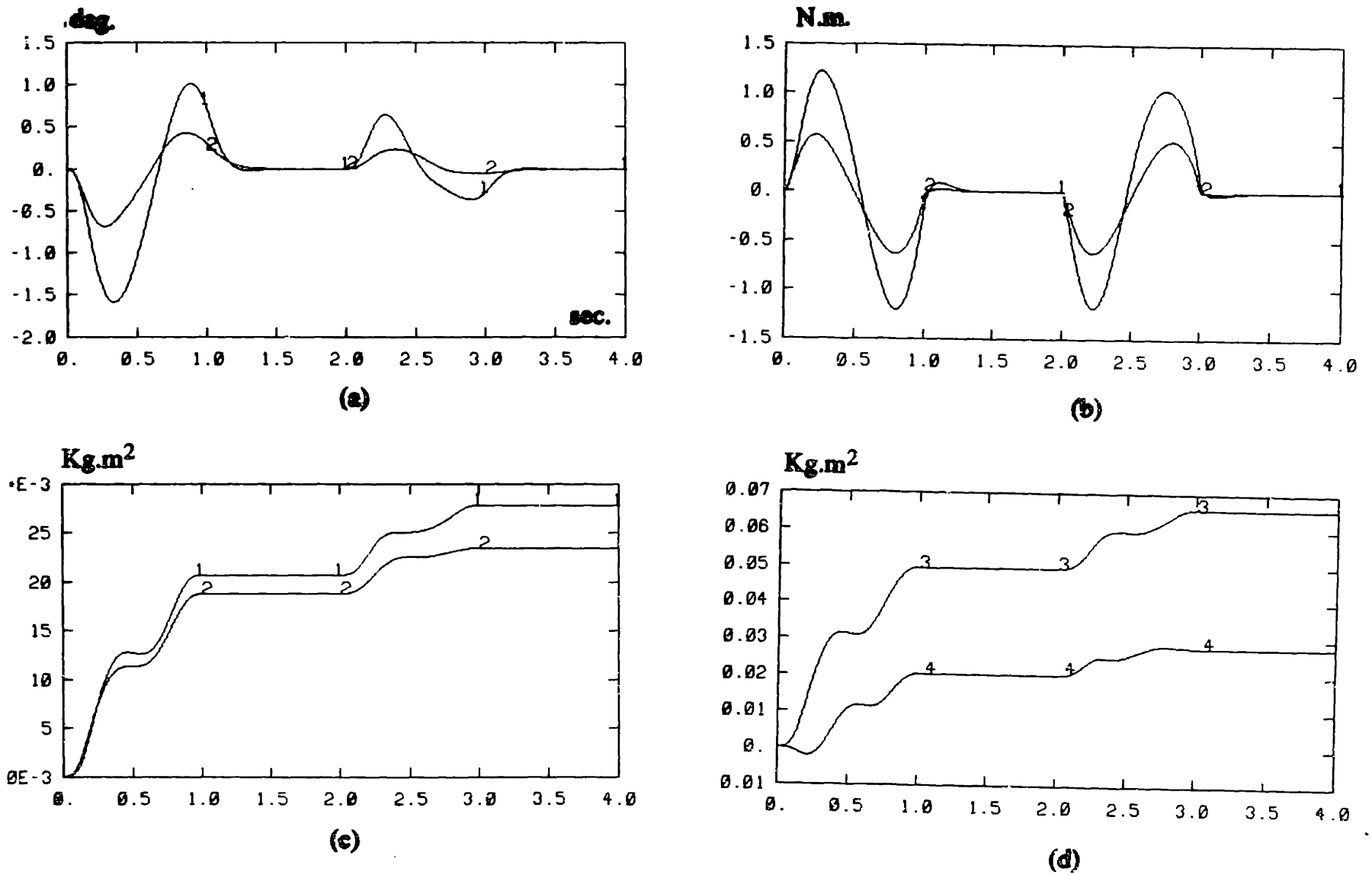


Figure 3-14: Adaptive Control on Polynomial+Rest Trajectories (1st 4 seconds)
(*a*) $\tilde{q}$; (*b*) τ; (*c*) $\hat{a}_1, \hat{a}_2$; (*d*) $\hat{a}_3, \hat{a}_4$

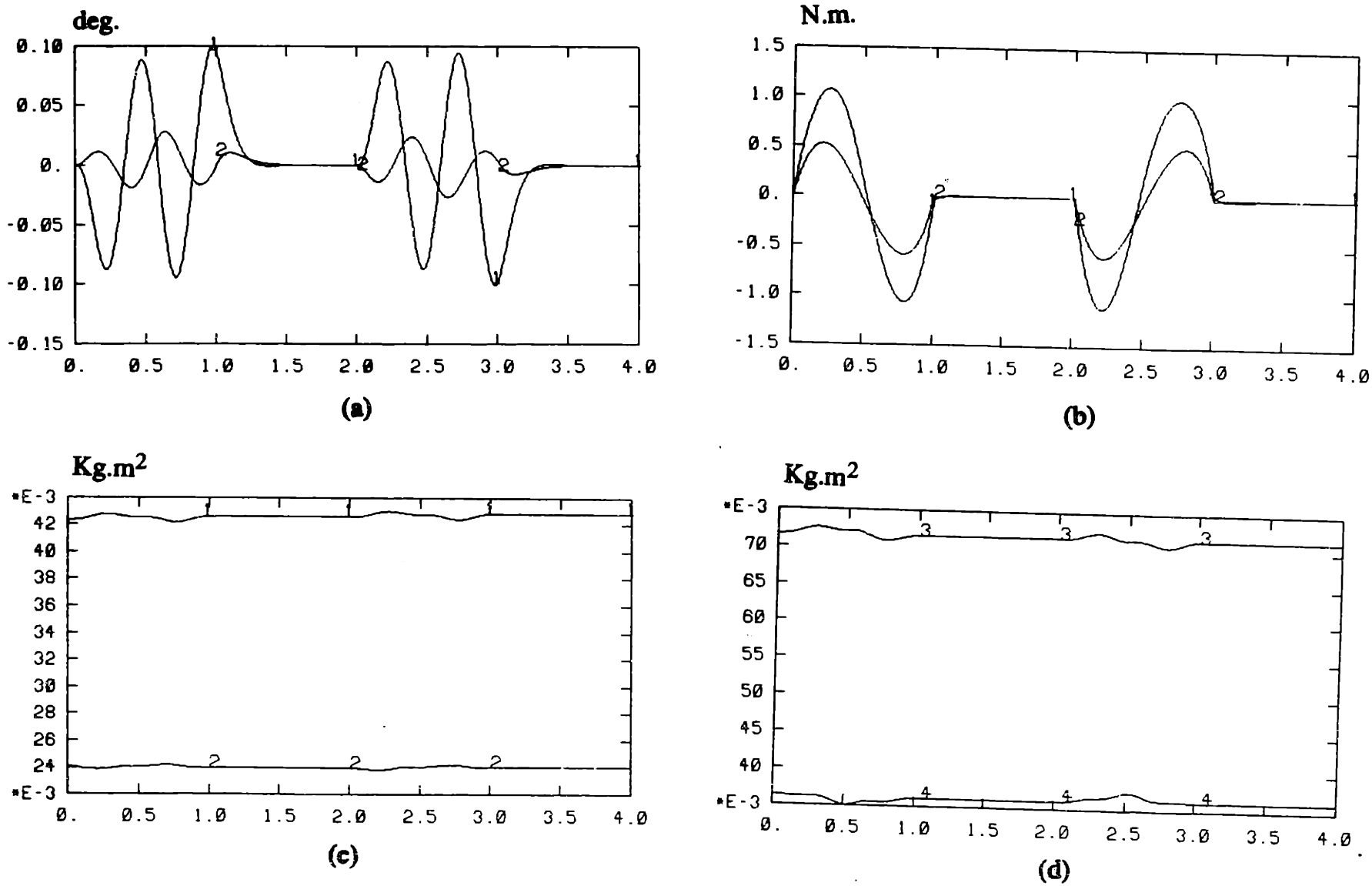


Figure 3-15: Adaptive Control on Polynomial+Rest Trajectories (last 4 seconds)
(a) $\dot{q}$; (b) τ; (c) $\hat{a}_1, \hat{a}_2$; (d) $\hat{a}_3, \hat{a}_4$

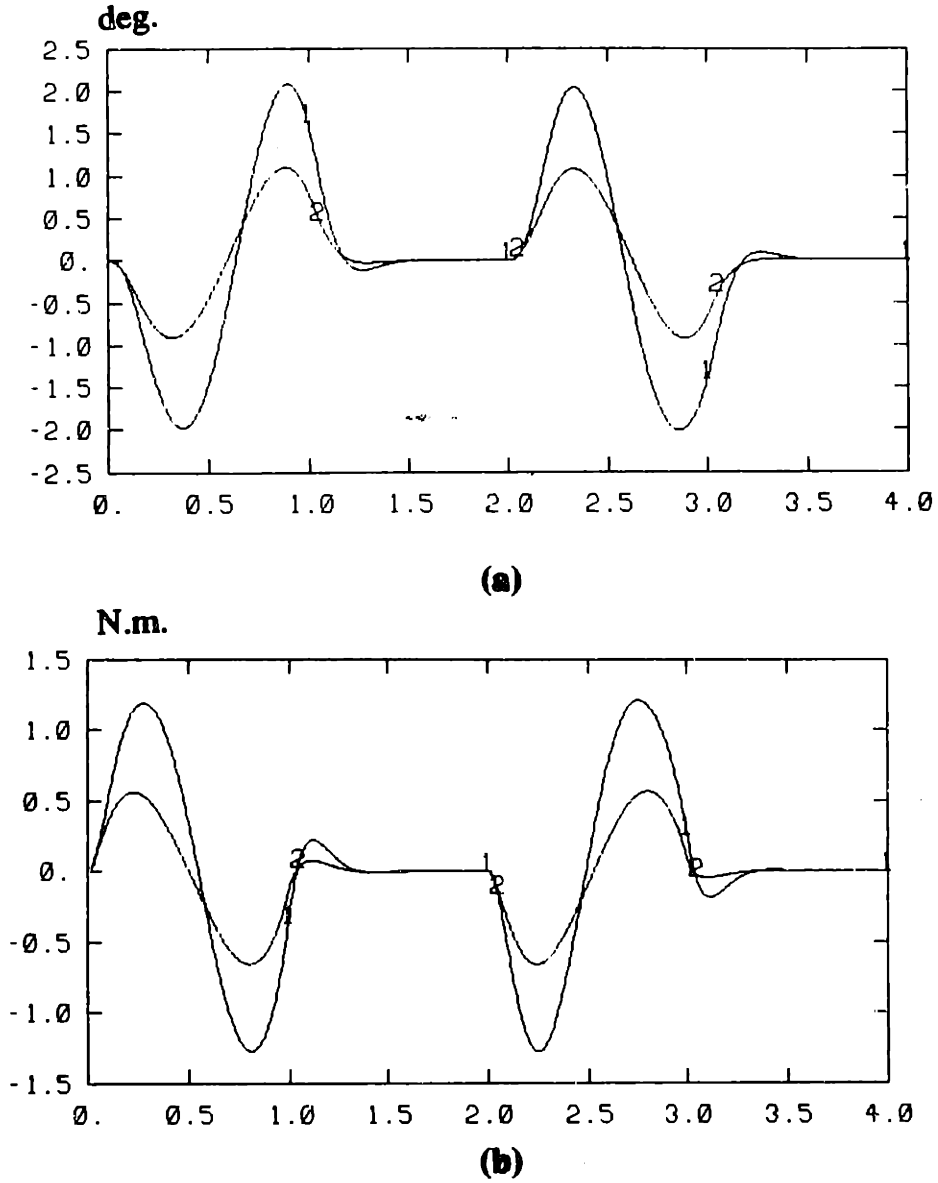


Figure 3-16: PD on polynomial+rest trajectories (a) $\tilde{q}$; (b) τ

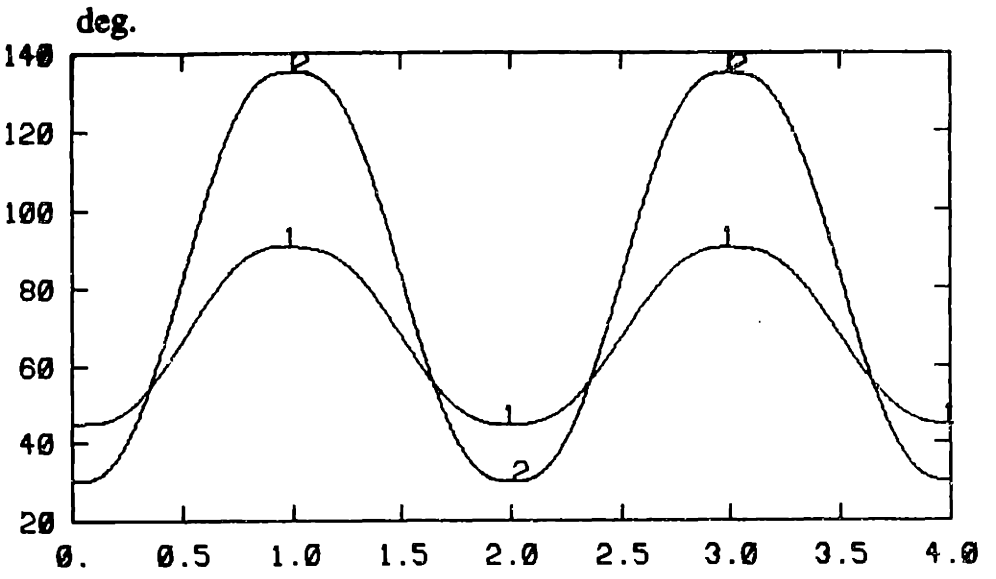


Figure 3-17: desired trajectories --- polynomials without rests

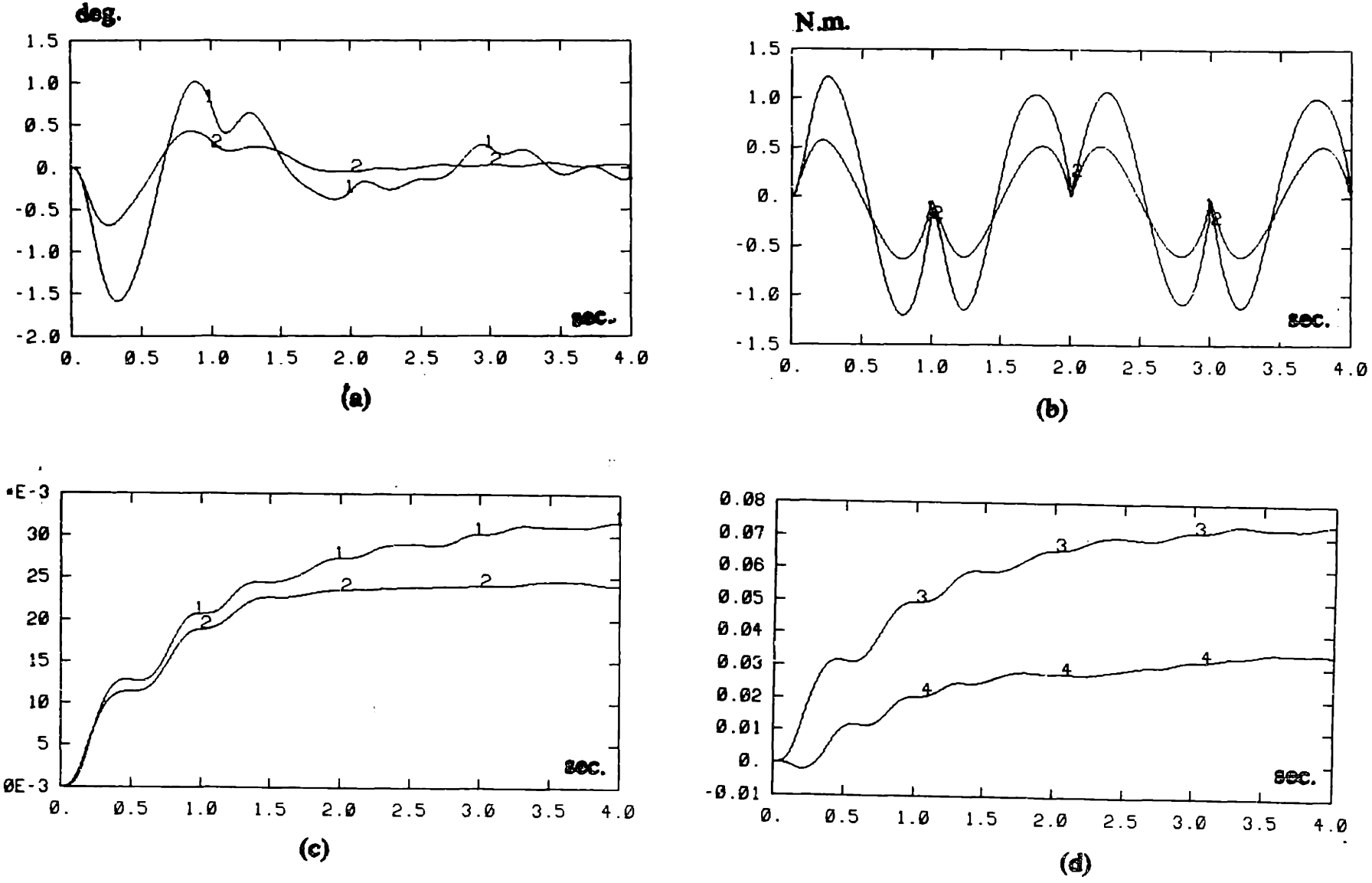


Figure 3-18: Adaptive Control on Polynomial Trajectories (1st 4 seconds)
(*a*) $\tilde{q}$; (*b*) τ; (*c*) $\hat{a}_1, \hat{a}_2$; (*d*) $\hat{a}_3, \hat{a}_4$

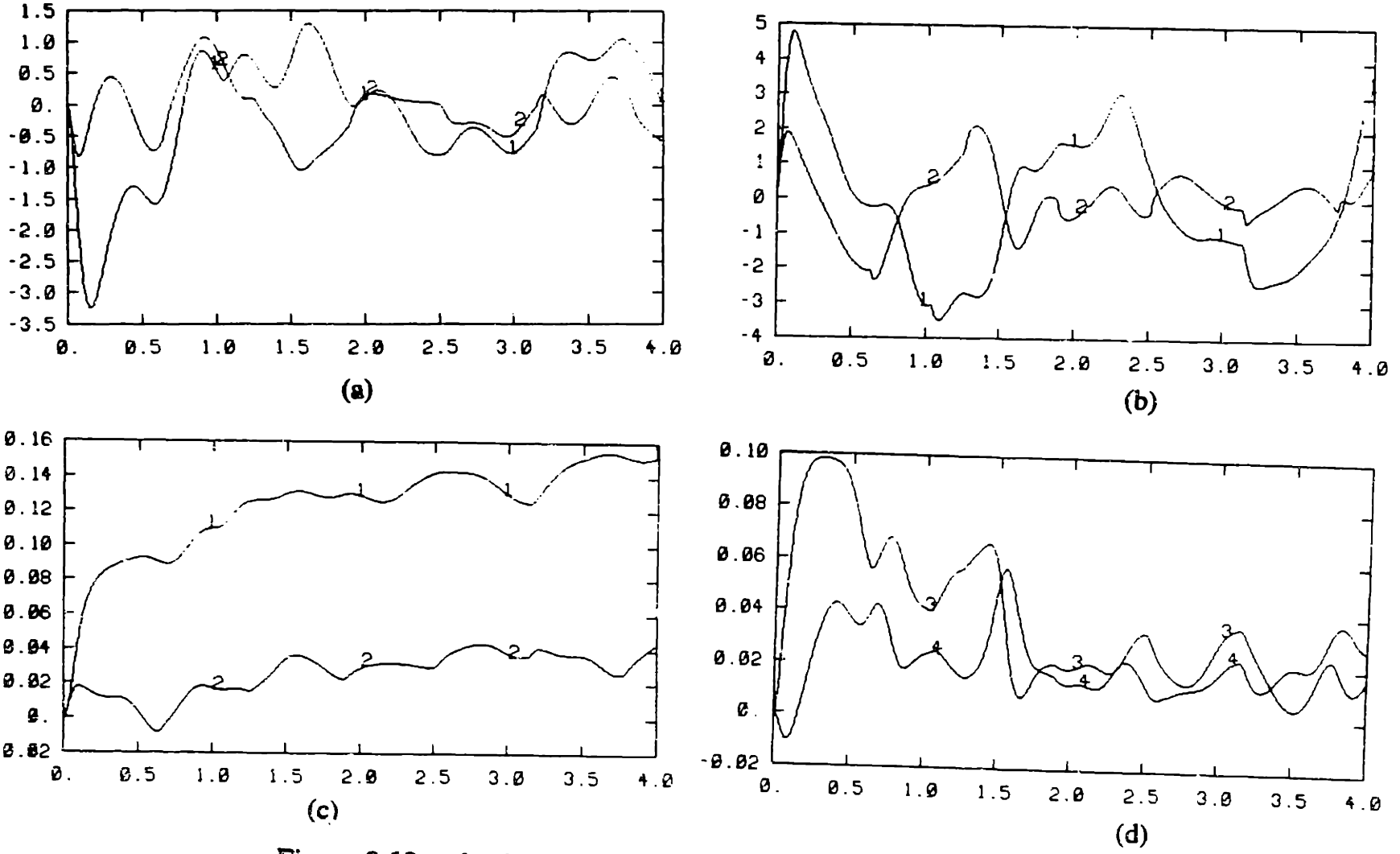


Figure 3-19: adaptive control on *sinusoidal-type* trajectories: with disturbance

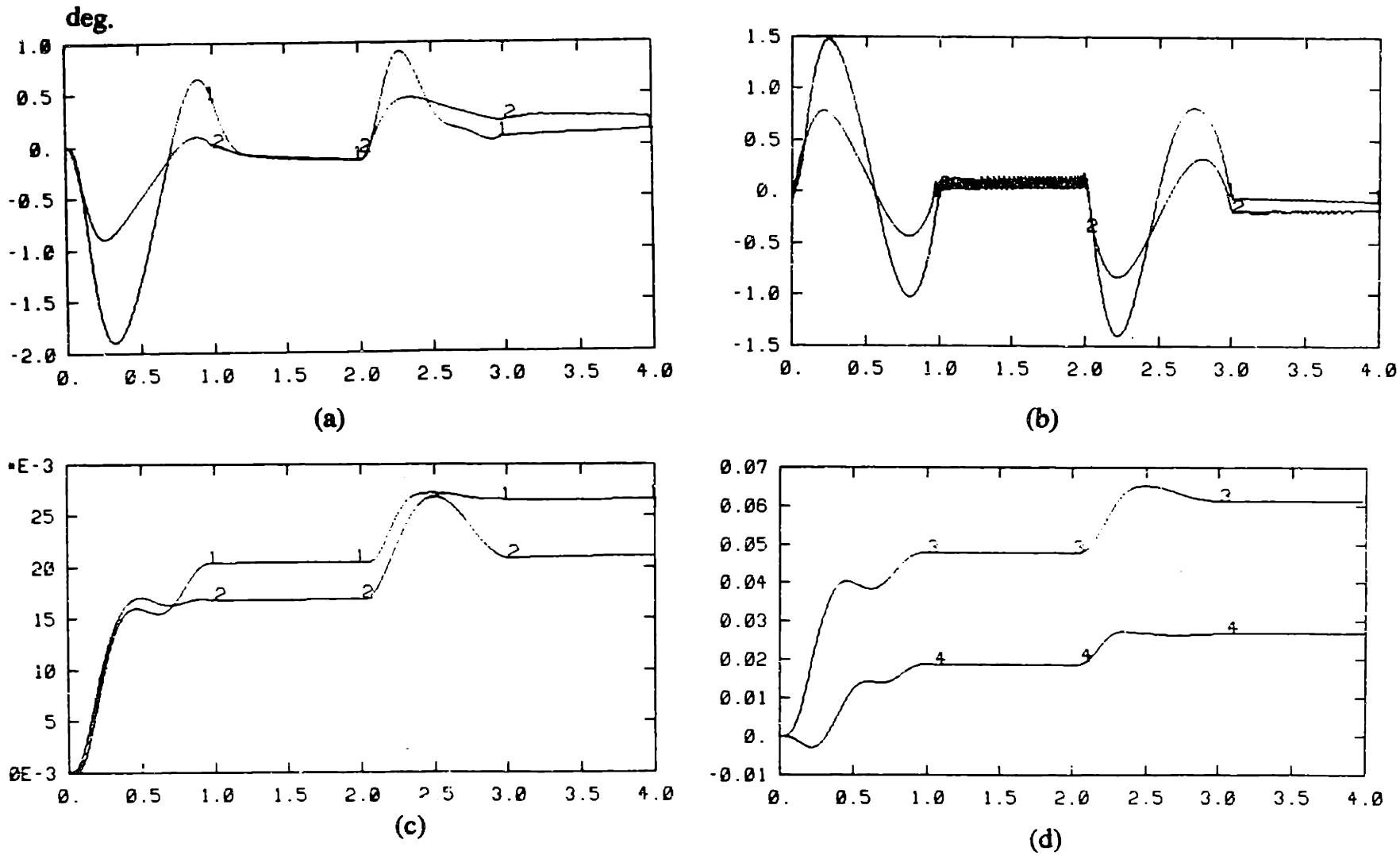


Figure 3-20: adaptive control on *polynomial-type* trajectories: with disturbance

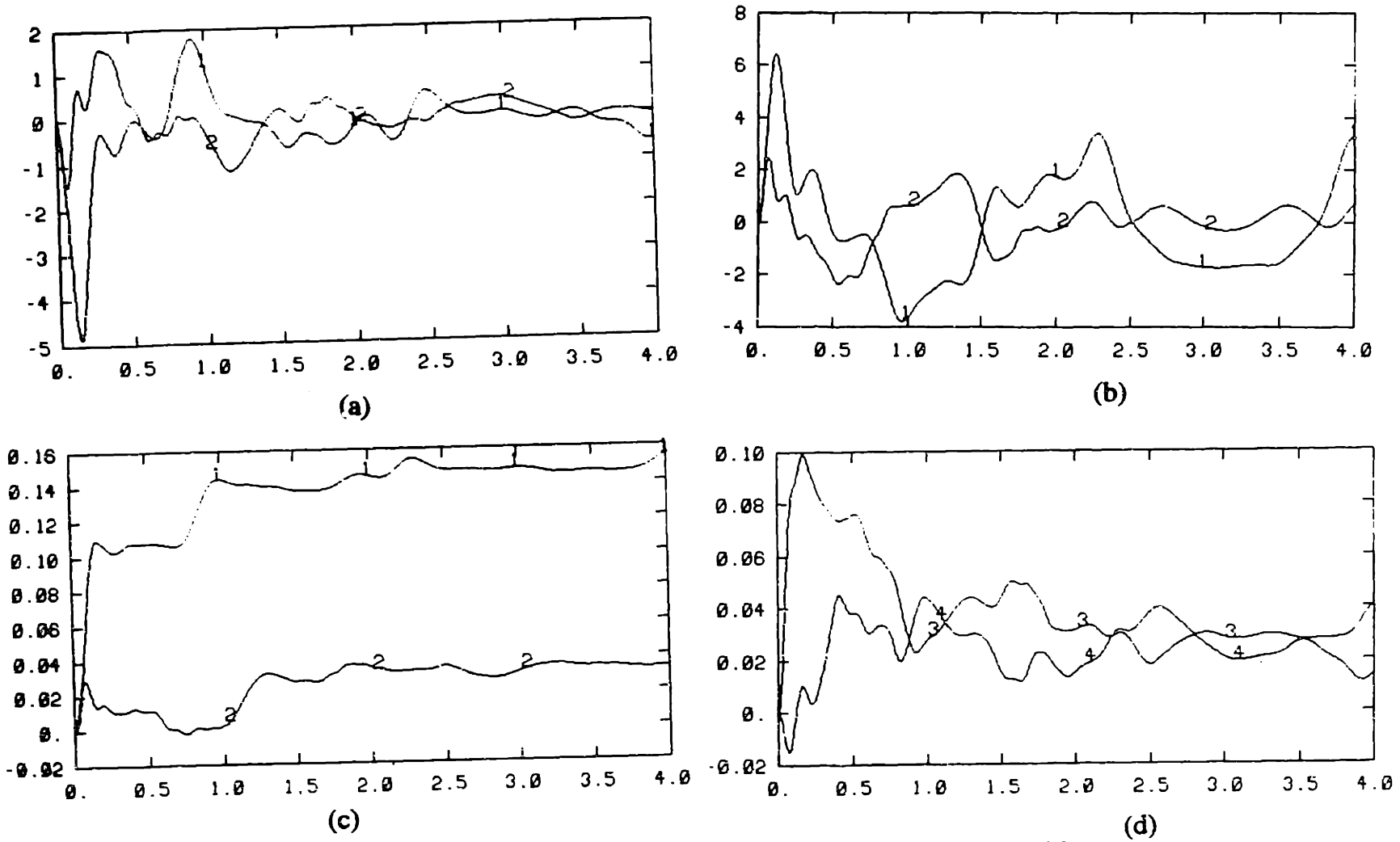


Figure 3-21: adaptive control on *sinusoidal-type* trajectories: with motor dynamics

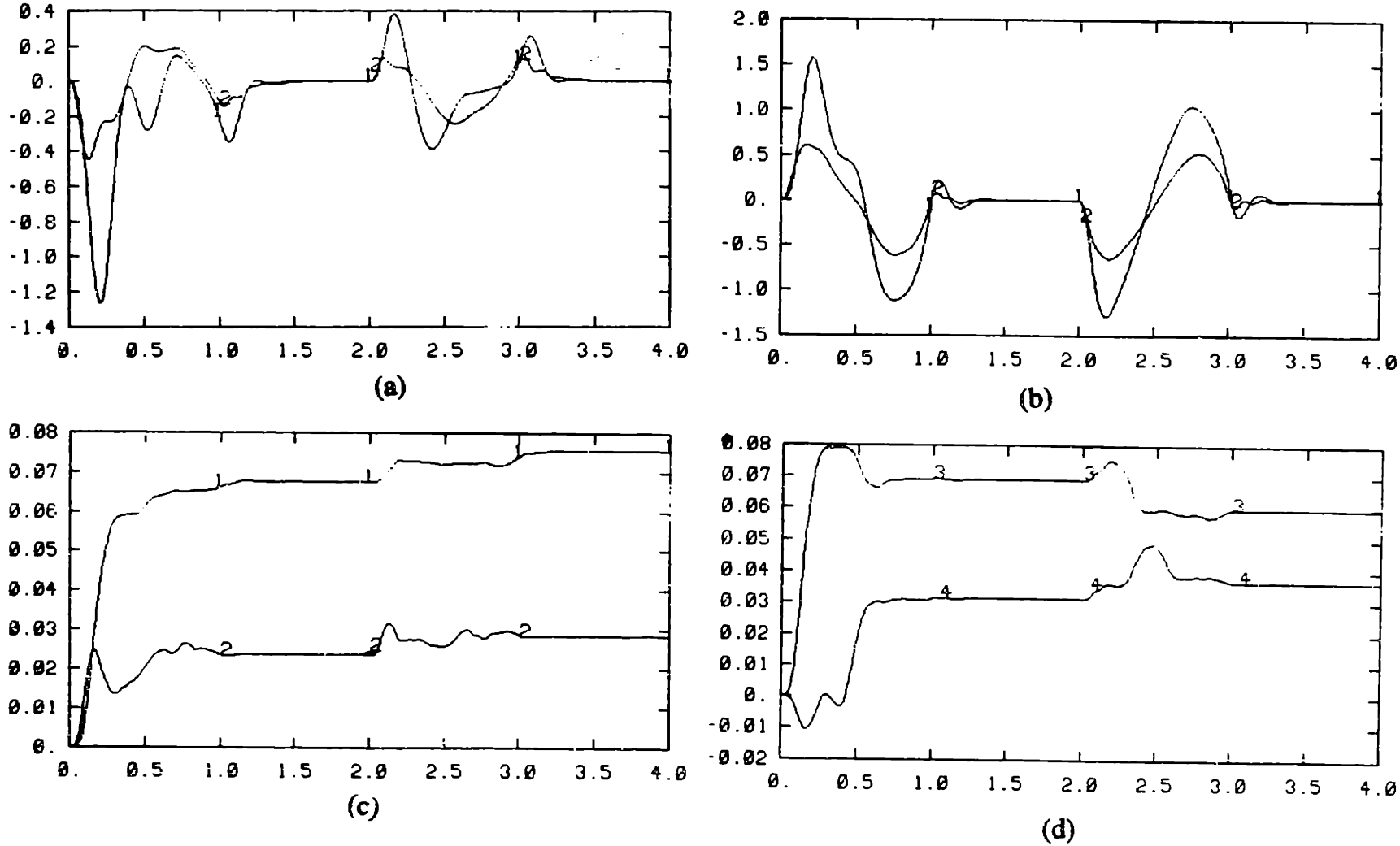
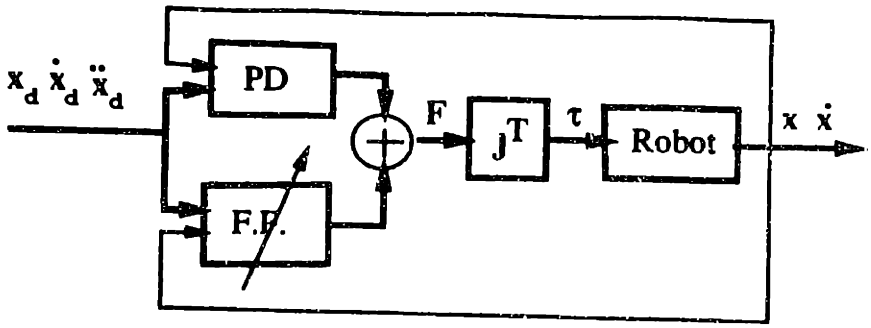


Figure 3-22. adaptive control on *polynomial-type* trajectories: with motor dynamics



PD: $-K_d s_c$ F.F.: $\hat{H}_c \ddot{x}_r + \hat{C}_c \dot{x}_r + \hat{g}_c$

Figure 3-23: Cartesian-Space Adaptive Control

Chapter 4
Parameter Convergence

Simulation results in chapter 3 indicate that the the estimated parameters in the adaptive control system converge for some desired trajectories but not for others. This raises the question of when exactly the estimated parameters converge to the true parameters. It is the objective of this chapter to study the analytical conditions which the desired trajectories have to satisfy in order for the estimated parameters to converge.

Although we are most concerned about the accuracy of tracking errors in tracking tasks, it is still of considerable interest to study the convergence issue associated with the estimated parameters. On the theoretical side, studying the convergence of estimated parameters allows us to gain insights about the functioning of the adaptive control system. On the practical side, convergence of estimated parameters has desirable implications on tracking accuracy, because it leads to reduced tracking errors in the long run and to system robustness with

respect to noise. The practical implications of parameter convergence will be explained further in section 4.4.

This chapter is organized as follows. In the first section, we discuss the concepts of sufficient richness and persistent excitation which will be used to characterize the kind of trajectories which can lead to parameter convergence. In section 4.2, the conditions of parameter convergence are established. Section 4.3 discusses the practical implications of the persistent excitation condition, *i.e.*, how persistent excitation of desired trajectories affect the accuracy, robustness and other aspects of the adaptive control operation. In view of the useful consequences of persistent excitation, section 4.4 provides a discussion on how to generate persistently exciting trajectories for adaptive robot control.

4.1 Sufficient Richness and Persistent Excitation

In this section, we discuss the concepts of sufficient richness and persistent excitation, to prepare for the rigorous proof of parameter convergence under these conditions. Before we give precise definitions of these concepts, let us provide the intuitive motivation for such definitions.

4.1.1 Trajectory Excitation and Parameter Convergence

It is interesting to see why the convergence of estimated parameters is dependent on the features of the desired trajectories. To do this, we note that the adaptation law in the adaptive robot controller adjusts the variable parameters in the control law to reduce the tracking errors. Since this adjustment is done without direct concern for the convergence of the variable parameters to the true parameters, it may or may not lead to parameter convergence. More specifically, if the desired motion trajectories are very simple in excitation (the word

"excitation" is used because the desired motion trajectories represent a source of motion for the otherwise immobile robot) so that many possible values of the variable parameters allow the tracking errors to converge, the parameter adaptation may eventually drive the estimated parameters to values other than the true parameters. If the desired trajectories are so rich in excitation (*i.e.*, so complex) that only the true parameters can result in tracking error convergence top zero, the parameter adaptation law will drive the estimated parameters to the true parameters to lead to tracking convergence (note that true parameters can guarantee perfect tracking for any desired trajectories). Now let us see such relations between the trajectory excitation and parameter convergence on the one-link robot in section 2.4.

With the robot dynamics given by (2-11) and the control law by (2-15), the closed-loop dynamics of can be written as (2-17), or more explicitly as

$$m\ddot{\tilde{q}} + (\lambda\hat{m} + k_d)\dot{\tilde{q}} + k_d\lambda\tilde{q} = \tilde{m}\ddot{q}_d \qquad (4\text{–}1)$$

First let us consider the case when the controller parameter $\hat{m}$ is of constant value. A few of observations can be made about the tracking convergence condition for the dynamics (4-1):

- *For any bounded desired acceleration*, the tracking error s converges to zero if the controller parameter is the true parameter m, *i.e.*, $\hat{m}$ = m;

- *If the desired acceleration* $\ddot{q}_d(t)$ *is zero or converges to zero*, the tracking error s will converge to zero for any value of $\hat{m}$ which satisfies

$$\lambda\hat{m} + k_d > 0 \qquad (4\text{–}2)$$

Obviously, any positive constant can satisfy (4-2). Furthermore, if the desired acceleration is zero and initial tracking errors are zero, perfect tracking ($\tilde{q}(t) = 0$)

will result for any of such parameter values.

• *If the desired acceleration is not convergent to zero and is smooth*, the tracking error will converge to zero only if the controller parameter is the true parameter, *i.e.*, $\hat{m} = m$. Note that a certain smoothness is required on $\ddot{q}_d$ to draw the above conclusion, because (4-1) can be regarded as a filter which can filter out the high frequency components in the input $\tilde{m}\ddot{q}_d$. The property that the desired acceleration does not converge to zero will later be more accurately described as sufficient richness or persistent excitation. The smoothness can be characterized by uniform continuity.

Now let us consider the case when the true parameter m is unknown and $\hat{m}$ is adjusted by the adaptation law. Since the tracking error of the adaptive control system converges to zero, the parameter adaptation rate converges to zero. This implies that after some time, the estimated parameter $\hat{m}$ will be essentially constant. Using observations similar to the above ones, we expect the estimated parameters must converge to the true parameter m if the desired acceleration is not convergent and is smooth, because otherwise the tracking error will not be essentially zero in the long run. If the desired acceleration converges to zero, we cannot expect the estimated parameter $\hat{m}$ to converge to the true parameter, because values other than the true parameter value can also maintain the tracking error to be essentially zero.

In the general case with multiple robot links, the nonlinearity of robot dynamics and the coupling of the joints introduce a lot complexity into the above kind of reasoning. Despite the complexity, the basic relation between the desired trajectory excitation and parameter convergence is the same. However, in order the precisely characterize such relations for multi-link robots, we need concepts

like sufficient richness and persistent excitation.

4.1.2 The Concept of Sufficient Richness (SR)

The definition of sufficient richness below is similar to that in [Morgan and Narendra, 1977].

Definition 4.1: *A time-varying matrix* $M(t)$ *is sufficiently rich (SR) if there are positive numbers* T_0, ε_0, and δ_0 *such that for any* $t_1 \geq 0$ *and any unit vector* $w \in R^m$, *there exists* $t_2 \in [t_1, t_1+T_0]$ *satisfying*

$$\left\| \int_{t_2}^{t_2+\delta_0} M(t)\, w\, dt \right\| \geq \varepsilon_0 \qquad (4\text{–}3)$$

Intuitively, the sufficient richness implies that, in an integral sense, the magnitude of the matrix Y is larger than a positive number *in any direction*. The relation between the various times is shown in Figure 4-1.

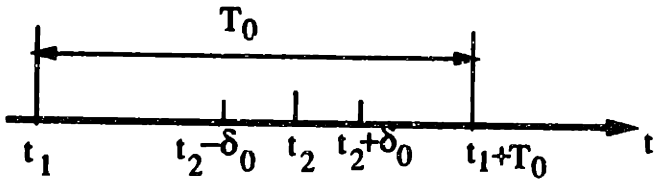


Figure 4-1: Times in Sufficient Richness Condition

The parameter convergence condition for our adaptive controller will be shown to be the sufficient richness of the matrix $Y(q, \dot{q}, \dot{q}_r, \ddot{q}_r)$. However, this condition is not convenient because $q(t)$ and $\dot{q}(t)$ are generated on-line and therefore there is no way of checking whether the estimated parameters will be convergent before the control operation starts. It is therefore more desirable to express the sufficient

richness condition in terms of the desired trajectories. The following lemma is useful for this purpose.

Lemma 4.1: *With* $\tilde{q} \to 0$ *and* $\dot{\tilde{q}} \to 0$, *the sufficient richness of* $Y(q, \dot{q}, \dot{q}_r, \ddot{q}_r)$ *is equivalent to the sufficient richness of* $Y(q_d, \dot{q}_d, \dot{q}_d, \ddot{q}_d)$

where $Y(q_d, \dot{q}_d, \dot{q}_d, \ddot{q}_d)$ is obtained by substituting the desired trajectories into the matrix Y. This substitution leads to the peculiar double appearance of desired acceleration $\ddot{q}_d$ the above notation.

Proof: The proof can be made based on the intuitive understanding that the tracking error convergence implies that q and $\dot{q}$ will be essentially the same as q_d and $\dot{q}_d$ after some time. For convenience, we shall denote $Y(q_d, \dot{q}_d, \dot{q}_d, \ddot{q}_d)$ by Y_d. To stress the fact that Y and Y_d are (implicit) functions of time, we shall often write them as Y(t) and $Y_d(t)$.

We show below that the sufficient richness of Y_d implies that of Y. The converse property can be derived using the same procedure.

From (3-1), (3-2), and $\tilde{q} \to 0$ and $\dot{\tilde{q}} \to 0$, we have $\dot{q}_r \to \dot{q}_d$ and $\ddot{q}_r \to \ddot{q}_d$. Therefore,

$$\| Y(q, \dot{q}, \dot{q}_r, \ddot{q}_r) - Y(q_d, \dot{q}_d, \dot{q}_d, \ddot{q}_d) \| \to 0 \quad \text{as } t \to \infty \tag{4-4}$$

Let ε_0 and δ_0 and T_0 be the positive constants required in the sufficient richness definition for Y, *i.e.*,

$$\| \int_{t_2}^{t_2+\delta_0} Y_d w \, dt \| \geq \varepsilon_0 \tag{4-5}$$

Then, given the positive number $\varepsilon_0/(2\,\delta)$, (4-4) implies that there exists an integer n_0, such that $\forall t > n_0 \delta$

$$\| Y - Y_d \| < \frac{\varepsilon_0}{2\delta_0} \tag{4-6}$$

Now take $T_0' = T_0 + n_0 T_0$. Then for any positive t_1 and any unit vector $w \in \mathbb{R}^m$, (4-5) and (4-6) implies that there exists $t_2' \in [t_1, t_1 + T_0']$ such that

$$\| \int_{t_2'}^{t_2'+\delta_0} Y w \, dt \| \geq \| \int_{t_2'}^{t_2'+\delta_0} Y_d w \, dt \| - \int_{t_2'}^{t_2'+\delta_0} \| (Y - Y_d) w \| dt$$

$$\geq \varepsilon_0 - \frac{\varepsilon_0}{2\delta_0}\delta_0 = \frac{\varepsilon_0}{2}$$

where t_2' is taken to be the $t_2 \in [(t_1 + n_0 T_0), (t_1 + n_0 T_0) + T_0]$ corresponding to the sufficient richness condition of $Y(q_d, \dot{q}_d, \ddot{q}_d, \dddot{q}_d)$. This shows the sufficient richness of $Y(q, \dot{q}, \dot{q}_r, \ddot{q}_r)$.

□

4.1.3 The Concept of Persistent Excitation (PE)

Given a time-varying function M(t), it is not very easy to check whether it is indeed persistently exciting because (4-3) involve arbitrary unit vectors w. A more convenient condition, which also characterizes the property that a function is non-zero in an integral sense, is the so called persistent excitation condition.

Definition 4.2: *If there exist positive constants* δ, α_1 and α_2, *such that for all* $t_1 \geq 0$

$$\alpha_1 I \leq \int_{t_1}^{t_1+\delta} M^T(t) M(t) \, dt \leq \alpha_2 I \tag{4-7}$$

the matrix M *is said to be* <u>persistently exciting</u> *(PE).*

The left-hand-side inequality indicates the smallest magnitude of the matrix

among all directions while the right-hand-side indicates the boundedness of the matrix.

4.1.4 Relation Between SR and PE

The persistent excitation condition is closely related to the sufficient richness condition, as reflected by the following lemma:

Lemma 4.2: *If a matrix* $M(t)$ *is uniformly continuous (with respect to time), and persistently exciting, it is sufficiently rich.*

Proof: From (4-5), for any unit vector w,

$$\alpha_1 \le \int_{t_1}^{t_1+\delta} \| M(t) w \|^2 dt \le \alpha_2 \qquad (4\text{–}8)$$

Based on this relation, we can now straightforwardly show that (4-3) is satisfied.

First, we note that the uniform continuity and the persistent excitation of M implies the boundedness of M, because otherwise one can show that the right-hand-side inequality in (4-8) would be violated. Due to this boundedness,

$$\| M(t) w \| \le M_0$$

$\forall t \ge 0$, where M_0 is a positive constant. Therefore, the left-hand side of (4-8),

$$\int_{t_1}^{t_1+\delta} \| M(t) w \| dt \ge \frac{\alpha_1}{M_0}$$

This indicates that the average value of the scalar function $\|M w\|$ over the δ interval is larger than $\alpha_1/(M_0\delta)$, *i.e.*,

$$\frac{1}{\delta}\int_{t_1}^{t_1+\delta} \| \mathrm{M}(t)\,\mathrm{w} \|\, dt \geq \frac{\alpha_1}{\delta M_0}$$

Therefore $\exists t_2 \in [t_1, t_1+\delta]$ such that

$$\| \mathrm{M}(t_2)\,\mathrm{w} \| \geq \frac{\alpha_1}{M_0 \delta}$$

Now let us find three positive constants T_0, δ_0, and ε_0 which guarantee the satisfaction of (4-3) for any unit vector w. Take $T_0 = \delta$ and

$$\varepsilon_0 = \frac{\alpha_1 \delta_0}{2\delta M_0}$$

where δ_0 is a positive constant to be determined, then

$$\| \int_{t_2}^{t_2+\delta_0} \mathrm{M}(t)\,\mathrm{w}\, dt \| \geq \| \int_{t_2}^{t_2+\delta_0} \mathrm{M}(t_2)\,\mathrm{w}\, dt \| - \| \int_{t_2}^{t_2+\delta_0} [\mathrm{M}(t) - \mathrm{M}(t_2)]\, dt \|$$

$$\geq \delta_0 \frac{\alpha_1}{\delta M_0} - \| \int_{t_2}^{t_2+\delta_0} [\mathrm{M}(t) - \mathrm{M}(t_2)]\, dt \|$$

Next, we choose δ_0 so that the second term in the last inequality be smaller than one half of the first term. This can be done by using the uniform continuity of the matrix M. Given the positive number

$$\varepsilon_0 = \frac{\alpha_1}{2\delta M_0}$$

the uniform continuity implies that we can find a positive number δ_0 , such that $\forall t \in [t_2, t_2+\delta_0]$,

$$\| \mathrm{M}(t) - \mathrm{M}(t_2) \| \leq \varepsilon_0$$

Use of this in the previous inequality leads to

$$\| \int_{t_2}^{t_2+\delta_0} M(t)\, w\, dt \| \geq \varepsilon_0$$

The definition of sufficient richness for M is thus satisfied.

□

Conversely, if a matrix M is bounded and sufficiently rich, it can be easily shown to be persistently exciting.

4.2 Conditions for Parameter Convergence

We now study the conditions for the convergence of the estimated parameters for the adaptive controller proposed in the last chapter. The sufficient richness condition will be shown to guarantee the parameter convergence and then this condition will be translated into the more convenient persistent excitation condition.

4.2.1 Parameter Convergence under SR condition

This condition of parameter convergence for the adaptive robot control system is similar to that for the adaptive linear-plant control systems in literature. Since our closed-loop dynamics of adaptive robot control system, due to the nonlinearity of the robot dynamics, is different from and more complicated than that of adaptive linear-plant control systems, the proof in the literature [Morgan and Narendra, 1977] cannot be used here. An independent and more complex proof has to be devised. Let us provide a detailed statement of the parameter convergence result first.

Theorem 4.1: *If the desired trajectories* q_d, $\dot{q}_d$, *and* $\ddot{q}_d$ *are bounded, and such that the matrix* $Y(q_d, \dot{q}_d, \dot{q}_d, \ddot{q}_d)$ *is sufficiently rich, the estimated parameters are*

guaranteed to converge asymptotically to the true parameters.

Proof: We shall use a method of contradiction to prove the above theorem, *i.e.*, if parameter error does not converge to zero, we can infer from the closed loop dynamics (3-15) that the tracking error s does not converge to zero. A complex sequence of reasoning is needed to accomplish this.

To start with, let us note that the assumed SR condition on $\mathbf{Y}_d$ implies the sufficient richness of the matrix $\mathbf{Y}$, according to lemma 4.1. Therefore, there exist positive numbers T_0, ε_0, and δ_0 such that $\forall t_1 \geq 0$, $\exists t_2 \in [t_1, t_1 + T_0]$ satisfying

$$\| \int_{t_2}^{t_2+\delta_0} \mathbf{Y}(t)\, dt \| \geq \varepsilon_0 \tag{4-9}$$

Now, let us consider the change of s over an interval $[t_2, t_2 + \delta_0]$ with δ_0 being the positive number associated with the sufficient richness of $\mathbf{Y}$.

From (3-15),

$$\frac{d}{dt}[\mathbf{Hs}] = \mathbf{Y}(t)\,\tilde{\mathbf{a}}(t) - (\mathbf{K}_D + \mathbf{C} + \dot{\mathbf{H}})\,\mathbf{s}$$

Integration of this equation is

$$\int_{t_2}^{t_2+\delta_0} \frac{d}{dt}[\mathbf{Hs}]\, dt = \int_{t_2}^{t_2+\delta_0} \mathbf{Y}(t)\,\tilde{\mathbf{a}}(t)\, dt - \int_{t_2}^{t_2+\delta_0} (\mathbf{K}_D + \mathbf{C} + \dot{\mathbf{H}})\,\mathbf{s}\, dt$$

Repeated use of the triangular inequalities leads to

$$\begin{aligned} &\|\mathbf{H}(t_2+\delta_0)\,\mathbf{s}(t_2+\delta_0)\| \\ &\geq \|\int_{t_2}^{t_2+\delta_0} \mathbf{Y}(t)\,\tilde{\mathbf{a}}(t_2)\, dt\| - \|\int_{t_2}^{t_2+\delta_0} \mathbf{Y}(t)\,[\tilde{\mathbf{a}}(t_2) - \tilde{\mathbf{a}}(t)]\, dt\| \\ &- \|\int_{t_2}^{t_2+\delta_0} (\mathbf{K}_D + \mathbf{C} + \dot{\mathbf{H}})\,\mathbf{s}\, dt\| - \|\mathbf{H}(t_2)\,\mathbf{s}(t_2)\| \end{aligned} \tag{4-10}$$

where $\mathbf{H}[q(t)]$ has been denoted by $\mathbf{H}(t)$ to stress its time-dependence.

Now let us examine the terms on the right-hand side of inequality (4-10). Due to the proven boundedness q, $\dot{q}$ and the assumed boundedness of the desired trajectories, the upper boundedness of the matrices $\mathbf{H}(q)$, $\dot{\mathbf{H}}(q)$, $\mathbf{C}(q,\dot{q})$ and $\mathbf{Y}$ can be shown, as explained in the last chapter. Since s has been shown to converge to zero, the third (integration on a fixed interval) and fourth terms will converge to zero as t_2 tends to infinity. Since the adaptation law (3-12) and the convergence of s imply that $\dot{\hat{a}}$, the rate of change of the estimated parameters, asymptotically converges to zero, the difference

$$\tilde{a}(t_2) - \tilde{a}(t) = \dot{\tilde{a}}(\xi)(t_2 - t) \qquad t_2 \le \xi \le t \le t_2 + \delta_0$$

converges to zero as $t_2 \to \infty$. Therefore, the second term on the right-hand side of (4-10) also converges to zero. If we use $L3(t_2)$ to denote the sum of the last three terms on the left hand side of (4-10), *i.e.*,

$$L3(t_2) = \left\| \int_{t_2}^{t_2+\delta_0} \mathbf{Y}(t)[\tilde{a}(t_2) - \tilde{a}(t)]\, dt \right\| - \left\| \int_{t_2}^{t_2+\delta_0} (\mathbf{K}_D + \mathbf{C} + \dot{\mathbf{H}})\, s\, dt \right\| - \|\mathbf{H}(t_2)\, s(t_2)\|$$

then $L3(t_2) \to 0$ as $t_2 \to \infty$.

At this point, we are ready to make an intuitive sketch of the proof itself. Let the time t_2 be extremely large, so that the last three terms on the right-hand side of (4-10) be essentially zero. Then, (4-10) is essentially,

$$\|\mathbf{H}(t_2 + \delta_0)\, s(t_2 + \delta_0)\| \ge \left\| \int_{t_2}^{t_2+\delta_0} \mathbf{Y}(t)\tilde{a}(t_2)\, dt \right\|$$

If $\tilde{a}(t_2)$, does not converge to zero, then the above relation together with the sufficient richness of $\mathbf{Y}$ and the lower boundedness of $\mathbf{H}$ will be shown to indicate that s does not converge to zero either. This would constitute a contradiction to

the shown convergence of s.

Let us now formalize the above intuitive arguments. Assume that $\tilde{a}(t)$ does *not* converge to zero as $t \to \infty$. This implies that there exists at least one positive number ε_1, such that $\forall T_1 > 0$, >, we can find $t > T_1$ which satisfies

$$\|\tilde{a}(t)\| \geq \varepsilon_1 \tag{4-11}$$

On the other hand, because of the asymptotic convergence of $L3(t_2)$ as $t_2 \to \infty$, for the positive number

$$\varepsilon = \frac{\varepsilon_1 \varepsilon_0}{2} > 0$$

with ε_1 given by (4-9), $\exists T_2 > 0$, $\forall t > T_2$

$$\|L3(t)\| < \varepsilon \tag{4-12}$$

Similarly, due to the asymptotic convergence of Hs, for the above ε, $\exists T_3 > 0$ such that for any $t > T_3$

$$\|H(t)s(t)\| < \varepsilon \tag{4-13}$$

i.e., the magnitude of $H(t)s(t)$ is *smaller than* ε if t is sufficiently large.

Now take the T to be the largest among T_1, T_2 and T_3, *i.e.*, T = max(T_1, T_2, T_3), then $\exists t > T$ such that (4-11), (4-12) and (4-13) are all satisfied.

Now let t be chosen this way and t_2 chosen according to the sufficient richness definition, *i.e.*, $t_2 \in [t, t + T_0]$ and (4-9) is satisfied. Then since $t_2 + \delta_0 > t > T_3$, we have

$$\| H(t_2 + \delta_0)s(t_2 + \delta_0) \| \leq \varepsilon \tag{4-14}$$

similarly to (4-13).

However, with the above inequalities satisfied, inequality (4-13) and (4-11) lead to

$$\| H(t_2 + \delta_0)s(t_2 + \delta_0) \| \geq \| \tilde{a}(t_2) \| \| \int_{t_2}^{t_2+\delta_0} Y(t) \frac{\tilde{a}(t_2)}{\| \tilde{a}(t_2 \|)} dt \| \; - \; \varepsilon$$

$$\geq \varepsilon_1 \, \varepsilon_0 - \varepsilon = \varepsilon$$

where the definition of sufficient richness has been used with the unit vector being

$$w = \frac{\tilde{a}(t_2)}{\| \tilde{a}(t_2 \|)}$$

This contradicts the condition (4-14), implying that tracking error s will not converge to zero if parameter error does not converge to zero. Therefore $\tilde{a}(t) \rightarrow 0 \quad$ as $t \rightarrow \infty$. □

Note that the above theorem also explains why condition (4-8) is called "sufficient richness" condition: it is sufficient to guarantee the convergence of the estimated parameters.

4.2.2 Parameter Convergence Under PE Condition

In view of the comparative convenience (no need of dealing with arbitrary vector w) of checking persistent excitation, it is of interest to restate the above theorem in terms of the persistent excitation condition:

Theorem 4.2: *If the desired trajectories are such that the matrix* $Y(q_d, \dot{q}_d, \dot{q}_d, \ddot{q}_d)$ *is persistently exciting and uniformly continuous, then the estimated parameters asymptotically converge to the true parameters.*

This theorem results straightforwardly from theorem 4.1 and lemma 4.1. For

convenience, when Y_d is persistently exciting, we shall also say that the desired trajectories are persistently exciting. It is also interesting that the condition of uniform continuity appears in the above theorem. The necessity of this condition can be explained in view of the integrator-type adaptation law: the excitation contents of the signals will be filtered out by the integrator unless the signals are smooth. The necessity can also be seen in the discussion in subsection 4.1.1. It is useful to remark that, from examining composition of the matrix $Y(q, \dot{q}, \dot{q}_r, \ddot{q}_r)$ as given by (3-4), one easily conclude the uniform continuity of Y_d is guaranteed by the uniform continuity of $\ddot{q}$.

4.3 Practical Implications of Persistent Excitation and Parameter Convergence

The concept of persistent excitation has useful and interesting interpretations. The simplest interpretation of persistent excitation is that the trajectory is so complex (or rich in excitation) that only the true parameters can guarantee exact tracking. Alternatively, the persistent excitation of a trajectory means that, during the operation along this trajectory, the parameter error in any direction of parameter space, *i.e.*, any combination of the errors in different parameters, will always cause error in tracking motion. This is seen in the closed-loop dynamics (3-15), where the parameter error, multiplied by the matrix Y, is seen as the cause for the tracking error s. Of course, from an information point of view, we can also say that persistent excitation implies that the tracking error contains adequate information about parameter errors.

Although the convergence of estimated parameters appears to be a purely theoretical issue, in view of the fact that the objective of motion tracking is to make the tracking errors to be small, it is indirectly and but closely related to the

practical performance of the system. First, if the desired trajectories are persistently exciting, parameters will be close to the true parameters after some time. Thus the controller will be using basically correct parameters and the tracking errors will be small thereafter regardless of the feature of the later portion of the desired trajectories. The relationship of parameter convergence and tracking convergence is clearly seen in Figure 3-6(a) and Figure 3-13(a). In 3-6(a), the tracking error remains small after the estimated parameters get close to the true parameters, while, in 3-13(a), the tracking errors become large each time the desired trajectories encounter some excitation. simply speaking, parameter convergence implies better tracking error convergence.

Secondly, parameter convergence is particularly useful for operations on more than one set of desired trajectories. Consider a robot carrying the an unknown load (*e.g.*, a tool) which is supposed to move along a straight line for a while and then required to draw a circle. If the motion along the straight line is persistently exciting, the estimated parameters will soon be close to the true parameters. This implies that the estimated parameters at the end of the straight line motion can be used for the non-adaptive or adaptive control of the following circular motion, leading to accurate tracking even in the beginning of the circular motion. If the first trajectory does not have persistent excitation, the estimated parameters at the end of the first path operation will be meaningless, and use of them as initial parameters in the control on the new trajectory is likely to cause large initial tracking errors or even excitation of unmodeled dynamics. This discussion also suggests the usefulness of monitoring the persistent excitation of a trajectory which indirectly gives the indication of parameter convergence (note that one does not know the true parameters and therefore the closeness of the estimated parameters to the true ones). It is useful to point out that the parameter

convergence condition in theorem 4.1 is expressed in terms of the desired trajectories. Therefore, one can check off-line whether a trajectory is persistently exciting and parameters will be converging.

Thirdly, on persistently exciting trajectories which lead to convergent parameters, the adaptive control system is more robust to non-parametric uncertainties. On trajectories without persistent excitation, the tracking errors do not contain adequate information on parameter errors, and the adaptation mechanism may interpret the noise and disturbance components as parameter information, leading to the so called "parameter drift". Note that the level of robustness of the adaptive control system is dependent on the level of persistent excitation in the desired trajectories, implying that the adaptive controller can withstand larger disturbances and have smaller steady-state errors on trajectories with strong persistent excitation. There is currently no standard measure of the persistent excitation level. One meaningful measure is the condition number of the persistent excitation matrix

$$\mathbf{P} = \int_0^T \mathbf{Y}_d^T \mathbf{Y}_d \, dr$$

where T is the total time for executing the desired motion (the conditioning number is the ratio of the largest eigenvalue over the smallest eigenvalue). Condition number close to 1 implies good persistent excitation and large conditioning number reflects poor persistent excitation.

4.4 Generation of Persistently Exciting Trajectories

After satisfying other requirements such as torque limits and path constraints, the desired trajectories should be planned to be as persistently exciting as possible. The generation of persistently exciting desired trajectories is an

interesting research topic because of its relevance to system performance, particularly the robustness of the adaptive system to noise, disturbances and unmodeled dynamics. If the desired trajectories are persistently exciting, we can expect the resulting adaptive motion to be accurate, the estimated parameters convergent and the adaptive system robust to non-parametric uncertainties. Note that, in saying so, we do not imply that the adaptive controller will not perform well on non-persistently exciting trajectories (the tracking errors are convergent and the system can be robustified by simple techniques such as dead-zones).

However, the issue of generating persistently exciting trajectories is a difficult one because of the following reasons.

- there are no simple rules on what specific forms of desired trajectories can make the matrix Y_d persistently exciting;

- there is no efficient numerical method of planning persistently exciting trajectories;

- there are many other constraints which the desired trajectories have to satisfy besides persistent excitation;

Now let us discuss each of these aspects of the trajectory generation problem.

A. Persistent Excitation and Sinusoids

In adaptive linear-plant control, the condition of parameter convergence is also a persistent excitation condition like (4-7). There is a simple result on satisfaction of persistent excitation condition for that case: persistent excitation is guaranteed for an adaptive controller with m unknown parameters if the desired trajectories

have (m/2) or more sinusoid signals. However, this simple result *cannot* be shown true for our adaptive robot control system because the technical difficulty caused by the fact that the matrix Y contains a lot of nonlinear functions (which are mostly quadratic functions of joint velocities due to the Coriolis and centripetal forces, trigonometric functions of joint positions, and their products).

It is possible that a weaker condition is possible for our persistent excitation condition due to the presence of the nonlinearity in Y_d, since passing a sinusoidal function through a nonlinearity may generally generates higher harmonics and subharmonics and thus more excitation. Consider a nonlinear function f(x) with x being a sinusoid function of frequency ω, *i.e.*, $x(t) = A\sin\omega t$. Then, $f(\sin\omega t)$ is also a periodic function. Using Fourier series, one can expand this nonlinear function as the sum of many sinusoids. Specifically, it is interesting to note that quadratic functions have the property of altering the sinusoid frequency instead of generating a lot of other harmonies, as seen in the relation $\sin^2(t) = (1 - \cos 2t)/2$. This implies that if the desired trajectory contain q_d contains one sinusoid frequency ω, the matrix Y (which includes quadratic functions of velocities due to Coriolis and centripetal forces in robot dynamics) will contain both frequency ω and frequency 2ω.

In the Y_d matrix, the terms corresponding to the Coriolis and centripetal forces are nearly quadratic. For example, for the 2-DOF robot in Figure 3.1, the component $Y_d(1, 3)$ is

$$Y_d(1, 3) = c_{d21}\ddot{q}_{d2} - s_{d21}\dot{q}_{d2}$$

where

$$c_{d21} = \cos(q_{d2} - q_{d1}) \quad s_{d21} = \sin(q_{d2} - q_{d1})$$

Assume that the desired position is

$$\begin{bmatrix} q_{d1} \\ q_{d2} \end{bmatrix} = \begin{bmatrix} 2\sin t \\ \sin t \end{bmatrix}$$

then

$$Y_d(1,3) = -\cos(\sin t)\sin t + \sin(\sin t)\cos^2 t$$
$$= -\cos(\sin t)\sin t + \frac{\sin(\sin t)}{2}(1 - \cos 2t)$$

That is to say, that desired positions with only one sinusoidal component leads to Y_d matrix with two sinusoidal components. If we use intuition from adaptive linear plant control, this suggests that m sinusoidal signals might allow us to estimate *more than* $2m$ parameters.

Even though the nonlinearity in the Y matrix may allow more excitation to be produced, there are no clear rules on the number of sinusoids and the number of estimatable parameters.

B. Computational Methods

In practice, polynomial trajectories are often selected because of their simplicity. However, polynomial trajectories have poor persistent excitation because they are too smooth. This indicates that there is a need for algorithms which can find trajectories with good persistent excitation. Currently, there are few algorithms which can achieve this purpose. In [Armstrong, 1988], a optimization method was proposed to choose persistently exciting trajectories. But this method was mainly developed for parameter estimation tasks using least-square-type estimators. Furthermore, the computational burden of this trajectory generation method seems to be excessive for adaptive control problems where the

trajectory planning should be finished very quickly.

C. Constraints in generating persistently exciting trajectories

In trying to plan persistently exciting desired trajectories for the adaptive robot control system, we must keep in mind that many other constraints should be accounted for. In tasks where the path has been given, one only has freedom in choosing the motion variation pattern along the path. In tasks where the path is not given, one has to make sure that the planned trajectories do not encounter obstacles in the robot workspace. There are other factors such as torque limits, acceleration jerks and computation requirement. Torque limits should be observed because actuation saturation should be avoided. Acceleration jerks should be reduced as much as possible to avoid the excitation of unmodeled robot dynamics. Computational burden should be considered because of productivity requirement.

In terms of trajectory excitation, it is useful to make some remarks. On one hand, it is clear that the level of persistent excitation bears certain relation to the complexity of desired trajectories. Therefore, one has to make the desired trajectories reasonably complicated in order to have certain level of persistent excitation and, accordingly, adaptive control robustness. On the other hand, one should not choose too complex trajectories. The reason is that too complex trajectories tends to excite unmodeled dynamics and thus pose the danger of control instability. Complex trajectories also tend to be hard to follow, leading to large initial tracking errors in adaptive control.

4.5 Summary

The estimated parameters in the adaptive controller are shown to be convergent to the true parameters if the desired motion trajectories are sufficiently rich or persistently exciting. Intuitively, persist excitation means that the desired trajectories are so complex only the true parameters can result in tracking error convergence. Though it is desirable for the desired trajectories to be persistently exciting, generation of such trajectories is difficult. This is partly due to the complexity of the nonlinear matrix Y_d and partly to the multiple constraints on the desired trajectories. In practice, desired trajectories should be generated such that the various constraints on the trajectories are satisfied *and* there is as high a level of persistent excitation as possible.

Chapter 5
Experimental Implementation

In chapters 3 and 4, the analytical and simulation studies have provided considerable insights into the behavior of proposed adaptive joint-space controller. Note that the studies are made based on the idealized model of the robot dynamics. However, in practice, robot control systems contain all sorts of non-parametric uncertainties, such as joint and link flexibility, motor dynamics, measurement noise, computational delay. Therefore, it is natural for one to wonder how the proposed adaptive controller will perform in real-world implementations: Will the adaptive control system still be stable? Will the motion tracking be accurate? From a practical point of view, these are indeed the most important questions about the adaptive controller.

This chapter addresses these questions by showing the experimental results of the adaptive controller implementation on a two-DOF manipulator at the Whitaker College of MIT. In the experiments, the adaptive controller is used to control the

manipulator which moves with high speeds and has high uncertainty in inertial parameters. The consistent results of the adaptive controller under various loading conditions demonstrate the effectiveness of the adaptive controller in practical applications. To provide a reference point for comparing the performance of the adaptive controllers, the results of the PD and computed-torque controllers are also presented.

This chapter is arranged as follows. Section 5.1 presents the experimental setup. Section 5.2 describes the adaptive control system for the robot, Section 5.3 provides the experimental results of the adaptive controller and the PD controller, highlighting the effects of adaptation. Section 5.4 shows the comparison of adaptive and computed-torque controllers, given the same amount of initial knowledge about the link/load parameters. Section 5.5 shows the performance of adaptive, PD and computed-torque controllers when the robot carries a large unknown load. Section 5.6 provides a sketchy analysis of the factors neglected in the basic stability and convergence analysis.

5.1 Experimental Equipment

The arm used for the experiments is a 2-dof manipulator developed at the Whitaker College of Health Sciences at MIT. Its physical structure is shown in Figure 5.1. The arm was originally designed to be used as an experimental apparatus for investigating human arm movements [Faye, 1986]. Because it is capable of high speed movement and there is significant nonlinearity and coupling effects among the two degrees of freedom, it is well suited for the study of the performance of the adaptive robot controller.

The arm involves a somewhat peculiar mechanism, more clearly seen in

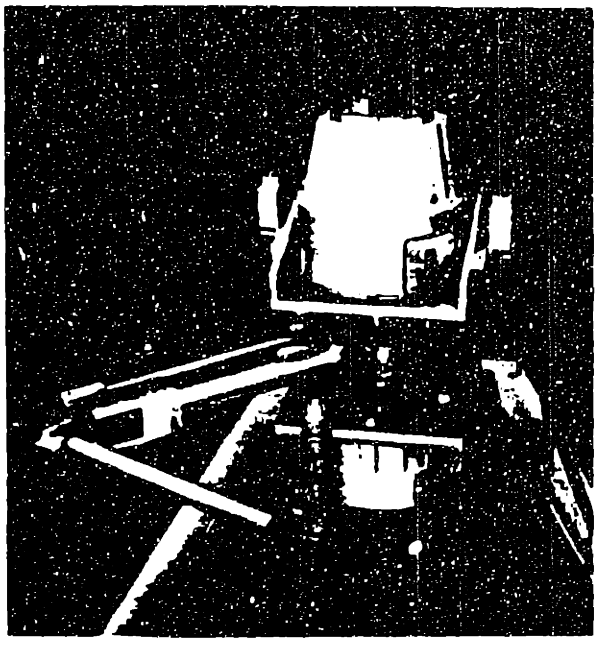


Figure 5-1: the Whitaker College robot

Figure 3.1. The robot is called a semi-direct-drive robot because the second link is indirectly driven by a motor located at the base through a four-bar mechanism. This unique design represents a compromise between the conventional geared-driven (or belt-driven) robots and the genuine direct-drive robots, avoiding the low stiffness and low accuracy problems caused by the gears and the necessity of installing heavy motors at the joints.

The robot control system hardware consists of a two-link arm, two DC servo motors with amplifiers, two optical encoders, two tachometers and a microcomputer PDP 11/73. The composition of this system is symbolically shown in Figure 5.2.

The arm lies in the horizontal plane, and therefore the effects of gravity are

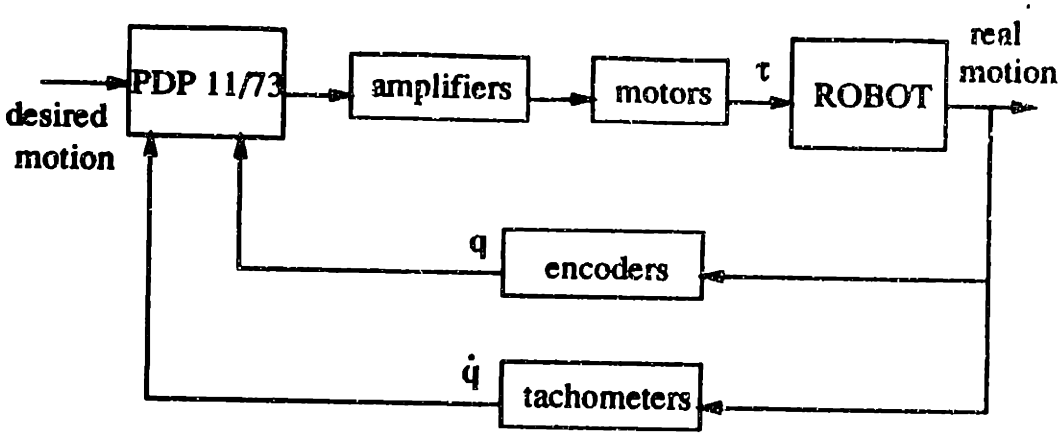


Figure 5-2: the robot control system

absent (originally constructed this way for reasons unrelated to adaptive control implementation. In fact, the presence of gravity would further demonstrate the advantages of adaptive parameter estimation). The two links are made of aluminum, with lengths of 0.37 m and 0.34 m, and masses of 0.9 kg and 0.6 kg, respectively. Although it uses a four-bar linkage mechanism, the arm is not dynamically mass-balanced, and therefore presents full coupling effects. The motors are driven by PMI SSA 40-10-20 pulse-width-modulated switching servo amplifiers. The inductance of the motors is low enough so that the amplifiers be considered current sources to the motors. The two JR16M4CH motors, mounted on a rigid supporting frame, are rather large and heavy (16 kgs each), but this does not represent a practical problem since they are both located at the manipulator base and do not move with the arm. The four-bar linkage mechanism is used to transmit the torque from the upper motor to the outer link. The motion of the relative angle between the inner and outer links can range from 39° to 139°. This range is made possible by using an offset in the elbow of the manipulator. Due to the fluctuation and nonlinearity limitations of the amplifier capacity, the maximum torque which each motor can generate is restricted to be 9 N.m. In the controller

implementation, the motors together with the amplifiers are regarded as having a constant of 1.12 N.m/volt as their transfer function.

The joint positions are measured by incremental optical encoders attached to the output shaft of each torque motor, with a resolution of 12 bits/180°, *i.e.*, 0.045°. The joint velocities are directly measured by tachometers. The tachometers, built in the motor housings, were originally designed for high speed operations of motor shafts, and therefore have low output voltages for the comparatively low speed rotations of the direct-drive arm shafts. Rather than modify the tachometers themselves, their output signals are amplified. Since tachometer signals tend to be noisy, this amplification makes the resulting signals even more noisy. Another problem with the tachometers is their sensitivity to the vibrations of the supporting frame of the arm, which exhibits natural structural modes at about 400 Hz. Analysis in [Faye, 1986] suggests the use of a lowpass filter consisting of a cascade of four passive first-order filters, each with a nominal cutoff frequency of 50 Hz. While this filter is very effective in eliminating the noises in the amplified tachometer signals, it leads to significant phase lag at low frequencies (about 5 degrees at 1 Hz). This phase lag is believed to be one major source of residual tracking error for the adaptive control implementations.

Position and velocity measurements, after proper A/D conversions, are sent to the PDP 11/73 for torque computation. The control programs in the PDP computer are written in the C language. In order to reduce the computational delay, the computer programs are written to be as efficient as possible. However, since the PDP 11/73 computer is quite old and slow, we only managed to obtain a sampling frequency of 200 Hz. Though sampling frequency is relatively low, it still allows the benefits of adaptive control to be clearly observed in the following experiments.

5.2 Overview of the Experiments

Before presenting the results in the specific experiments, we discuss some general issues.

A. Adaptive Controller Design

The robot in Figure 5.1 is schematically represented by the diagram in Figure 3.1. The adaptive control design in the experiments is based on the dynamic model in (3-19).

The adaptive controller is almost the same as the adaptive controller in section 3.3. The difference is that the gain matrices $\mathbf{K_D}$, $\mathbf{\Lambda}$, $\mathbf{P}$ are now chosen to diagonal matrices,

$$\mathbf{K}_D = diag(k_{d1}, k_{d2}) \tag{5-1a}$$

$$\mathbf{\Lambda} = diag(\lambda_1, \lambda_2) \tag{5-1b}$$

$$\mathbf{\Gamma} = diag(\gamma_1, \gamma_2, \gamma_3, \gamma_4) \tag{5-1c}$$

instead of choosing them as the products of positive constants and unit matrices. This is done to allow more flexibility and , therefore, better control performance. However, the matrices are not chosen to be full matrices to avoid the complexity associated with too many parameters.

As a result of the gain choices in (5-1), the control law in the adaptive controller has the form

$$\tau_1 = Y_{11} a_1 + Y_{13} a_3 + Y_{14} a_4 - k_{d1} s_1 \tag{5-2a}$$

$$\tau_2 = Y_{22} a_2 + Y_{23} a_3 + Y_{24} a_4 - k_{d2} s_2 \tag{5-2b}$$

where quantities Y_{11}, Y_{13}, Y_{14}, Y_{22}, Y_{23},Y_{24}, s_1, s_2 are all defined in section 3.3.

The adaptation law can be explicitly written as

$$\dot{\hat{a}}_1 = -p_1 Y_{11} s_1$$

$$\dot{\hat{a}}_2 = -p_2 Y_{22} s_2$$

$$\dot{\hat{a}}_3 = -p_3 (Y_{13} s_1 + Y_{23} s_2)$$

$$\dot{\hat{a}}_4 = -p (Y_{14} s_1 + Y_{24} s_2)$$

The initial values of the estimated parameters will be chosen in the specific experiments. The sampling frequencies of all the controllers, including the PD controller and the PD part of the adaptive controller, are kept at 200 hz.

B. Fairness of comparison

It is important to note that the performance of a controller depends not only on the form of the controller, but also on the design parameters of the controller. This implies that, in comparing different controllers, it is necessary to make sure that the best achievable performances of the controllers are compared. Otherwise, the comparison is not fair.

For our adaptive controller, the design parameters are contained in the gain matrices Γ, K_D and Λ. As indicated in section 3.3, larger values of these parameters lead to higher accuracy of the adaptive controller. But these parameters cannot be increased to arbitrarily large values because of the possibility of exciting unmodeled dynamics. There is a tradeoff of tracking accuracy and control stability involved. In the experiments, the design parameters are chosen by trial and errors. Specifically, the design parameters are initially chosen to be relatively small

values, and then gradually increased until some shakiness is detected in the system signals. The values obtained by discounting the maximum values by 20% are regarded to be the optimal parameters and the corresponding performance is regarded as the best performance of the adaptive controller. The design parameters of the PD and computed controllers are chosen similarly.

C. The Desired Trajectories

In the experiments, the desired trajectories lasted for one second, such that the robot moved from point A (Figure 3.8) to point B in the first half second and then stays at the end point for the next half second. The positions of the points A and B are defined by the joint angles following equation (3-24). The desired position trajectories are plotted in Figure 5.3. The desired positions in the first half second are constructed by interpolation with a fifth-order polynomial for each joint. The velocities and accelerations at the beginning and the end of the half second are all specified as zero. These trajectories are essentially the same as the first two seconds of the trajectories in Figure 3.7 except that the time is reduced to one second now.

These trajectories are very fast. The maximum value of the desired velocities occur at t=0.5 sec and have the values

$$\dot{\mathbf{q}}_{max} = [300 \;\; 262.5]^T \;\; (deg./sec.) \tag{5-3}$$

In fact, these are essentially the fastest polynomial trajectories the robot can follow without running into the possibility of motor saturation. Slower polynomial trajectories were also considered in the experiments, but the results are not presented in this chapter because they are similar to ones here.

These polynomial trajectories are interesting because they resemble the smooth

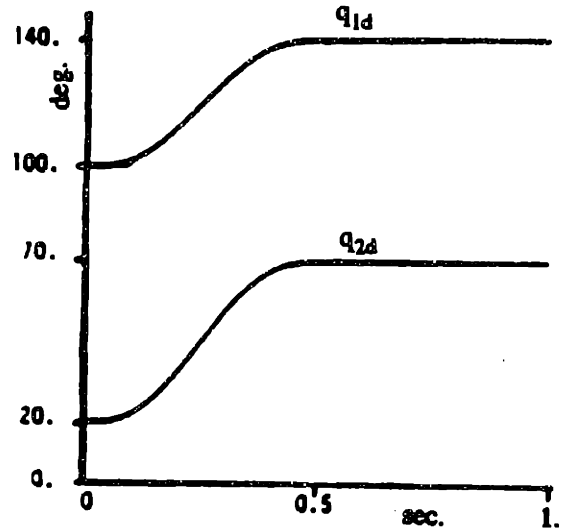


Figure 5-3: the desired position trajectories

trajectories frequently chosen in practical tracking tasks. They are also interesting from a control point of view because their lack of persistent excitation represents a challenge to the adaptive controller. We remark that more complex trajectories such as those obtained by superimposing sinusoids would have been useful in demonstrating parameter convergence. But at the time (late 1986) the experiments were carried out, the issue of parameter convergence was not understood and it was not known what kind of trajectories could lead to convergent parameters. By the time the parameter convergence conditions were found and understood, the experimental equipment was no longer available.

5.3 Comparison of Adaptive and PD Controllers

The first set of experiments intends to illustrate the effects of adaptation on the tracking errors. Therefore, the adaptive controller is compared with a PD controller. In these experiments, no load was attached to the second link of the manipulator and the parameter uncertainty for the adaptive controller came from the two links of the manipulator. For the adaptive controller, the initial values of

the parameter estimation were taken to be zero, that is, the parameters of the robot links were assumed to be totally unknown at the beginning. Thus, similarly to the simulations before, the adaptive controller started as a PD controller and the feedforward part became more active as the parameter adaptation was driven by the tracking errors. Good values of the adaptation gains γ_i were found to be all equal to 0.02, *i.e.*,

$$\gamma_1 = \gamma_2 = \gamma_3 = \gamma_4 = 0.02$$

As seen in the experiments, smaller values tended to lead to larger tracking errors while further increase of the gains tended to lead to oscillations in the torque and motion signals, indicating the tendency of exciting unmodeled dynamics.

The PD controller has the following form

$$\tau_1 = -k_{d1}\dot{\tilde{q}}_1 - k_{p1}\tilde{q}_2 \tag{5-4a}$$

$$\tau_2 = -k_{d2}\dot{\tilde{q}}_2 - k_{p2}\tilde{q}_2 \tag{5-4b}$$

For both controllers, proper choices of the feedback gains $\mathbf{K_D}$ and $\mathbf{K_P}$ (for the adaptive controller, $\mathbf{K}_P = \mathbf{K}_D\Lambda$) were determined by gradually increasing their values and monitoring the control operations. The two controllers were found to become unstable at essentially the same values of these matrices. This suggested that the adaptive controller had basically the same level of robustness to noises and high-frequency unmodeled dynamics as the PD controller. The following feedback gains

$$\mathbf{K}_d = \begin{bmatrix} 2.0 & 0 \\ 0 & 1.5 \end{bmatrix}$$

$$K_p = \begin{bmatrix} 40. & 0 \\ 0 & 30. \end{bmatrix}$$

or, equivalently for the adaptive controller, $\lambda_1 = 20$ and $\lambda_2 = 15$, were found to yield the best accuracy while avoiding noticeable excitation of the vibrational modes of the links.

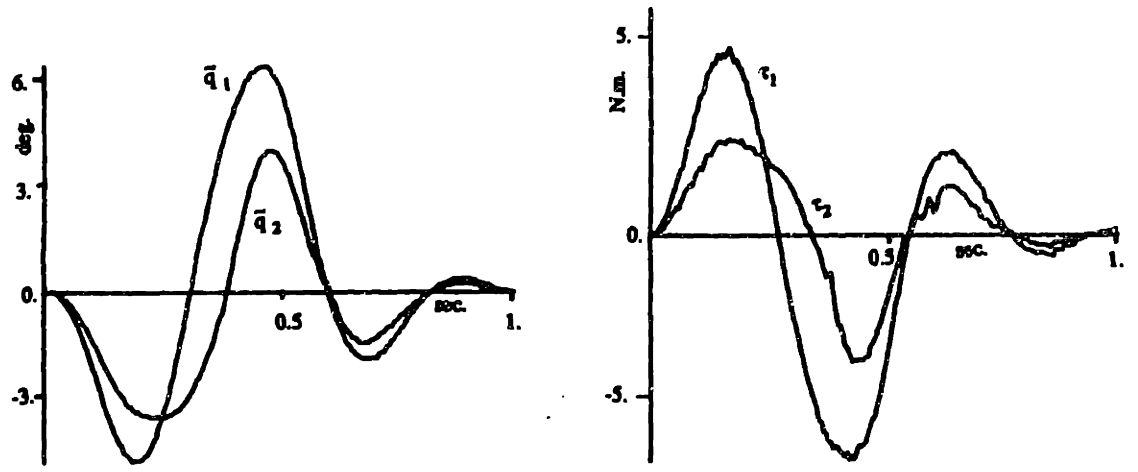


Figure 5.4 : PD Control : (a) position errors $\tilde{q}$; (b) joint torques τ

The results of the PD controller are plotted in Figure 5.4, while those of the adaptive controller are given in Figure 6. The maximum joint errors for the PD controller are 6.57° and 4.°, but those for the adaptive controller are only -2.12^o and -2^o. As expected, the errors and control torques of the two controllers are very close in the initial period, but at roughly $t = 0.15$ second, the tracking errors in the adaptive controllers stop growing due to the effects of the adaptive feedforward action (which would otherwise reach $-5.^o$ and -3.7^o, respectively). It is seen that parameter estimates of the adaptive controller are quickly driven by the tracking

errors. The parameter estimates in Figure 5(c) are seen to be very smooth though the measurement signals contain noises. This smoothness is due to the integrator structure of the adaptation law. It is desirable because the smoothness of the estimated parameters avoids the excitation of the vibrational modes of the links. At the end of the tracking operation in the first-half-second, usually an important instant in applications like pick-and-place tasks, the joint errors of the adaptive controller are only -0.4^o in both angles, while the PD controller is suffering almost maximum joint errors. In the last-half-second, the joint errors are regulated by the controllers and the arm settles down to the end position $\mathbf{q}_e$.

Since these polynomial trajectories do not have good persistent excitation, as seen in the simulation results of section 3.4, the estimated parameters can not be expected to be convergent. In the presence of non-parametric uncertainties, it is possible for the parameters to drift if long-time operation or repetitive motion is made. To avoid such parameter drift effect, small dead zones can used in the computation of s (*i.e.*, replacing s_1 by zero in the adaptation law if $|s_1| < \delta_1$, and similarly for the second joint). The sizes of the dead zone are determined based on the resolutions of the encoders and the A/D converters for the tachometers. With small dead zones used, experimental results indicated relatively little drift of estimated parameters in a 20-times repetition of the same motion.

It is useful to make a remark about reducing the steady-state errors due to stiction. Small steady state joint errors of -0.09^o (2 encoder counts) and -0.135^o (3 encoder counts) are observed in Figure 5.5 for the adaptive controller. These errors arise from the stiction effects of the static frictions at the motor shafts of this manipulator. The magnitudes of stiction were determined to be roughly 0.17 N.m at each joint. Small stiction-compensation terms $[-0.15\,sgn(s_1), -0.10\,sgn(s_2)]^T$ were actually added in the control laws for Figure 5.5 and 5.6, otherwise the

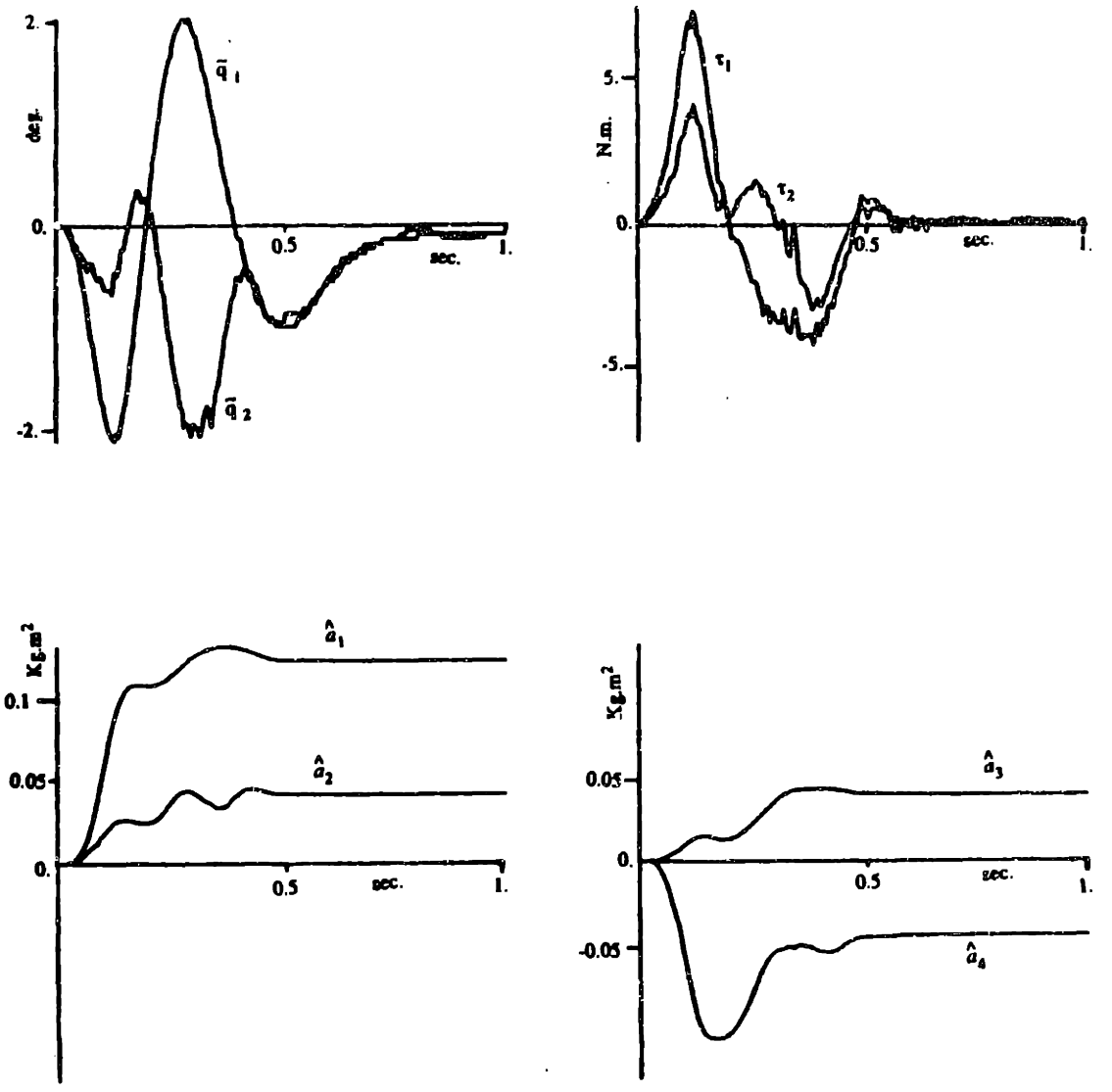


Figure 5.5 : Experimental Results of adaptive control:

(a) position errors $\tilde{q}$; (b) joint torque τ;

(c) estimates $\hat{a}_1, \hat{a}_2$; (d) estimates $\hat{a}_3, \hat{a}_4$

transient errors would be basically the same but steady-state errors would have been about -0.3^o. The magnitudes of these compensation terms are limited to small values by the condition that the system should not cross the dead zones in s

in less than two sampling periods, thus avoiding small-magnitude chattering. The PD controller has the same level of steady state errors, although they can not be clearly seen in Figure 5.4 due to the larger scale used.

5.4 Comparison of Computed Torque and Adaptive Controllers

The purpose of the set of experiments in this section is to show how to use *a priori* parameter information in adaptive robot control and, to show the advantage of adaptive control over computed-torque control when given the same amount of initial parameter knowledge.

In practice, one always has some a priori knowledge of the physical parameters of the robot and load, possibly by computation based on design data or computer vision information. Based on this knowledge, one may obtain a set of rough (not very accurate) estimates of the parameter values. These "rough estimates" can be used to initialize the estimated parameters in the adaptive scheme and to temporarily stop adaptation on a parameter if the bound of an equivalent parameter is known and the estimated parameter reaches that bound.

With initial information available on the parameters, one may alternatively use the computed torque method to control the robot with the "rough estimates" as parameter values in the inverse dynamics computation. The computed torque method is a fairly standard approach, whose formulation can be found in a number of papers [*e.g.*, Luh, *et al.*, 1980; Khosla and Kanade, 1986; An *et al.*, 1987]. For the 2-DOF robot at hand (which is not subject to the gravity), its input torque can be written as

$$\tau = \hat{H} (\ddot{q}_d - k_1 \dot{\tilde{q}} - k_2 \tilde{q}) + \hat{C} \dot{q}$$

We now use a set of experiments to compare the adaptive controller and the computed-torque controller. Let us take the following choice for the design parameter matrices in the computed-torque controller

$$k_1 = \begin{bmatrix} 2\omega_1 & 0 \\ 0 & 2\omega_2 \end{bmatrix}$$

$$k_2 = \begin{bmatrix} \omega_1^2 & 0 \\ 0 & \omega_2^2 \end{bmatrix}$$

as is commonly done in robot control, where ω_1 and ω_2 are two positive constants. With this choice of k_1 and k_2, a critically damped error dynamics would be obtained if the exact parameters were used. Selecting k_1 and k_2 experimentally as before, the best values of ω_1 and ω_2 are determined to be $\omega_1 = 20$ and $\omega_2 = 30$.

For the adaptive controller, the design parameters are the same as before except that Γ is doubled. This doubling is made because reasonable initial parameters are already available and less oscillation will be encountered in estimated parameters (magnitude of the adaptation gain is limited by the excitation of the modeled dynamics through the oscilation of the estimated parameters). In the experiments, the parameter values used for the computed torque method and as initial values of the adaptive control are

$$a_1 = 0.11\ Kg.m^2$$

$$a_2 = 0.0285\ Kg.m^2$$

$$a_3 = 0.033\ Kg.m^2$$

$$a_4 = 0.$$

These values of the parameters were determined by Faye in his unrelated work on the impedance control of the manipulator [Faye, 1986]. They were computed from the engineering drawings of the arm links. In addition to the discrepancies between the real quantities and those on the drawings, the mass of the force sensor attached to the endpoint were not accounted for in this computation and caused some inaccuracy in the above values. A rough estimate indicates that the above values of the initial parameters had an error of 20% or so.

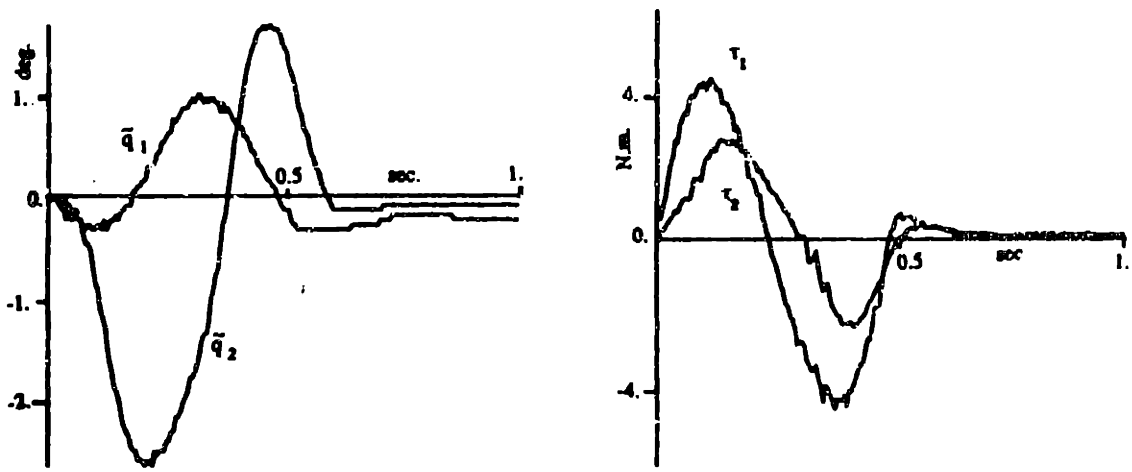


Figure 5.6 : Computed-Torque control: (a) position errors $\tilde{q}$; (b) joint torque τ

The results are shown in Figure 5.6 for the computed torque method and Figure 5.7 for the adaptive controller. The maximum joint tracking errors for the computed torque are $1.^o$ and -2.5^o , respectively, while those for the adaptive controller are 0.95^o and -0.96^o. The tracking error of the first joint is smaller

because the parameter uncertainty is larger in the four-bar mechanism associated with the second link.

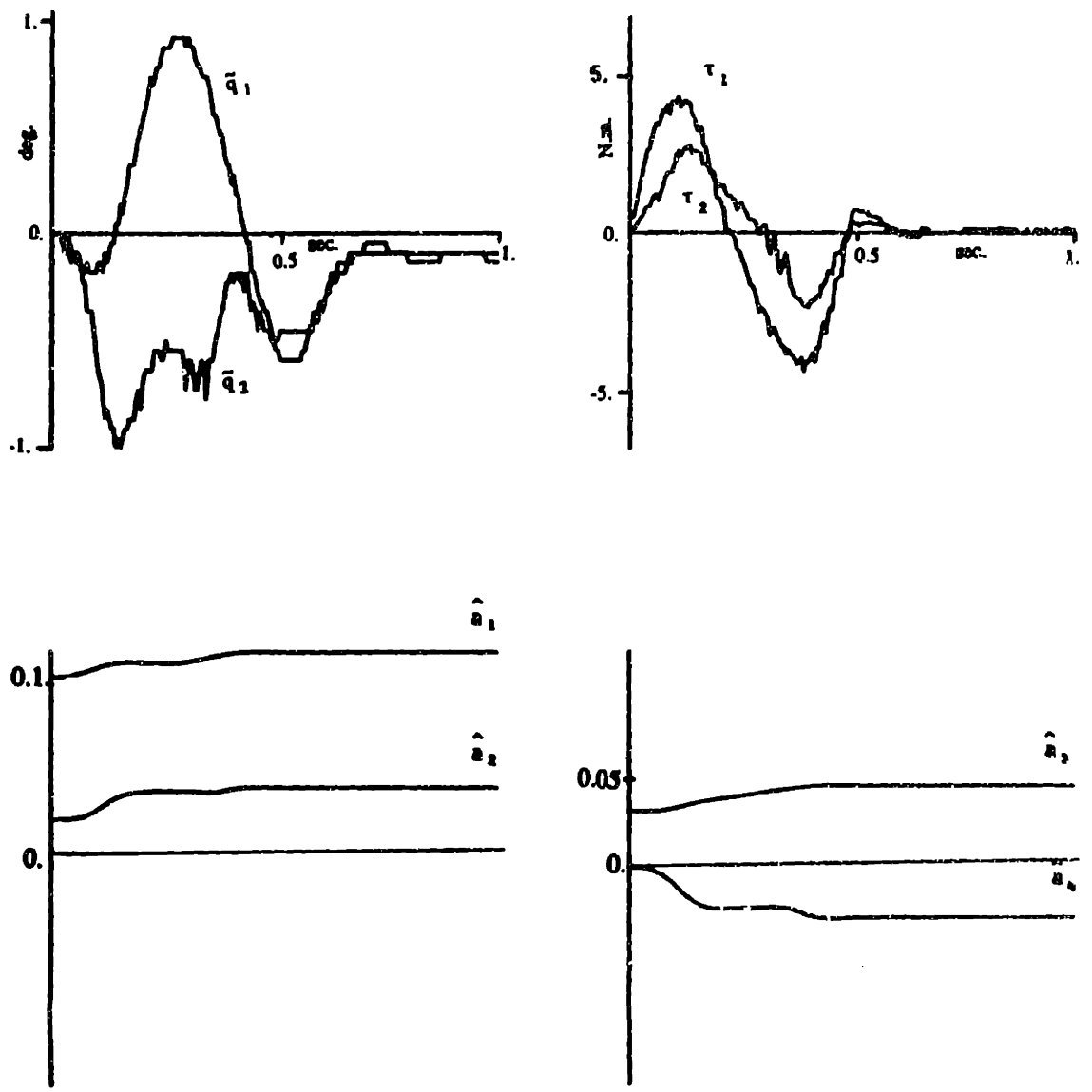


Figure 5.7 : adaptive control with initial parameter information:

(a) position errors $\tilde{q}$; (b) joint torque τ;

(c) estimates $\hat{a}_1, \hat{a}_2$; (d) estimates $\hat{a}_3, \hat{a}_4$

5.5 Comparisons in the Presence of a Large Load

This section studies the effectiveness of the adaptive controller in a realistic setting.

To start with, we note that the purpose of using adaptive control for manipulation is to maintain tracking accuracy in the face of significant parameter uncertainty in load. To emulate such a situation, a large load was attached to the end of the second link of the arm. The load was a clutch which had roughly half the size and weight of the second link. The attachment of this load to the last link was unknown to the robot control system.

The same trajectory was successively controlled by the PD , the computed torque, and the adaptive controllers. The PD controller was identical to the one in section 5.3 and the computed torque and adaptive controllers were the same as the ones in section 5.4, both in design parameters and initial parameters. The results are plotted in Figures 5.8-5.10.

For the PD controller, the maximum tracking errors increase to 10.2^{o} and 7.4^{o}. Even at the end of the last-half-second regulation, the tracking errors are still far from settled down (1.3^{o} and 0.9^{o}, respectively). The maximum tracking errors of the computed torque controller are 1.9^{o} and -4.7^{o} , representing increases of 0.9^{o} and 2.2^{o} . The much larger increase in the second joint is due to the fact that the attached clutch causes a larger increase in the uncertainty of the second joint dynamics than in that of the first joint. The maximum tracking errors for the adaptive controller are now 0.9^{o} and -2.0^{o}, with the second joint error increased by 1.04^{o} but no increase in the first joint error.

When the parameter estimates obtained at the end of this run are used for a

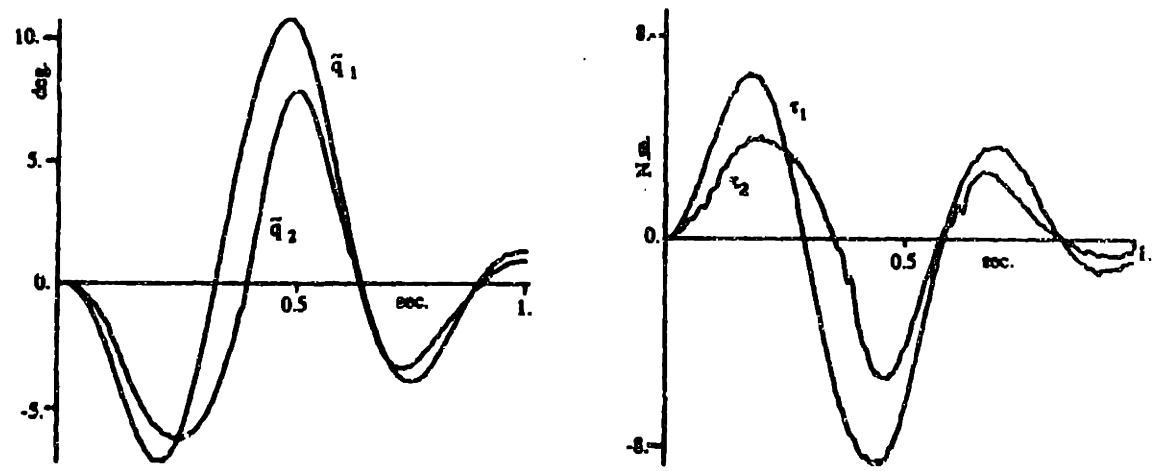


Figure 5.8 : PD control of the large load: (a) position errors $\tilde{q}$; (b) joint torque τ

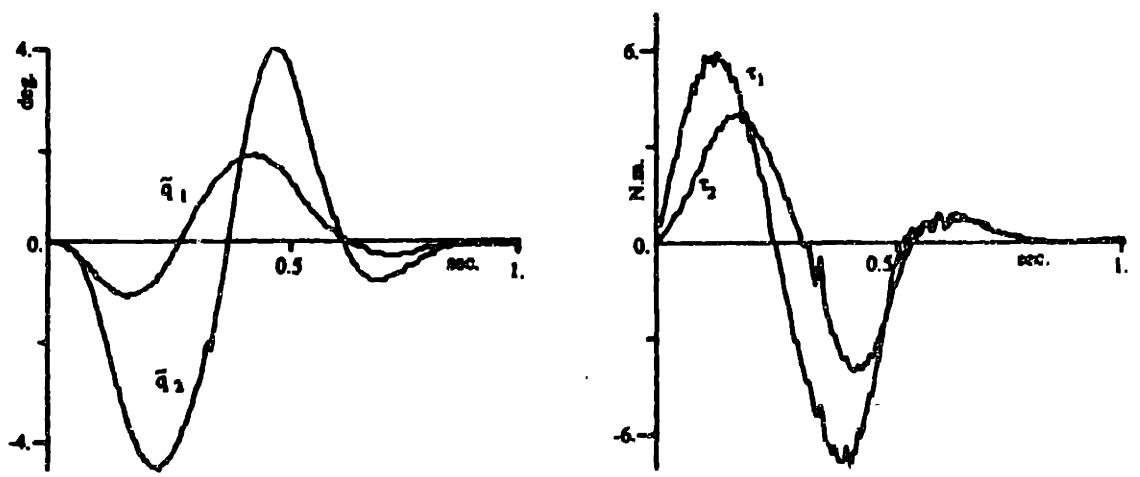


Figure 5.9 : Computed-Torque control of the large load:

(a) position errors $\tilde{q}$; (b) joint torques τ

second run, the maximum errors of the adaptively controlled robot are found to stay within 1^{o} for both joints. The PD and computed-torque controllers would, of

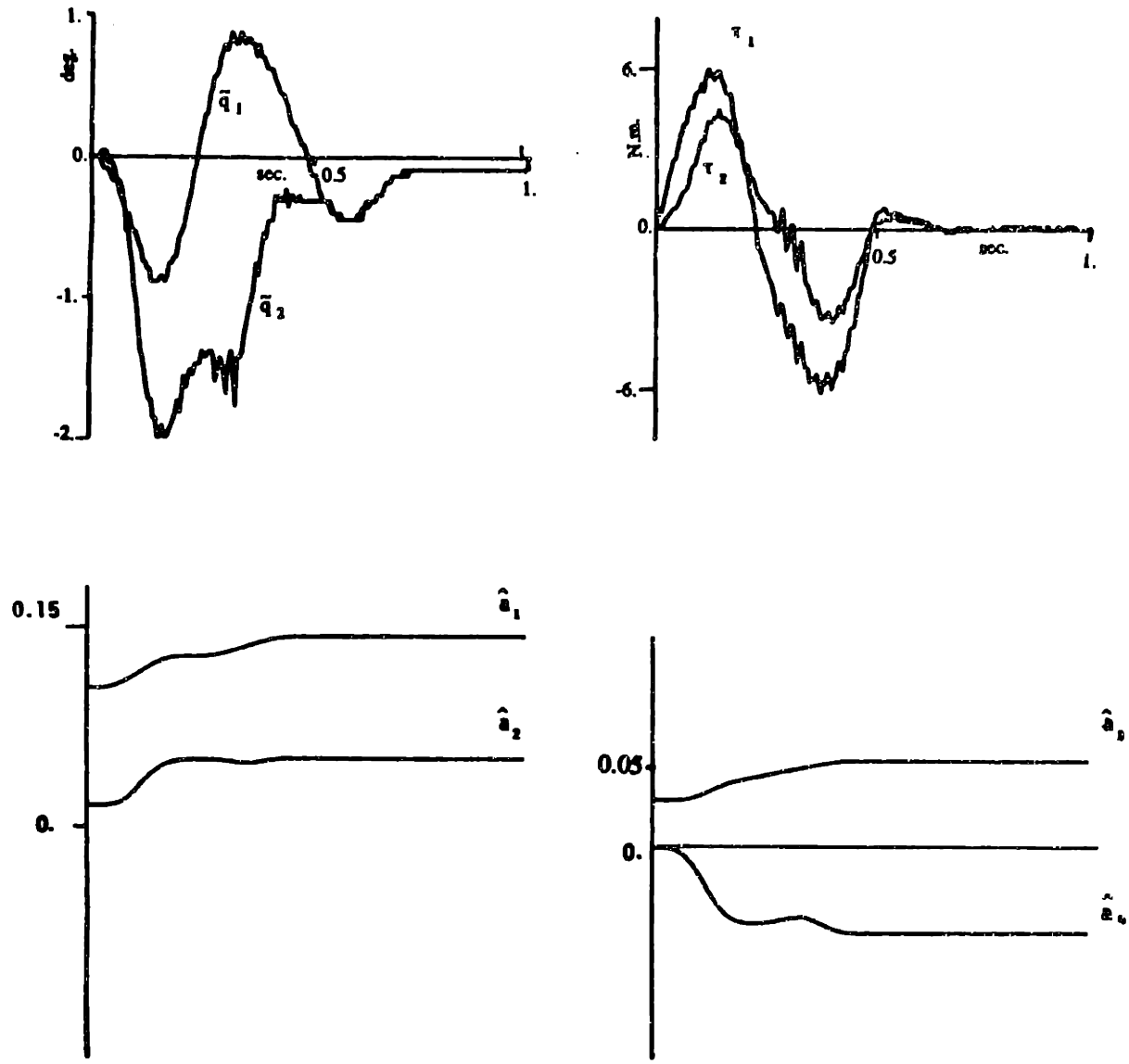


Figure 5.10 : adaptive control of the large load:
(a) position errors $\tilde{q}$; (b) joint torque τ;
(c) estimates $\hat{a}_1, \hat{a}_2$; (d) estimates $\hat{a}_3, \hat{a}_4$

course, essentially repeat their errors. In further adaptive experiments along varied and longer trajectories, long-term parameter drift was found to be virtually absent, due to the stopping of adaptation in the dead zones described in section

5.3.

5.6 Error Sources in the Experiments

The theory in section 2 predicts that the tracking errors of the adaptive controller globally converge to zero, while the tracking errors in the second and later runs along the same trajectory merely stay within the one degree range, without exact convergence. The discrepancy is due to various noises, disturbances and unmodeled dynamics, which are inherent in the experiments but ignored in the theoretical analysis. The tracking errors observed arise from many hardware or software sources. These include, in particular:

a. Arm Modeling Errors: The substantial Couloumb frictions at the motor shafts are not modeled. The Coulomb and viscous frictions at the linkage joints are not compensated at all in the experiments. The neglected frictions are believed to contribute a significant portion of the tracking errors. Furthermore, the vibrational dynamics of the links may also have contributed a certain amount of error.

b. Actuation Errors: For simplicity, the amplifiers and the motors have been modeled as constant gains, while they actually have dynamics of their own, which may not be negligible for this half-second fast operation. Furthermore, preliminary testing indicates an error of about 3% in the amplifier gain specified by the manufacturer and used in this experiment, in addition to a small torque ripple . In the absence of joint torque sensors, no attempt has been made to compensate for the torque inaccuracies.

c. Measurement Errors: Joint velocity measurements contain a considerable amount of error. The tachometer signals are small and therefore are sensitive to noise. The signals after amplification and filtering contains quite severe phase lag.

In addition, the A/D converters for the velocity signals create a certain amount of error. For this high accuracy controller, the relatively low resolution of the optical encoders (0.045^o, i.e., 12 bits/180^o) does not allow numerical differentiation to advantageously replace the tachometer signals.

d. Real-Time Computing Limitations: The relatively low sampling frequency (200 hz) is believed to be one of the major sources of residue tracking errors. To obtain a rough idea of how much tracking errors the computational delay can introduce, let us perform a simple calculation. For this polynomial trajectory, at the instant $t = 0.5$ (sec) when the desired speed reaches its maximum (given by (5-3)) , the change of desired position in one sampling interval is

$$\begin{bmatrix} 300 \\ 262.5 \end{bmatrix} 0.005 = \begin{bmatrix} 1.5^0 \\ 1.31^0 \end{bmatrix}$$

This implies that the desired trajectories could change this much without any compensation reaction from the controller. These reflect the largest effects the computational delay can have on the tracking errors, and this tracking errors have comparable magnitudes as those of the residue tracking errors in the experiments. Thus, computational delay contributes a considerable portion of the tracking errors. Roundoff errors and, especially, sampling limitations, are further sources of error at these high speeds.

5.7 Summary

The proposed adaptive controller was implemented on a 2-DOF robot. The experimental results confirmed the insights gained in the theoretical and simulation studies in Chapter 3 and 4. The adaptive controller was shown to be superior to either PD and computed-torque controllers in practical applications.

Even though the desired trajectories in the experiments were of polynomial type and had little persistent excitation, the adaptive controller has been seen to work well and parameter drift was relatively insignificant due to the incorporation of small dead-zones.

Chapter 6
On-Line Parameter Estimation

On-line parameter estimation is one approach for dealing with parameter uncertainty or parameter variations. On one hand, it can be used to provide parameters for self-tuning control. On the other, it can be combined with the previously derived adaptive controller to yield improved adaptive control performance, as will be shown in the next chapter. It can also be used to monitor load change or slippery in robot hands.

In this chapter, we describe various parameter estimation schemes, emphasizing two new schemes we propose to improve the classical least-square method. Specifically, we shall examine the drawbacks of the standard least-square method, propose two new variations, analyze the performance of the estimators in the presence of disturbance and parameter variation. Note that the results in this chapter can be used to any system identification problems with linearly parametrized models, including those in robotics.

6.1 Introduction

In this section, we shall describe the framework of parameter estimation.

A. Linear Parametrization Model

A fundamental problem often encountered in system identification involves the on-line estimation of the $m\times1$ parameter vector a from the following linear parametrization model:

$$\mathbf{y}(t)=\mathbf{W}(t)\mathbf{a} \tag{6-1}$$

based on the measurement or computed values of the $n\times1$ vector y and the $n\times m$ matrix W. Model (6-1), although simple, is actually quite general. Any linear system can be rewritten in this form after filtering both sides of the system dynamics equation through an exponentially stable filter of proper order, as shown in [*e.g.*, Astrom and Wittenmark, 1989]. For instance, consider the first-order system

$$\dot{y}=-a_1 y+b_1 u \tag{6-2}$$

where the parameters a_1 and b_1 are unknown, and the output y and input u are measured. To avoid $\dot{y}$ in the above equation, let us filter both sides of the equation by $1/(p+\lambda_f)$ (where p is the Laplace operator and λ_f is a known positive constant), and rearranging, which leads to the previous form

$$y(t)=y_f(\lambda_f-a_1)+u_f b_1 \tag{6-3}$$

where

$$y_f=\frac{y}{p+\lambda_f} \qquad u_f=\frac{u}{p+\lambda_f}$$

with the subscript f denotes filtered quantities. Note that, as a result of the

filtering operation, the only unknown quantities in (6-3) are the parameters $(\lambda_f - a_1)$ and b_1.

The nonlinear dynamics of robots can also be put in the form of (6-1), as we now show. From equation (2-5), we can write

$$\tau = Y_1(q, \dot{q}, \ddot{q})\, a \tag{6-4}$$

with Y_1 being a nonlinear matrix function of q, $\dot{q}$ and $\ddot{q}$, and a being a $m \times 1$ vector of equivalent parameters, using the linear parametrization property in section 2.2. This relation cannot be directly used for parameter estimation because of the presence of the unmeasurable joint acceleration $\ddot{q}$. To avoid the joint acceleration in this relation, we can also use the filtering technique. Specifically, let $w(t)$ be the impulse response of a stable, proper filter (for example, for the first order filter $\lambda/(p+\lambda)$, the impulse function is $e^{-\lambda t}$). Then, convolving both sides of (2-5) by w yields

$$\int_0^t w(t-r)\tau(r)\, dr = \int_0^t w(t-r)[\, H\ddot{q} + C\dot{q} + G\,]\, dr \tag{6-5}$$

Using partial integration, the first term on the right-hand side of (6-5) can be rewritten as

$$\int_0^t [\, H\ddot{q}]\, dr = w(t-r)\, H\dot{q}\Big|_0^t - \int_0^t \frac{d}{dr}[w H]\dot{q}\, dr$$

$$= w(0)\, H(q)\dot{q} - w(0)\, H[q(0)]\dot{q}(0) - \int_0^t [w(t-r)\dot{H}\dot{q} - \dot{w}(t-r)\, H\dot{q}]\, dr$$

This means that equation (6-5) can be rewritten as

$$y(t) = W(q, \dot{q})\, a \tag{6-6}$$

where y is the filtered torque and W is the filtered version of Y_1. Thus, the

matrix W can be computed from available measurements of q and $\dot{q}$. The filtered torque y can also be computed (assuming no actuator dynamics) because the torque signals issued by the computer are known.

B. Prediction-Error-Based Estimation

Assume that the parameter vector a in model (6-1) is unknown and to be estimated. A prediction of $y(t)$ can be made based on the estimated parameters $\hat{a}(t)$ and the measurable matrix $W(t)$ at time t

$$\hat{y}(t) = W(t)\hat{a}(t) \tag{6-7}$$

The difference between the predicted value $\hat{y}(t)$ and the measured y is the prediction error

$$e = \hat{y}(t) - y(t) = W\tilde{a} \tag{6-8}$$

where $\tilde{a}(t) = \hat{a}(t) - a$ is the parameter estimation error. Equation (6-8) indicates that the prediction error e, which can be easily computed, contains information on the parameter error. Various methods can be used to extract this information. The estimators to be discussed belong to the prediction-error-based approach, all having the following form of parameter update law

$$\dot{\hat{a}} = -PWe \tag{6-9}$$

where $P(t)$ is a (possibly time-varying) positive definite (*p.d.*) gain matrix. Note that a (possibly time-varying) *p.d.* weighting matrix $R(t)$ can be incorporated in front of e in (6-9), in order to reflect the relative reliability of the prediction errors in different joints and at different times, but it is omitted here for simplicity. The gain matrix P may be generated from different perspectives, leading to

different estimators and performances.

C. A Lemma for Convergence and Robustness

Later discussion will involve the following linear time-varying dynamics

$$\dot{\mathbf{x}}(t) = -\mathbf{A}(t)\mathbf{x}(t) + \mathbf{d}(t) \tag{6-10}$$

where the vector $\mathbf{x}(t)$ is the state, $\mathbf{A}(t)$ a *symmetric* and *uniformly p.d.* matrix, the vector $\mathbf{d}$ denotes disturbance. By *uniformly p.d.*, we mean that the matrix is lower bounded by a constant *p.d.* matrix, *i.e.*,

$$\forall t \geq 0\,,\ \mathbf{A}(t) \geq \varepsilon_o \mathbf{I}$$

where ε_o is a positive constant and $\mathbf{I}$ the identity matrix.

Lemma 6.1: *If the disturbance* $\mathbf{d}(t)$ *in (6-10) is zero, then* $\mathbf{x}(t)$ *converges to zero exponentially; if* $\mathbf{d}(t)$ *is bounded, so is* $\mathbf{x}(t)$.

Proof: In the absence of disturbance, the exponential convergence can be shown using the Lyapunov function $V(t) = \mathbf{x}^T\mathbf{x}$. Since

$$\dot{V}(t) = -2\mathbf{x}^T\mathbf{A}\,\mathbf{x} \leq -\varepsilon_o \mathbf{x}^T\mathbf{x}$$

one has $V(t) \leq V(0)e^{-\varepsilon_o t}$ and therefore $\mathbf{x}$ is exponentially convergent. One can easily show that this exponential convergence in turn implies that the state transition matrix $\mathbf{G}(t,r)$ associated with (6-10) satisfies

$$\|\mathbf{G}(t,r)\| \leq \varepsilon_1\, e^{-\varepsilon_o (t-r)/2}\, \mathbf{I} \tag{6-11}$$

with ε_1 denoting another positive constant. By using the explicit solution

$$\mathbf{x}(t) = \mathbf{G}(t,0)\,\mathbf{x}(0) + \int_0^t \mathbf{G}(t,r)\,\mathbf{d}(r)\,dr \tag{6-12}$$

together with the boundedness of d(t) and (6-11), one easily verifies the boundedness of x(t), with the bound being proportional (in norm) to that of the disturbance. □

D. Least-Square and Its Modifications

The motivation for on-line estimation, instead of off-line estimation, usually stems from the need to deal with parameter variation or sudden parameter change during the operation, which requires that the estimator continuously track or monitor the varying parameters. The well-known standard least-square method (without data forgetting) cannot be used for this purpose, because it tends to gradually turn itself off due to diminishing estimation gain matrix. Several techniques have been studied in the literature to enable the estimator to handle both constant parameters and slowly varying parameters. A simple way of modifying the standard least-square method [Goodwin and Sin, 1984] is to directly reset the gain matrix in standard least-square scheme whenever its largest eigenvalue decreases to a prespecified lower bound. Another popular approach is to incorporate data exponential forgetting into least-square method. Despite the intuitive appeal of data forgetting, the proper choice of forgetting factor is an unresolved issue. If a positive constant forgetting factor is used, which is a seemingly natural choice, then unbounded gain growth may result in the absence of signal persistent excitation [*e.g.*, Astrom, 1980]. This may imply excessive sensitivity to noise and disturbances, as can be seen from (6-9) (any disturbance in the prediction error will be multiplied by the huge gain, leading to violent oscillation of the parameter update). [Irving, 1979; Lozano-Leal and Goodwin, 1985] studied variable forgetting factors leading to constant-trace gain, and other modifications of the gain update. However, such modifications are quite complex

in computation, and the performance of the resulting estimators in the face of parameter variations and disturbances are difficult to analyze.

Based on a detailed examination of the gain wind-up problem associated with least-square estimator with constant forgetting factors, we recently proposed two schemes for maintaining data forgetting while preventing gain wind-up [Li and Slotine, 1987, 1988a,b]. In this chapter, a detailed analysis of the convergence and robustness properties of the new schemes is provided. The first scheme involves a novel and simple way of automatically tuning the forgetting factor according to the norm of the gain matrix. The resulting exponentially-forgetting least square estimator, called bounded-gain-forgetting method, will be shown to have exponentially convergent parameters under *p.e.* signals and upper bounded (by a specified bound) gain matrix regardless of signal excitation. The other scheme is obtained by adding a simple "cushioning" term into the gain update equation, thus leading to upper boundedness of the gain independently of the signal excitation. Section 6.4 analyzes the performance of the new estimators in the face of parameter variation and disturbance. Section 6.5 presents a result stating that the filtered version of a persistently exciting (*p.e.*) signal is also *p.e.* under fairly weak conditions. This result is often useful in establishing global stability conditions for self-tuning and composite adaptive controllers based on the estimators reviewed and proposed in this chapter. Simulation results illustrating the performance of the new estimators are presented in section 6.6. Section 6.7 offers concluding remarks.

6.2 Conventional Estimators

In this section, we briefly review and analyze the three major types of existing estimators. This will not only illustrate the shortcomings of the methods, but also suggest ways of improvements. In sections 6.2-6.4, we assume that the parameter vector **a** is constant. The performance of the estimators in the presence of time-varying parameters **a**(t) is analyzed in section 6.5.

A. Gradient Estimator

In the gradient estimator, the gain matrix is simply held constant

$$\mathbf{P}(t)=\mathbf{P}_o \tag{6-13}$$

where $\mathbf{P}_0$ is a constant *p.d.* matrix. The estimator is called a gradient estimator because the parameter update law can be written as

$$\dot{\hat{\mathbf{a}}} = -\mathbf{P}_0 \frac{\partial [\mathbf{e}^T \mathbf{e}]}{\partial \hat{\mathbf{a}}}$$

with the partial derivative representing the gradient of the squared prediction error. It is well known [Anderson, 1977; Morgan and Narendra, 1977] that the gradient method leads to exponential parameter convergence if the signals are persistently exciting (*p.e.*), by which we mean that there exist positive constants α_1, α_2 and δ, such that

$$\forall t \geq 0, \quad \alpha_1 \mathbf{I} \leq \int_t^{t+\delta} \mathbf{W}^T(r)\,\mathbf{W}(r)\,dr \leq \alpha_2 \mathbf{I} \tag{6-14}$$

The drawbacks of the gradient estimator are that the convergence rate of the estimated parameters is generally small and also difficult to evaluate, and that the estimated parameters tend to be oscillatory for large estimation gain, because of the gradient nature of the search direction.

B. Standard Least-Square Estimators

The least square estimate of the parameters is generated by minimizing the total prediction error

$$J = \int_0^t \| \mathbf{y}(r) - \mathbf{W}(r)\hat{\mathbf{a}}(t) \|^2 dr \tag{6-15a}$$

with respect to $\hat{\mathbf{a}}(t)$. The estiamted paramter $\hat{\mathbf{a}}$ satisfies

$$[\int_0^t \mathbf{W}^T \mathbf{W}\, dr]\, \hat{\mathbf{a}}(t) = \int_0^t \mathbf{W}^T \mathbf{y}\, dr \tag{6-15b}$$

Define

$$\mathbf{P}^{-1}(t) = \int_0^t \mathbf{W}^T(r)\, \mathbf{W}(r)\, dr$$

We obtain

$$\frac{d}{dt}[\mathbf{P}^{-1}(t)] = \mathbf{W}^T(t)\, \mathbf{W}(t) \tag{6-16}$$

Differentiating (6-15b), we find that the parameter update satisfies (6-9), with P(t) being the gain matrix. In using (6-9) and (6-16) for on-line estimation, we have to provide an initial parameter value and an initial gain value to the estimator. Based on a Kalman-filter interpretation of the least-square estimator, the initial gain P(0) should be chosen to represent the covariance of the initial parameter estimates $\hat{\mathbf{a}}(0)$. In the implementation of this estimator and those in later sections, the identity relation

$$\mathbf{P}^{-1}\dot{\mathbf{P}}(t) + \frac{d}{dt}[\mathbf{P}^{-1}(t)]\,\mathbf{P}(t) = 0$$

may be used to avoid matrix inversion, leading to the following gain update

$$\dot{\mathbf{P}}(t) = -\mathbf{P}\mathbf{W}^T\mathbf{W}\mathbf{P}$$

The behaviour of the estimator can be best understood by solving the differential equations (6-9) and (6-16). From (6-16), (6-9) and (6-8), one easily shows that

$$\mathbf{P}^{-1}(t) = \mathbf{P}^{-1}(0) + \int_0^t \mathbf{W}^T(r)\,\mathbf{W}(r)\,dr \tag{6-19}$$

$$\frac{d}{dt}\left[\mathbf{P}^{-1}(t)\,\tilde{\mathbf{a}}(t)\right] = 0$$

Thus,

$$\tilde{\mathbf{a}}(t) = \left[\mathbf{P}^{-1}(0) + \int_0^t \mathbf{W}^T(r)\,\mathbf{W}(r)\,dr\right]^{-1}\ \mathbf{P}^{-1}(0)\ \tilde{\mathbf{a}}(0) \tag{6-20}$$

If W is such that

$$\lambda_{min}\left\{\int_0^t \mathbf{W}^T\mathbf{W}\,dr\right\} \to \infty \quad \text{as} \quad t \to \infty \tag{6-21}$$

where $\lambda_{min}[\cdot]$ denotes the smallest eigenvalue of its argument, then the gain matrix converges to zero, and the estimated parameters asymptotically (but unsually not exponentially) converge to the true parameters. Note that the "infinite-integral" condition (6-21) is a weaker condition than the persistent excitation (6-14). To prove this, one notes that, for any positive integer k,

$$\int_0^{k\delta+\delta} \mathbf{W}^T\mathbf{W}\,dr = \sum_{i=0}^{i=k} \int_{i\delta}^{i\delta+\delta} \mathbf{W}^T\mathbf{W}\,dr \geq k\alpha_1\,\mathbf{I}$$

Thus, if W is persistently exciting, (6-21) is satisfied, and $\mathbf{P}^{-1} \to \infty$ and $\mathbf{P} \to 0$. This implies that the standard least-square estimator cannot be used to estimate time-varying parameters because of the tendency of the estimator to gradually turn itself off (implied by $\mathbf{P} \to 0$).

C. Exponentially-Forgetting Least-Square Estimators

If exponential forgetting of data is incorporated into least-square

estimation, by minimizing

$$J = \int_0^t e^{-\int_r^t \lambda(r)\,dr} \| y(s) - W(s)\hat{a}(t) \|^2 ds$$

instead of (6-15), then the parameter update law is generated by (6-9), and the gain update law by

$$\frac{d}{dt}[P^{-1}] = -\lambda(t)P^{-1} + W^T(t)W(t) \qquad (6\text{–}22)$$

where $\lambda(t) \geq 0$ is the time-varying forgetting factor. The estimator gain in the least-square estimator with any constant or time-varying factor can be explicitly solved to be

$$P^{-1}(t) = P^{-1}(0)e^{-\int_0^t \lambda(r)\,dr} + \int_0^t exp[-\int_r^t \lambda(v)\,dv]\, W^T(r)\, W(r)\, dr \qquad (6\text{-}23)$$

To solve the explicit form of the parameter error, let us note that one can easiy obtain

$$\frac{d}{dt}[P^{-1}\tilde{a}] = -\lambda P^{-1}\tilde{a}$$

Therefore,

$$\tilde{a}(t) = exp[-\int_0^t \lambda(r)\,dr]\, P(t)P^{-1}(0)\tilde{a}(0) \qquad (6\text{-}24)$$

That is,

$$\tilde{a}(t) = [P^{-1}(0) + \int_0^t exp(\int_0^r \lambda(v)\,dv)\, W^T(r)\, W(r)\, dr]^{-1} P^{-1}(0)\tilde{a}(0)$$

Comparing this and (6-20), and noting that $exp(\int_0^t \lambda(v)\,dv) \geq 1$, one sees why exponential forgetting improves parameter convergence over standard least square. It also shows that the "infinite integral" condition (6-21) for standard least-squares guarantees the asymptotic convergence of the estimated parameters.

Unlike the SLS method, the exponential forgetting leads to exponential convergence of the estimated parameters, provided that $\lambda(t)$ is larger than or equal to a positive constant, and signals are *p.e.*. Specifically, assume that $\lambda(t) \geq \lambda_o$ with λ_o being a positive constant. Then, we have, from (6-22),

$$\frac{d}{dt}[\mathbf{P}^{-1}] = -\lambda_o \mathbf{P}^{-1} + \mathbf{W}^T\mathbf{W} + (\lambda(t) - \lambda_o)\mathbf{P}^{-1}$$

$$\mathbf{P}^{-1}(t) = \mathbf{P}^{-1}(0)e^{-\lambda_o t} + \int_0^t e^{-\lambda_o(t-r)}\{\mathbf{W}^T\mathbf{W} + (\lambda(t) - \lambda_o)\mathbf{P}^{-1}\}\,dr$$

This guarantees from (6-14) that $\mathbf{P}^{-1}(t) \geq e^{-\lambda_o \delta}\alpha_1 \mathbf{I}$ and, accordingly, that

$$\mathbf{P}(t) \leq \frac{e^{\lambda_o \delta}}{\alpha_1}\mathbf{I}$$

for $t \geq \delta$. This and (6-24) show the exponential convergence of $\tilde{\mathbf{a}}(t)$ to zero with a rate of at least λ_o. It is interesting that the exponential convergence rate of the estimated parameters is the same as the forgetting factor in case a constant forgetting factor is used. However, a constant forgetting factor may lead to diminishing magnitude in certain directions of $\mathbf{P}^{-1}$ (and accordingly, unbounded magnitude in certain directions of $\mathbf{P}$) in the absence of *p.e.*, due to the exponential decaying components in (6-23). Unboundedness (or even large magnitude) of the gain matrix is undesirable since it implies that the disturbance and noise in the prediction error may, through the update law (6-9), lead to violent oscillation of the estimated parameters, as seen in many experimental results involving insufficiently-rich signals in set-point control [*e.g.*, Suzuki, 1988].

6.3 New Estimators

A good estimator should have the following characteristics:

- exponential convergence and non-vanishing P in the presence of *p.e.*
- upper bounded P and $\tilde{\mathbf{a}}$ in the absence of *p.e.*

In the presence of *p.e.*, the signals contain rich information about the true parameters and a good estimator should be able to identify constant parameters exponentially fast. A vanishing gain means that the estimator can not estimate time-varying parameters. The upper boundedness of the gain matrix ensures that the estimator will not be too sensitive to measurement noise in the absence of *p.e.* A zero forgetting factor (standard least-square) results in the lack of the first quality because of its vanishing gain, while a constant positive forgetting factor leads to the lack of the second quality because of its possible gain explosion in the absence of *p.e.* Since an estimator may encounter trajectories with various levels of *p.e.*, it is desirable to tune the forgetting factor automatically according to the excitation of the signals in such a way that the estimator has both afore-mentioned qualities.

6.3.1 Bounded-Gain-Forgetting (BGF) Estimator

We recently proposed the following forgetting factor tuning technique

$$\lambda(t)=\lambda_0\left(1-\frac{\|P\|}{k_0}\right) \tag{6-26}$$

with λ_0 and k_0 being positive constants representing the maximum forgetting rate and prespecified bound for gain matrix magnitude, respectively. The forgetting factor in (6-26) implies forgetting the data with a factor λ_0 if the norm

of P is small (indicating strong *p.e.*), reducing the forgetting speed if the norm of P becomes larger and suspends forgetting if the norm reaches the specified upper bound. Since a larger value of λ_0 means faster forgetting (which implies both stronger ability in following parameter variations, but also more oscillations in the estimated parameters due to shorter-time averaging of noisy data points), the choice of λ_0 represents a tradeoff between the speed of parameter tracking and the level of estimated parameter oscilation. The gain bound k_0 affects the speed of parameter update and also the effects of disturbance in the prediction error, thus involving a similar tradeoff. To be consistent with our gain-bounding intention, we choose $\|\mathbf{P}(0)\| \le k_0$ (hence $\mathbf{P}(0) \le k_0 \mathbf{I}$). We shall refer to the least-square estimator with the forgetting factor (6-26) as the bounded-gain-forgetting (BGF) estimator, because the norm of the gain matrix can be shown to be upper bounded by the pre-specified constant k_0 regardless of persistent excitation.

We now show that the form (6-26) of forgetting factor variation guarantees that the resulting gain matrix satisfies $\mathbf{P}(t) \le k_0 \mathbf{I}$ and, accordingly, that $\lambda(t) \ge 0$. With the forgetting factor form (6-26), the gain update equation (6-22) can be expressed as

$$\frac{d}{dt}[\mathbf{P}^{-1}] = -\lambda_0 \mathbf{P}^{-1} + (\lambda_0/k_0)\|\mathbf{P}\|\mathbf{P}^{-1} + \mathbf{W}^T\mathbf{W} \qquad (6\text{–}27)$$

This leads to

$$\mathbf{P}^{-1}(t) = \mathbf{P}^{-1}(0)e^{-\lambda_0 t}dr + \int_0^t e^{-\lambda_0(t-r)}\left[\frac{\lambda_0}{k_0}\|\mathbf{P}\|\|\mathbf{P}^{-1}\| + \mathbf{W}^T\mathbf{W}\right]$$

$$\ge (\mathbf{P}^{-1}(0) - k_0^{-1}\mathbf{I})e^{-\lambda_0(t-r)} + (1/k_0)\mathbf{I} + \int_0^t e^{-\lambda_0(t-r)}\mathbf{W}^T\mathbf{W}\,dr \qquad (6\text{–}28)$$

where we used the inequality $\|\mathbf{P}(t)\|\mathbf{P}^{-1}(t) \ge \mathbf{I}$, obtained from the fact that

$$\|\mathbf{P}\|\mathbf{P}^{-1}-\mathbf{I}=\mathbf{P}^{-1/2}[\,\|\mathbf{P}\|\mathbf{I}-\mathbf{P}\,]\mathbf{P}^{-1/2}\geq 0$$

Note that $\|\mathbf{P}(0)\|\leq k_0$ guarantees the positive-definiteness of $(\mathbf{P}^{-1}(0)-k_0^{-1}\mathbf{I})$, therefore

$$\forall t\geq 0\,,\ \mathbf{P}^{-1}(t)\geq \frac{1}{k_0}\mathbf{I}$$

hence $\mathbf{P}(t)\leq k_0\mathbf{I}$, and, from (6-26), $\lambda(t)\geq 0$.

If W(t) is *p.e.* as defined by (6-14), we can further show that $\lambda(t)\geq\lambda_1>0$, and, thus, the estimated parameters are exponentially convergent. To show this, note that, from (6-28) and (6-14),

$$\mathbf{P}^{-1}(t)\geq [\frac{1}{k_0}+e^{-\lambda_0\delta}\alpha_1\,]\mathbf{I}$$

$$\mathbf{P}(t)\leq\frac{k_0}{1+k_0\alpha_1 e^{-\lambda_0\delta}}\mathbf{I} \qquad (6\text{–}29)$$

This in turn leads to the uniform lower boundedness of the forgetting factor by a positive constant,

$$\lambda(t)=\frac{\lambda_0}{k_0}(k_0-\|\mathbf{P}\|)\geq\frac{\lambda_0 k_0\alpha_1 e^{-\lambda_0\delta}}{1+k_0\alpha_1 e^{-\lambda_0\delta}}=\lambda_1 \qquad (6\text{–}30)$$

This implies the exponential convergence of the estimated parameters, as pointed out in subsection 6.2. Note that, if W is strongly *p.e.*, *i.e.*, α_1 is very large, $\lambda(t)\approx\lambda_0$.

Under *p.e.*, one can also show that P(t) is uniformly lower bounded by a constant *p.d.* matrix, a property which is desirable for estimating time-varying parameters Indeed, from (6-23) and (6-30),

$$\mathbf{P}^{-1}(t) \le \mathbf{P}^{-1}(0) + \int_0^t exp[-\lambda_1 (t-\tau)]\, \mathbf{W}^T \mathbf{W}\, dr$$

The second term on the right hand side can be regarded as the output of the stable filter

$$\dot{\mathbf{M}} + \lambda_2 \mathbf{M} = \mathbf{W}^T \mathbf{W} \tag{6-31}$$

hence M is bounded if W is bounded. Thus, from (6-31) and (6-29), if W is *p.e.* and upper bounded, P will be upper bounded and lower bounded uniformly, *i.e.*,

$$k_2 \mathbf{I} \le \mathbf{P}(t) \le k_1 \mathbf{I}$$

where $0 < k_2 < k_1 < k_0$.

The properties of the BGF estimator are summarized below:

Theorem 6.1: *In the bounded-gain-forgetting estimator, the parameter errors and the gain matrix are always upper bounded; If* W *is persistently exciting, the estimated parameters converge exponentially and* P*(t) is upper and lower bounded uniformly by positive definite matrices.*

Note that if a prespecified lower bound is desired for the smallest eigenvalue of P for tracking parameter variations, one may simply stop updating P when the computed gain update goes below the specified bound, and resume updating P when the computed update goes back up. This amounts to temporarily run the estimator in a gradient mode.

6.3.2 Cushioned-Floor Estimator

In view of the fact that the problem with the non-zero forgetting factor is the possibility of $\mathbf{P}^{\#-1}$ diminishing, we may, as an alternative to variable-forgetting approach, use the following modification

$$\frac{d}{dt}(\mathbf{P}^{-1})(t) = -\lambda(t)(\mathbf{P}^{-1} - \mathbf{K}_0^{-1}) + \mathbf{W}^T\mathbf{W} \qquad (6\text{–}33)$$

where $\mathbf{K}_0$ is a symmetric *p.d.* matrix specifying the upper bound of the gain matrix $\mathbf{P}$, and $\lambda(t)$ is a constant or time-varying positive forgetting factor independent of $\mathbf{P}$. $\lambda(t)$ is chosen to satisfy $\lambda_o \le \lambda(t) \le \lambda_1$ with λ_0 and λ_1 denoting two positive constants. The first term in (6-33) acts as a "cushion" on the floor $\mathbf{P}^{-1}=\mathbf{0}$, so that $\mathbf{P}^{-1}$ is maintained higher than $\mathbf{K}_0^{-1}$, *i.e.*, $\mathbf{P}^{-1} \ge \mathbf{K}_o^{-1}$, $\mathbf{P} \le \mathbf{K}_o$. Indeed, the solution of (6-33) is

$$\mathbf{P}^{-1}(t) - \mathbf{K}_o^{-1} = [\mathbf{P}^{-1}(0) - \mathbf{K}_o^{-1}]e^{-\int_0^t \lambda(r)\,dr} + \int_0^t e^{-\int_r^t \lambda(v)\,dv}\,\mathbf{W}^T(r)\,\mathbf{W}(r)\,dr \quad (6\text{-}34)$$

which, given $\mathbf{P}(0) \le \mathbf{K}_o$, shows that $\mathbf{P}^{-1}(t) \ge \mathbf{K}_o^{-1}$, and accordingly, $0 \le \mathbf{P}(t) \le \mathbf{K}_o$. Unlike the BGF estimator, the estimator resulting from this gain update law, called cushioned-floor (CF) estimator, is not a special case of least-square estimation, but rather includes least-square-type estimators as special cases, with $\lambda(t) = 0$ corresponding to the SLS method, $\mathbf{K}_o \to \infty$ to the exponentially-forgetting least-square method, and $\lambda(t) \to \infty$ to the gradient method with $\mathbf{P}(t) = \mathbf{K}_o$. $\mathbf{P}$ will be uniformly lower bounded by a *p.d.* matrix if $\mathbf{W}$ is bounded. A prespecified *p.d.* lower bound on $\mathbf{P}$ may also be achieved by temporarily stopping the gain update, as mentioned for the BGF estimator.

Theorem 6.2: *The parameter errors from the CF estimators are guaranteed to be bounded, and the gain matrix is guaranteed to be uniformly*

upper bounded by a pre-specified p.d. matrix. Furthermore, if W *is p.e., the estimated parameters converge exponentially, and the gain matrix* $\mathbb{P}(t)$ *is upper and lower bounded uniformly by positive definite matrices.*

Proof: Consider the following Lyapunov function candidate

$$V = \frac{1}{2}\tilde{\mathbf{a}}^T \mathbf{P}^{-1}\tilde{\mathbf{a}}$$

Its derivative is

$$\dot{V} = \tilde{\mathbf{a}}^T \mathbf{P}^{-1}\dot{\hat{\mathbf{a}}} + \frac{1}{2}\tilde{\mathbf{a}}^T \frac{d}{dt}[\mathbf{P}^{-1}]\tilde{\mathbf{a}} = -\lambda\, \tilde{\mathbf{a}}^T [\mathbf{P}^{-1} - \mathbf{K}_0^{-1}]\, \tilde{\mathbf{a}}$$

Since $\mathbf{P}^{\#-1}$ is lower bounded by $\mathbf{K}_0^{-1}$ and $\mathbf{P}^{-1}(t) - \mathbf{K}_o^{-1}$ is a symmetric and positive semi-definite matrix, $\dot{V} \le 0$, and, thus, $\tilde{\mathbf{a}}(t)$ is bounded. If W is *p.e.*, then from (6-34) and (6-14),

$$\forall t \ge \delta \quad \mathbf{P}^{-1} - \mathbf{K}_o^{-1} \ge \alpha_1 e^{-\lambda_o \delta}\, \mathbf{I} \tag{6-35}$$

Thus, one can easily show that V, and accordingly $\tilde{\mathbf{a}}$, converges exponentially to zero, with a rate depending on λ_0, δ, and α_1.

By noting that the last term on the right-hand side of (6-34) is smaller than or equal to the output of a stable first-order filter with bandwidth λ_0, *i.e.*

$$\int_0^t e^{-\int_r^t \lambda(v)\,dv}\, \mathbf{W}^T(r)\, \mathbf{W}(r)\, dr \le \int_0^t e^{-\lambda_o (t-r)}\, \mathbf{W}(r)\, \mathbf{W}(r)\, dr$$

one can also conclude that $\mathbf{P}^{-1}(t)$ is uniformly upper bounded and P(t) uniformly lower bounded if W(t) is bounded. □

Similarly to the BGF case, the design parameters λ_0 and $\mathbf{K}_0$ should be chosen to reflect the speed of parameter variation and the noise characteristics. For processes with time-varying noise characteristics, one may want to use time-

varying bounds on the gain matrix. For example, if we want $\mathbf{P}(t) \leq k(t)\mathbf{I}$, with k(t) being a scalar positive function, we can choose the gain update law to be

$$\frac{d}{dt}[\mathbf{P}^{-1}(t) - k^{-1}(t)\mathbf{I}] = -\lambda(t)[\mathbf{P}^{-1}(t) - k^{-1}(t)] + \mathbf{W}^T\mathbf{W} \tag{6-36}$$

i.e.,

$$\frac{d}{dt}[\mathbf{P}^{-1}(t)] = -\lambda(t)[\mathbf{P}^{-1}(t) - k^{-1}(t)] + \mathbf{W}^T\mathbf{W} - \frac{\dot{k}}{k^2}\mathbf{I} \tag{6-37}$$

6.4 Robustness to Disturbances and Parameter Variations

The previous analysis characterizes the basic features of the BGF and CF estimators. However, the analysis is based on two assumptions: constancy of the true parameters and absence of disturbances. In practice, the true parameters are usually time-varying, which is one of the main original reasons for on-line estimation rather than off-line estimation, and furthermore the real dynamics always contains disturbances such as noise and unmodeled friction forces. Can parameter variations and disturbances drive the parameter errors to infinity despite the exponential convergence result obtained before?

This section provides some reassuring answers to this question. The central result states that the parameter estimation error in BGF and CF estimators will be bounded as long as the true parameters have bounded rates of variation, the disturbance (in additive form) has bounded magnitude, and W is *p.e.* and bounded. Specifically, we consider the following form of system dynamics

$$\mathbf{y}(t) = \mathbf{W}(t)\mathbf{a}(t) + \mathbf{d}(t) \tag{6-38}$$

where $\mathbf{d}(t)$ is the bounded disturbance and the time derivative of $\mathbf{a}(t)$ is bounded, *i.e.*,

$$\|\mathbf{d}(t)\| \le B_1 \qquad \|\dot{\mathbf{a}}(t)\| \le B_2$$

with B_1 and B_2 denoting two positive constants. Note that the gain matrix update is not affected by the introduction of parameter variation and of the above additive disturbance.

Let us analyze the BGF estimator first. Suppose the estimator has the parameter and gain update laws (6-9), (6-22) and (6-26) as before. Then, from (6-7) and (6-38), the prediction error is

$$\mathbf{e}(t) = \hat{\mathbf{y}} - \mathbf{y} = \mathbf{W}\tilde{\mathbf{a}} - \mathbf{d} \tag{6-39}$$

The relation between the parameter estimation error and disturbance and parameter variation is given by the interesting first-order expression

$$\frac{d}{dt}[\mathbf{P}^{-1}\tilde{\mathbf{a}}] + \lambda(t)\mathbf{P}^{-1}\tilde{\mathbf{a}} = -\mathbf{P}^{-1}\dot{\mathbf{a}} + \mathbf{W}\,\mathbf{d} \tag{6–40}$$

obtained from (6-9), (6-22), and (6-39). Equation (6-40) describes a first-order filter of bandwidth $\lambda(t)$ with input $(-\mathbf{P}^{-1}\dot{\mathbf{a}} + \mathbf{W}\,\mathbf{d})$ and output $\mathbf{P}^{-1}(t)\tilde{\mathbf{a}}(t)$.

If $\mathbf{W}$ is *p.e.* and bounded, then $\mathbf{P}^{-1}(t)$ will be upper bounded, and $\lambda(t)$ lower bounded by a positive number λ_0 as shown in section 6.3. Then, the filter output is easily shown to be bounded, which, together with the upper boundedness of $\mathbf{P}$, shows the boundedness of the parameter error $\tilde{\mathbf{a}}$, *i.e.*,

$$\|\tilde{\mathbf{a}}(t)\| \le \gamma_1 B_1 + \gamma_2 B_2$$

where the positive constants γ_1 and γ_2 depend on the design parameters k_0 and λ_0, and on the constants α_1, α_2, and δ, which quantify the *p.e.* strength as given

by (6-14).

The same global stability result can be proven for the CF estimator. Now,

$$\frac{d}{dt}[\mathbf{P}^{-1}\tilde{\mathbf{a}}] + \lambda(t)(\mathbf{P}^{-1} - \mathbf{K}_o^{-1})\tilde{\mathbf{a}} = -\mathbf{P}^{-1}\dot{\mathbf{a}} + \mathbf{W}\,\mathbf{d} \qquad (6\text{–}41)$$

Multiplying both sides by the *p.d.* matrix $\mathbf{K}_0^{1/2}$ leads to

$$\frac{d}{dt}[\mathbf{K}_o^{1/2}\mathbf{P}^{-1}\tilde{\mathbf{a}}] + \mathbf{A}(t)[\mathbf{K}_0^{1/2}\mathbf{P}^{-1}\tilde{\mathbf{a}}] = \mathbf{d}(t)$$

This is in the form (6-10) with

$$\mathbf{D}(t) = \mathbf{K}_0^{-1}[-\mathbf{P}^{-1}\dot{\mathbf{a}} + \mathbf{W}\,\mathbf{d}]$$

$$\mathbf{A}(t) = \lambda(t)\mathbf{K}_o^{-1/2}[\mathbf{K}_o - \mathbf{P}]\mathbf{K}_o^{-1/2}$$

In the presence of *p.e.*, from (6-35)

$$\mathbf{P}(t) \le [\mathbf{K}_o^{-1} + \alpha\mathbf{I}]^{-1}$$

with α being a positive constant. This implies that

$$\mathbf{P}(t) \le [\mathbf{I} + \alpha\mathbf{K}_0]^{-1}\mathbf{K}_0 \le \frac{1}{1+\alpha k_o}\mathbf{K}_o$$

with k_0 being the smallest eigenvalue of $\mathbf{K}_0$. Therefore, A(t) is a symmetric and uniformly *p.d.*, matrix, *i.e.*

$$\mathbf{A}(t) \ge \frac{1}{1+\alpha k_o}\mathbf{I}$$

Lemma 6.1 then indicates the boundedness of parameter error. We can summarize the above global stability results as

Theorem 6.3: *If the parameter variation rates, the signal matrix* $\mathbf{W}(t)$*, and*

the disturbance are all bounded, then the parameter estimation errors in BGF and CF estimation are guaranteed to be bounded under persistent excitation.

In other words, the estimated parameters $\hat{a}(t)$ will always remain in a neighborhood of the true parameters a(t), the size of which is proportional to the bound B_1 of disturbance and the bound B_2 on the parameter rate of variation. This represents a fairly powerful result, since the algorithms thus provide bounded-error estimates even for parameters linearly increasing or decreasing with time (to infinity), as long as the signals in W are *p.e.*. If W is strongly *p.e.* (which implies $\lambda(t) \approx \lambda_o$ as indicated in section 3), and $\dot{a}(t)$ changes only slowly, then (6-40) suggests that the steady state estimation error is, in the absence of disturbance,

$$\lambda_o P^{-1} \tilde{a} \approx -P^{-1} \dot{a}$$

that is,

$$\tilde{a}(t) \approx -\dot{a}(t)/\lambda_o$$

This implies that the estimated parameters $\hat{a}(t)$ lag behind the true parameters a(t) by $(\dot{a}(t)/\lambda_o)$. The faster the data forgetting (λ_o larger), or the slower the parameter variation ($\dot{a}(t)$ smaller), the smaller the estimation error. Similar intuitive results exist for CF estimator.

One important way of using theorem 6.3 is to regard the disturbance as a time-varying parameter and augment the parametrization accordingly. Theorem 6.3 suggests that the estimated parameters will be bounded as long as the disturbance has a bounded derivative (even if the disturbance has an unbounded magnitude). In the case that the disturbance is composed of a slowly-varying but possibly large-magnitude part and a small-magnitude but fast-varying part, the extra parameter can be regarded as characterizing the slowly-varying part, and the

high frequency, small magnitude components of the disturbance can be considered as the nominal disturbance.

What will happen if W is not *p.e.*? The estimators can no longer guarantee the boundedness of parameter errors in the presence of parameter variation and disturbances. But P (being independent to the additive disturbances and parameter variations) is bounded and the parameter errors cannot diverge too fast. Using simulations, onve can show that parameter estimates may drift slowly when there is measurement noise. One effective technique to relieve the parameter drift problem is to use dead zones.

6.5 Effects of Filtering on Persistent Excitation

It is clear so far that the exponential convergence of estimated parameters in the previous estimators hinge upon the persistent excitation of W in the model (6-1). Note that model (6-1) usually results from dividing both sides of the original system dynamics by an exponentially stable polynomial $A(p)$. The original dynamics can be written in the form

$$y_1 = W_1 a$$

with (in the frequency domain) $y_1 = A(p)y$ and $W_1 = A(p)W$. It is interesting to see what properties the original matrix W_1 must have in order for the filtered version W to be *p.e.* and for the parameters to be convergent. In the following, we prove that the persistent excitation and uniform continuity of W_1 are sufficient to guarantee the *p.e.* of the filter output W. In the case of the first order plant (6-2), because the original dynamics can be written as

$$\dot{y} + \lambda_f y = [\lambda_f - a_1]y + b_1 u$$

this implies that $(y, u)^T$ should be persistently exciting and uniformly continuous. In the case of the robotics, this implies that Y_1 in (6-4) should satisfy the *p.e.* and *u.c.* conditions.

This result is particularly useful in establishing the exponential convergence condition for our composite and self-tuning adaptive controllers for robot manipulators [Slotine and Li, 1987, 1988; Li and Slotine, 1988]. The uniform continuity (*u.c.*) requirement is intuitively reasonable because it means that the input to the exponentially stable and strictly proper (hence low-pass) filter is smooth, and therefore that its excitation will be largely retained through the filter. Lack of uniform continuity suggests that the signal has sharp fluctuations that the low-pass filtering may wipe out, thus leading to possible loss of persistent excitation. In the special case where the input signal is composed of superimposed sinusoids and the plant is linear, the following results are obvious since the output of the filter is composed of the same frequencies.

Lemma 6.2: *Let u(t) be the uniformly continuous and persistently exciting signal, and $x = L(p)u$, with $L(p)$ being an exponentially stable, strictly proper and minimum phase linear transfer function. Then x(t) is also u.c. and p.e.*

Proof: By persistent excitation of u(t), we mean that there exists positive constants δ_0, α_1 and α_2, such that $\forall r \geq 0$

$$\alpha_1 \leq \int_r^{r+\delta_0} u^2(t)\,dt \leq \alpha_2 \tag{6-42}$$

The *u.c.* of *u*, together with the right-hand-side inequality, can be used to show the boundedness of *u* by contradiction, by noting that an unbounded yet *u.c.* function must violate the right-hand-side inequality at some points.

The boundedness of $u(t)$ guarantees the uniform boundedness and *u.c.* of

the output $x(t)$, because of the exponential stability and strict properness of the filter [Desoer and Vidyasagar, 1975]. In section 6.9, it is shown that, $\forall r \geq 0$, there exist positive constant α_3 independent of r, such that, with the same δ_0 as that in (6-42),

$$\int_r^{r+\delta_0} x^2(t)\,dt \geq \alpha_3 \tag{6-43}$$

We conclude from (6-43) and the uniform boundedness of x(t) that x(t) is also *p.e.*. □

We now extend this result to the vector and matrix cases.

Theorem 6.5: *If a matrix (or vector)* $\mathbf{U}(t)$ *is p.e. and uniformly continuous, the matrix (or vector)* $\mathbf{X}(t)$ *is obtained from* $\mathbf{x} = L(p)\mathbf{u}$, *with* $L(p)$ *being a exponentially stable, strictly proper, and minimum phase linear transfer function. The* $\mathbf{x}$ *is also p.e. and u.c.*

Proof: Let U(t) be the *u.c.* and *p.e.* input and X(t) be the output. The *p.e.* of U implies that there exist positive constants δ, β_1 and β_2 such that $\forall r \geq 0$ and for any constant unit column vector v

$$\beta_1 \leq \int_r^{r+\delta} \mathbf{v}^T \mathbf{U}^T(t)\mathbf{U}(t)\mathbf{v}\,dt \leq \beta_2 \tag{6-44}$$

Now we proceed to show that there exist positive constants β_3 and β_4 such that $\forall r \geq 0$, and the same interval length δ,

$$\beta_3 \mathbf{I} \leq \int_r^{r+\delta} \mathbf{v}^T \mathbf{X}^T(t)\mathbf{X}(t)\mathbf{v}\,dt \leq \beta_4 \mathbf{I} \tag{6-45}$$

The right-hand-side inequality in (6-45) can be established as in lemma 6.2 based on the *u.c.* and *p.e.* of U and the filter's properties. Only the left-hand-side inequality has to be proven below.

If U and X are vectors, this is easy to show by noting that the scalar function X(t)v is the filtered output of the scalar U(t)v. Since U(t)v is a *p.e.* and *u.c.* scalar, lemma 6.2 shows the *p.e.* of X(t)v , and therefore satisfaction of (6-45).

If U and X are $n \times m$ matrices, as is the case in robot parameter estimation, then we can write

$$\mathbf{U}^T\mathbf{U} = \sum_{i=1}^{i=m} \mathbf{u}_i^T \mathbf{u}_i$$

in terms of the row vectors $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n$, and correspondingly

$$\mathbf{X}^T\mathbf{X} = \sum_{i=1}^{i=m} \mathbf{x}_i^T \mathbf{x}_i$$

Therefore, $\forall r \geq 0$ and for any constant unit column vector v

$$\int_r^{r+\delta} \mathbf{v}^T\mathbf{U}^T(r)\,\mathbf{U}(r)\mathbf{v}\,dt = \sum_{i=1}^{i=m} \int_r^{r+\delta} (\mathbf{v}^T\mathbf{u}_i(t))^2\,dt$$

Because of (6-45), among the n terms on the right side, there is at least one term, say the second term, larger than β_1/n , *i.e.*,

$$\int_r^{r+\delta} (\mathbf{v}^T\mathbf{u}_2(t))^2\,dt \geq \beta_1/n$$

Since the scalar function $\mathbf{v}^T\mathbf{x}_2$ is the filtered value of the scalar $\mathbf{v}^T\mathbf{u}_2$, the result in the Appendix indicates that

$$\int_r^{r+\delta} (\mathbf{v}^T\mathbf{x}_2(t))^2\,dt \geq \beta_3$$

where β_3 is a positive constant independent of r. Therefore,

$$\int_r^{r+\delta} \mathbf{v}^T\mathbf{X}^T(t)\,\mathbf{X}(t)\mathbf{v}\,dt \geq \int_r^{r+\delta} (\mathbf{v}^T\mathbf{x}_2(t))^2\,dt \geq \beta_3$$

Note that, in general, β_3 represents the smallest positive number in the few

possible situations, and depends both on the input and on the filter's characteristics. □

6.6 Simulation Results

Let us illustrate the performance of the new estimators using some simulations on the 2-link robot in chapter 3. To generate the data for estimation, we run the tracking the desired sinusoidal-type trajectories in Figure 3.5 using the PD controller there. This means that the input and motion data for estiamtion can be found from Figure 3.7 and Figure 3.5. In avoiding the joint acceleartion in prediction, a first order filter with bandwidth 30 (rad/sec) is used for each joint.

The estimation results of the gradient estimator, with gain matrix being $\mathbf{P} = 0.4\mathbf{I}$, are given in Figure 6.1. The estimated parameters are seeen to be very oscilatory and converge slowly. The results of the standard least square estiamtor with initial gain being $P(0) = 0.4$ are shown in Figure 6.2, with smooth convergence observed. The results of the GBF estimator, with gain bound being 0.4 and maximum forgetting factor being 4 (1/sec), are shown in Figure 6.3. The parameters are seen to be very smooth and the convergence is faster than that of the SLS estimator. Figure 6.4 shows the results of the CF estimator, with the bounding matrix being $\mathbf{K}_0 = 0.4\mathbf{I}$ and forgetting factor being $\lambda = 4$ (1/sec.) The results of the BGF and CF estimators look very similar.

The results of the parameter estimation are affected by the design parameters associated with the estimation gain matrix P. For gradient estimation, this relation between the gain magnitude and estimtor performance is somewhat complicated. Figure 6.5(a) and Figure 6.5(b) show the estimated parameters from a gradient

estimator when the gain matrix is changed to be a smaller gain, $P = 0.03$. One notes that the parameters are actually smoother and converge faster than those in Figure 6.1 corresponding to a much larger gain. This reflects the characteristics that the convergence rate of the gradient estimator improves as the estiamtion gain is increased but, beyond some point, further increase of the gain matrix may lead to slower and more oscilatory parameter convergence. This means that a good gain should be not too large and not too small. But the problem of determining good gain value for egradient estimator is difficult, particularly because the value may also depend on the signals.

The relations between gain maginitude and estimator convergence are simpler for the other estimators: larger gains always lead to faster convergence. for Figures 6.5 (c) and 6.5(d) show the estimated parameters of the standard least square method, which converge slower than those in Figure 6.2 but are still smooth. Figure 6.6 show the estimated parameters of the GAF and CF estimators, and they are seen to be smooth and converge faster than the gradient and SLS estimators.

6.7 Implementation Issues

In order to have good estimation performance, a lot of implementation issues have to be carefully considered, *e.g.*,

1. choice of the bandwidth of the model filter;
2. choice of initial parameter and initial gain matrix;
3. choice of forgetting rate and gain bound;
4. choice of excitation signals;

Tradeoffs and judgement have to be used in making the above choices. The

filter bandwidth should be chosen to be larger than the plant bandwidth so that the system signals are able to pass through. But it should be smaller than frequency range (usually high frequency) of the noise. The initial parameters should be chosen to be as accurate as possible, of course. The initial gain matrix should be chosen as large as possible, preferably to be the maximum allowable upper bound for the gain. The forgetting factor should be chosen to be large enough so that parameter variation can be tracked sufficiently accurately. However, it cannot be chosen too large lest the gain matrix is too large or too oscillatory. The gain bound is chosen based on the knowledge of the noise magnitude and the allowable level of estimated parameter oscillation, with larger noise leading to smaller gain bound. The excitation signals should contain sufficient spectrum lines to allow parameter convergence, and their frequencies should be within the bandwidth of the plant so as to be able to excite the plant. Note that, though unknown parameters can be time-varying, the speed of the variation should be much smaller compared with the plant bandwidth, or the parameter dynamics should be modeled.

6.8 Summary

To avoid the gain-diminishing problem in standard least-square method and the possible gain explosion problem in exponentially forgetting least-square estimation, two simple but effective schemes, called the BGF and CF methods, were studied. The methods, besides guaranteeing the boundedness of the gain matrix, have desirable exponential convergence properties, and good robustness properties with respect to parameter variation and disturbances. They can guarantee bounded parameter errors in the presence of persistent excitation if the parameter variation rate and disturbances are bounded. Such properties imply that the estimators are well-suited to estimate time-varying parameters. The simulation

results indicate that they consistently perform better than the gradient and standard least-square estimators.

6.9 Proof of an Inequality

In this section, we want to show the inequality (6-43), *i.e.*, given the uniform continuity of u(t) and x(t), and (6-42), that there exists a positive constant α_3, such that $\forall r \geq 0$,

$$\int_r^{r+\delta_o} x^2(t)\,dt \geq \alpha_3 \tag{6-43}$$

The proof of (6-43) is done in three steps. The first step shows that, because of the lower boundedness of the integral in (6-42) and the *u.c.* of $u(t)$, there must exist a subinterval, within $[r, r+\delta]$, where $|u(t)|$ is larger than a positive constant (and therefore where $u(t)$ has a constant sign). This, in turn, is used in the second step to show there is another subinterval in which $|x(t)|$ is larger than another positive constant. Integration of |x(t)| in this subinterval then guarantees the satisfaction of (6-43).

Let h(t) be the impulse response of the filter and the positive number T_o the first instant at which h(t) is equal to zero, *i.e.*, $h(T_o)=0$. Then h(t) is nonzero and of constant sign on $[0, T_o]$. It can be easily shown from the exponential stability of the filter that

$$|h(t)| \leq h_o e^{-lt} \leq h_o$$

with h_o and l being positive constants.

From (6-42),

$$(1/\delta_o)\int_r^{r+\delta_o} u^2(t)\,dt \geq \alpha/\delta_o \tag{6-46}$$

Step 1: There must be at least one point t_1 in the interval $[r, r+\delta_o]$ at which the value of u^2 is larger than the above average, *i.e.*,

$$u^2(t_1) \geq \alpha/\delta_o$$

i.e., $|u(t_1)| \geq \sqrt{\alpha/\delta_o}$. Using the uniform continuity property of u(t), there must exist a positive number δ_1 (made $\delta_1 \leq \delta_o/2$) independent of r, such that $\forall t \in [t_1-\delta_1, t_1+\delta_1]$

$$|u(t)| \geq (1/2)\sqrt{\alpha/\delta_o} \tag{6-47}$$

Note that $u(t)$ in this subinterval has a constant sign, because of its uniform continuity. Since $[r, r+\delta_o]$ includes t_1, it should encompass either $[t_1-\delta_1, t_1]$, or $[t_1, t_1+\delta_1]$. We can assume , without loss of generality, that $[t, t+\delta_o]$ includes $[t_1, t_1+\delta_1]$. Step 1 is accomplished.

Step 2: We now show that there exist another subinterval in which $|x(t)|$ is larger than a strictly positive number. To do so, we first find one point in $[t_1, t_1+\delta_1]$ at which $|x(t)|$ is strictly positive, and then use uniform continuity to translate this point property to an interval property. Let $\delta_2 = min(\delta_1, T_o)$. Then, $\forall\, t \in [t_1, t_1+\delta_2]$, both u(t) and $h(t_1+\delta_2-t)$ have constant signs. From the filter input-output relation, we have

$$x(t_1+\delta_2) = x(t_1)\, h(\delta_2) + \int_{t_1}^{t_1+\delta_2} u(t)\, h(t_1+\delta_2-t)\, dt$$

therefore,

$$|x(t_1+\delta_2)| + |x(t_1)|\,|h(\delta_2)| \geq \left|\int_{t_1}^{t_1+\delta_2} h(t_1+\delta_2-t))\, u(t)\, dt\right|$$

$$= \int_{t_1}^{t_1+\delta_2} |h(t_1+\delta_2-t)|\,|u(t)|\, dt \;\geq \zeta \tag{6-48}$$

where $\zeta = (\zeta_1/2)\sqrt{\alpha/\delta_o}$ and

$$\zeta_1 = \int_0^{\delta_2} |h(r)|\, dr > 0$$

From (6-48), either $|x(t_1+\delta_2)| \geq (\zeta/2)$ or $|x(t_1)||h(\delta_2)| \geq (\zeta/2)$.

In the first case, $|x(t_1+\delta_2)| \geq (\zeta/2)$, because of the *u.c.* of $x(t)$, there must exist a a positive constant δ_3 such that $|x(t)| \geq \zeta/4$ on the subinterval $[t_1+\delta_2, t_1+\delta_2+\delta_3]$ (or $[t_1+\delta_2-\delta_3, t_1+\delta_2]$) enclosed by $[r, r+\delta_o]$.

In the second case, namely $|x(t_1)||h(\delta_2)| \geq \zeta/2$, we have $|x(t_1)| \geq \zeta/(2h_o)$. There must exist a similar subinterval $[t_1, t_1+\delta_3]$ in which $|x(t)| \geq \zeta/(4h_o)$.

<u>Step 3:</u> Consider the first case. We have

$$\int_r^{r+\delta_o} |x(t)|\, dt \geq \int_{t_1+\delta_2}^{t_1+\delta_2+\delta_3} |x(t)|\, dt \geq \delta_3 \zeta/4$$

Therefore, using Schwarz inequality, $\int_r^{r+\delta_o} x^2(t)\, dt \geq (\delta_3\zeta/4)^2/\delta_o$. The treatment of the second case is similar.

In summary, there exists a strictly positive constant α_3 (the smallest lower bound among the previous possible situations) such that

$$\int_r^{r+\delta_o} x^2(t)\, dt \geq \alpha_3$$

Note that α_3 is independent of t. □

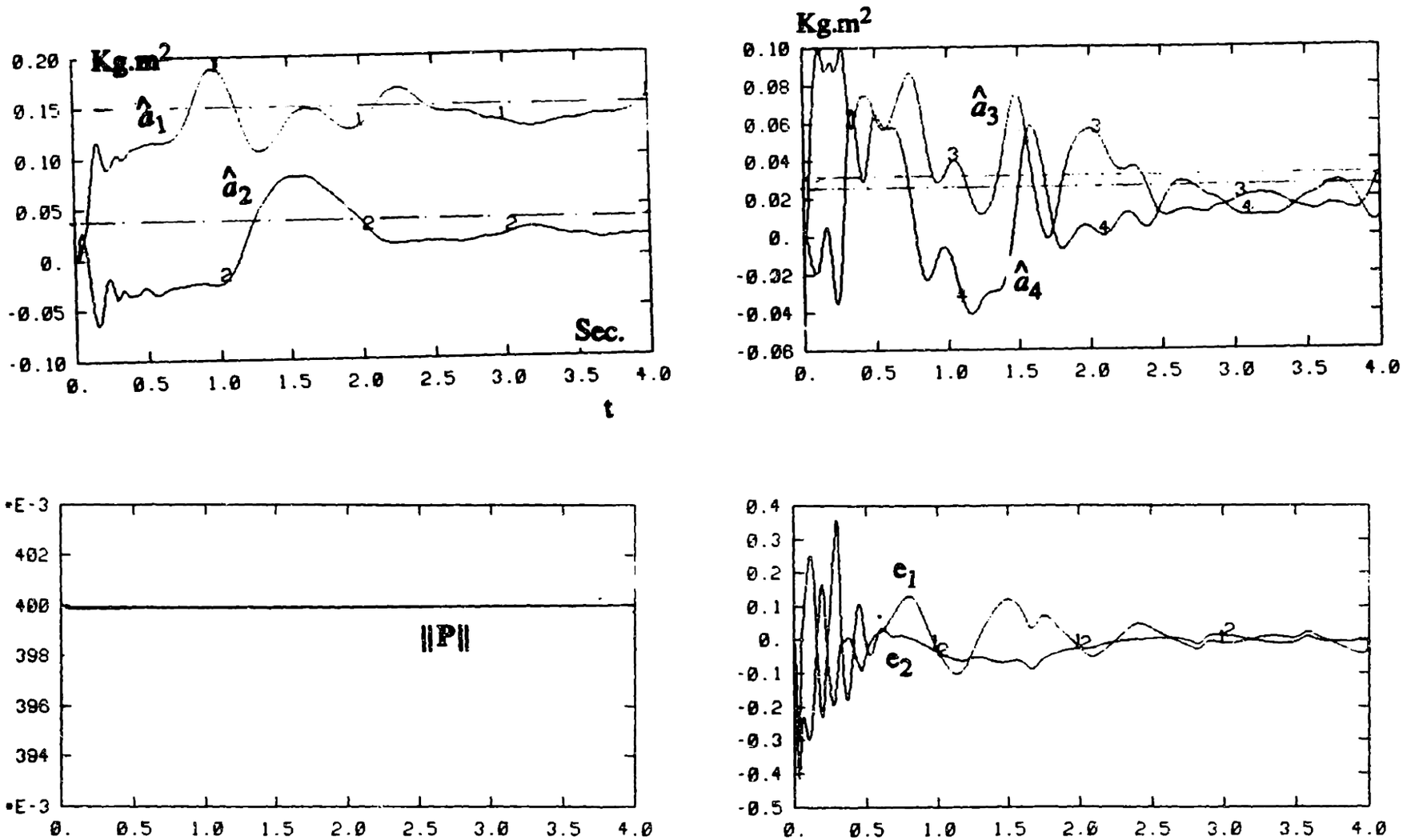


Figure 6-1: Gradient Estimator, P = 0.4 I

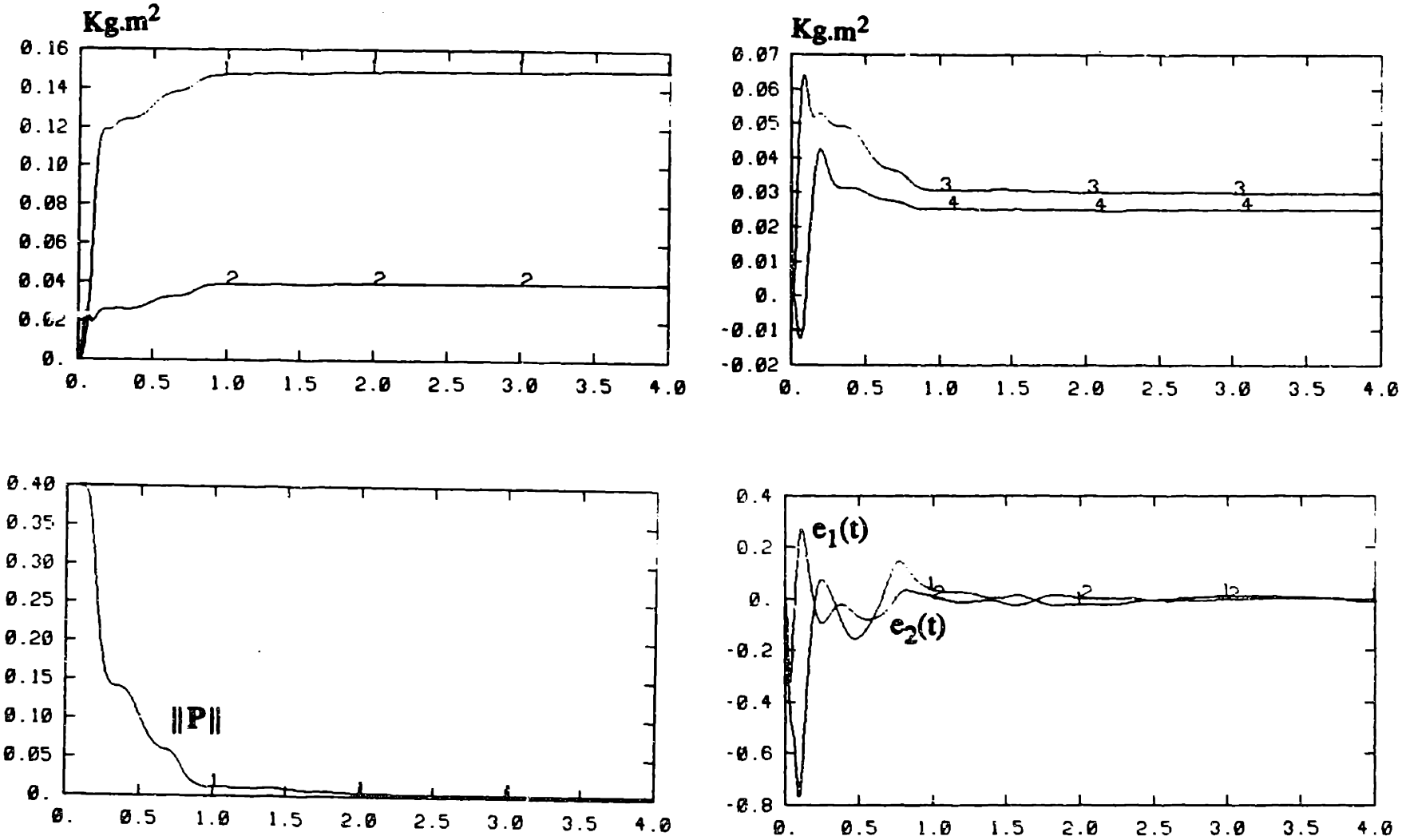


Figure 6-2: SLS Estimator, P(0) = 0.4 I

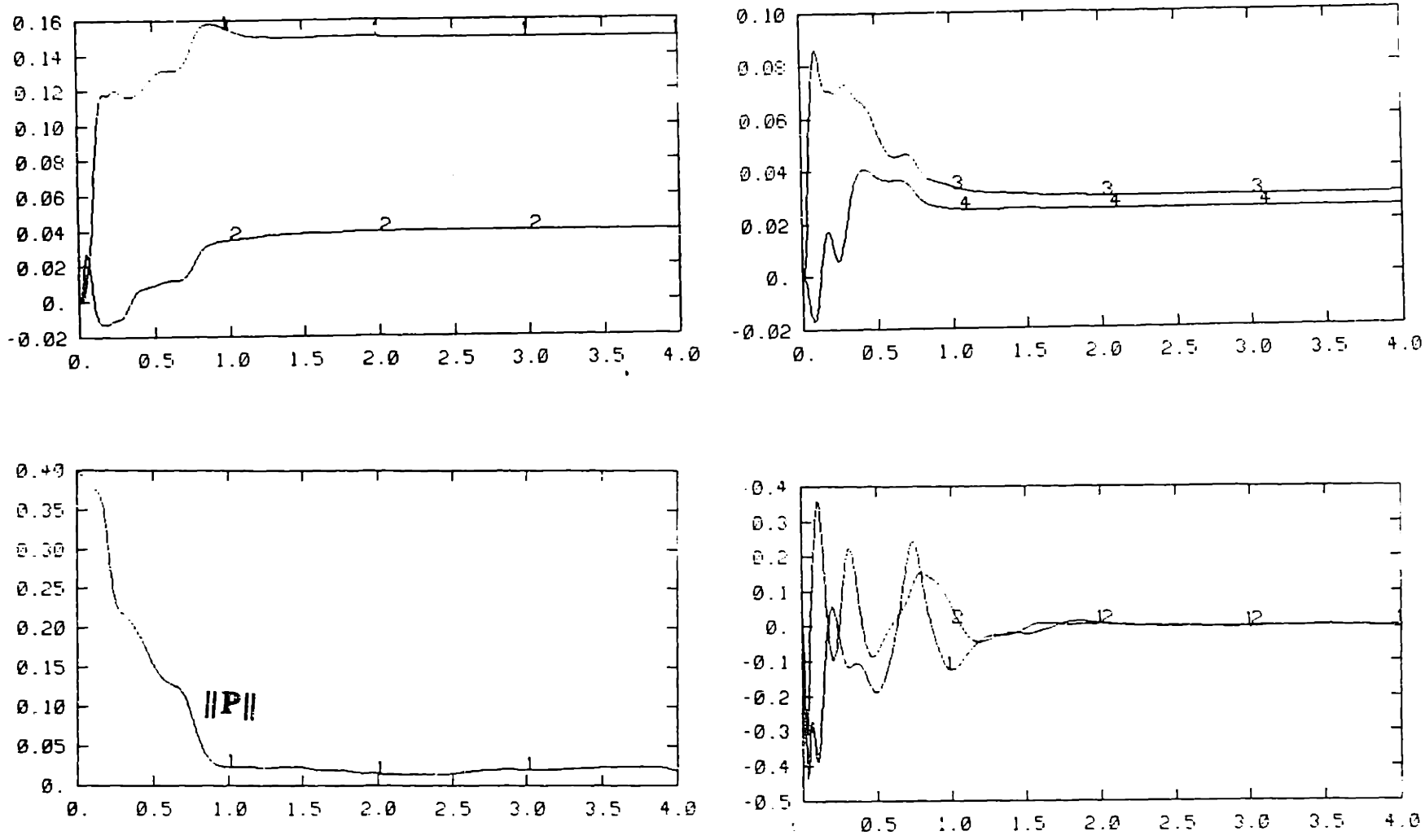


Figure 6-3: BGF Estimator, $k_0 = 0.4$

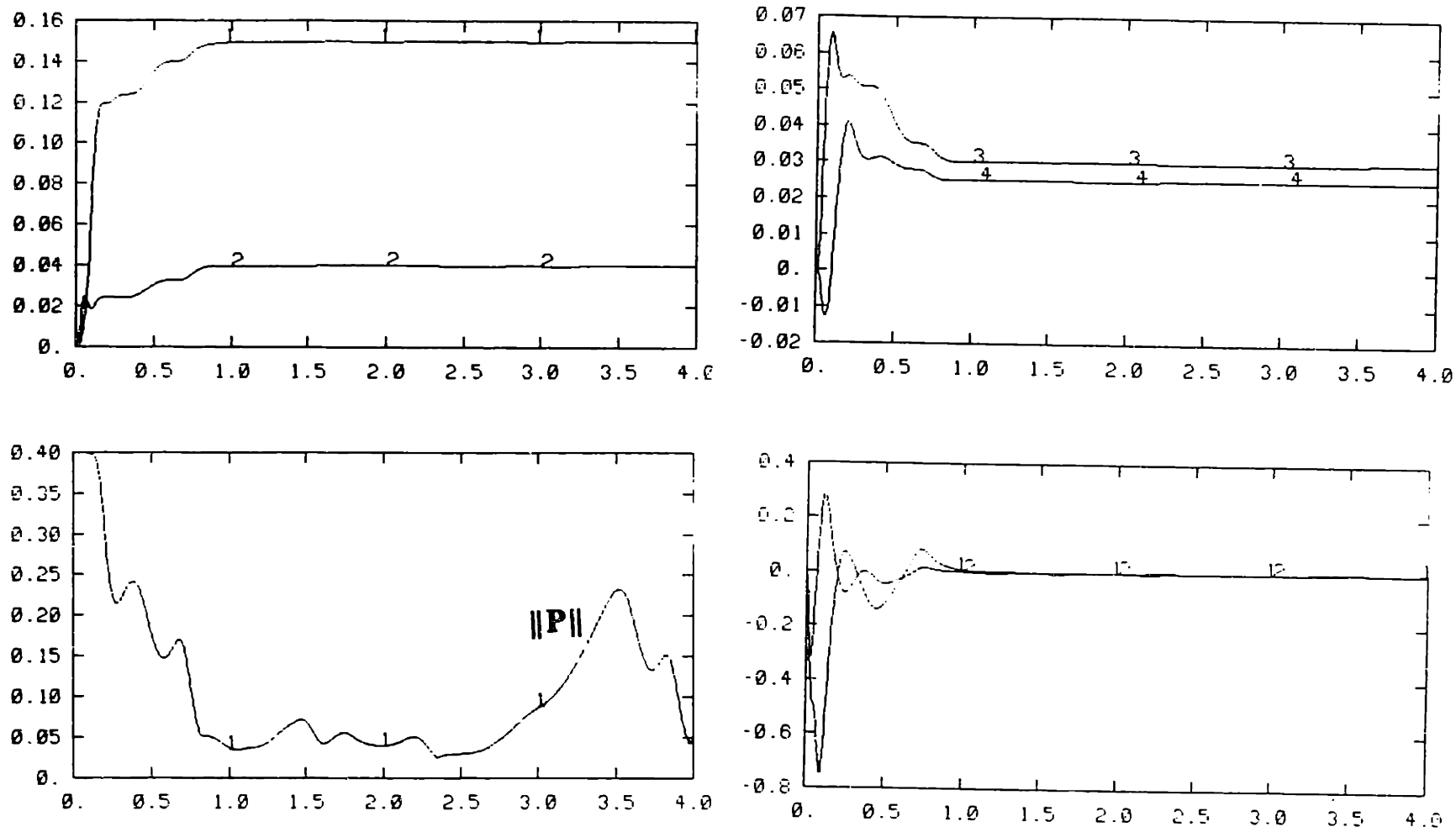


Figure 6-4: CF Estimator, $\mathbf{K}_0 = 0.4\,\mathbf{I}$

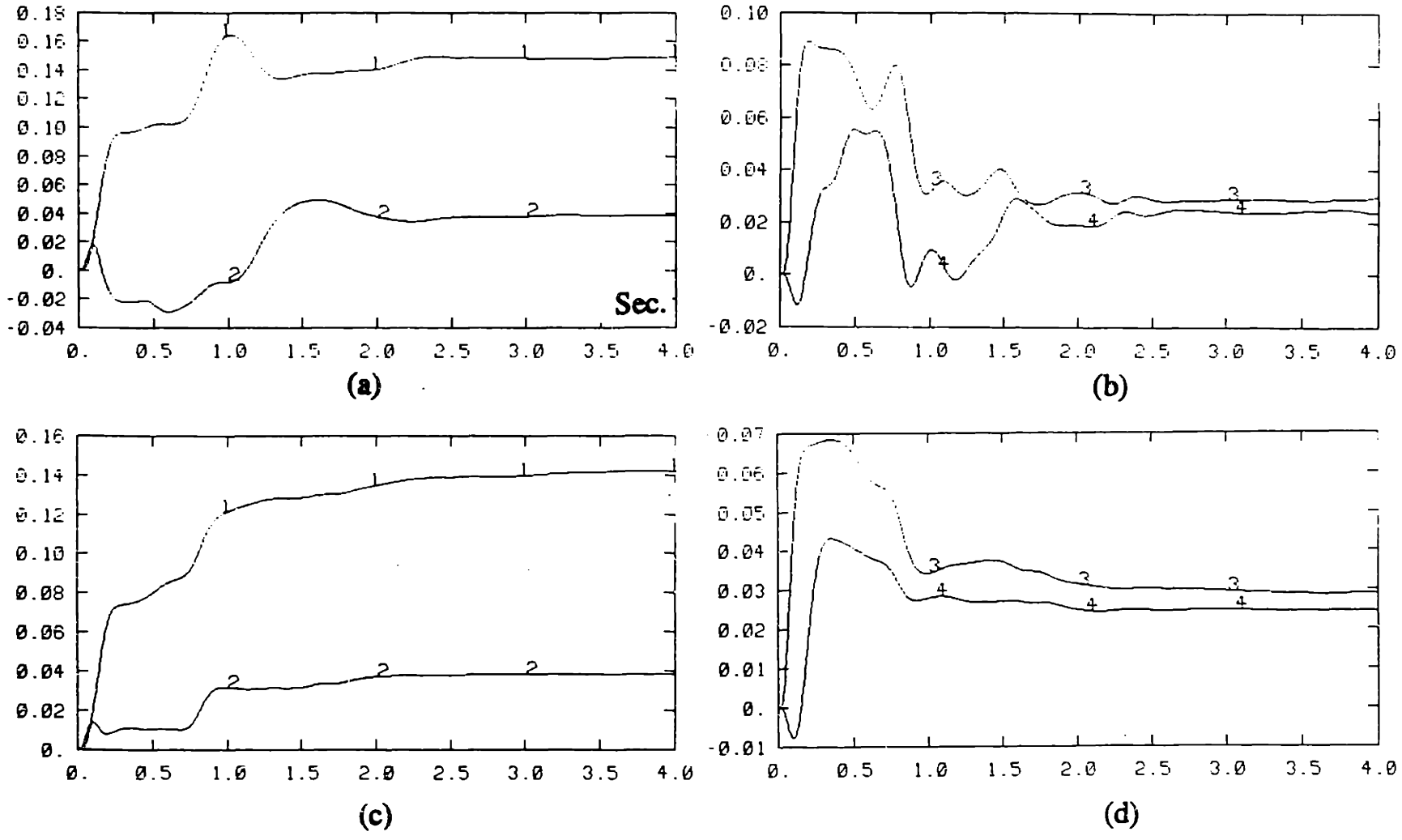


Figure 6-5: smaller gain (a) (b) Gradient Estimator; (c) (d) SLS Estimator

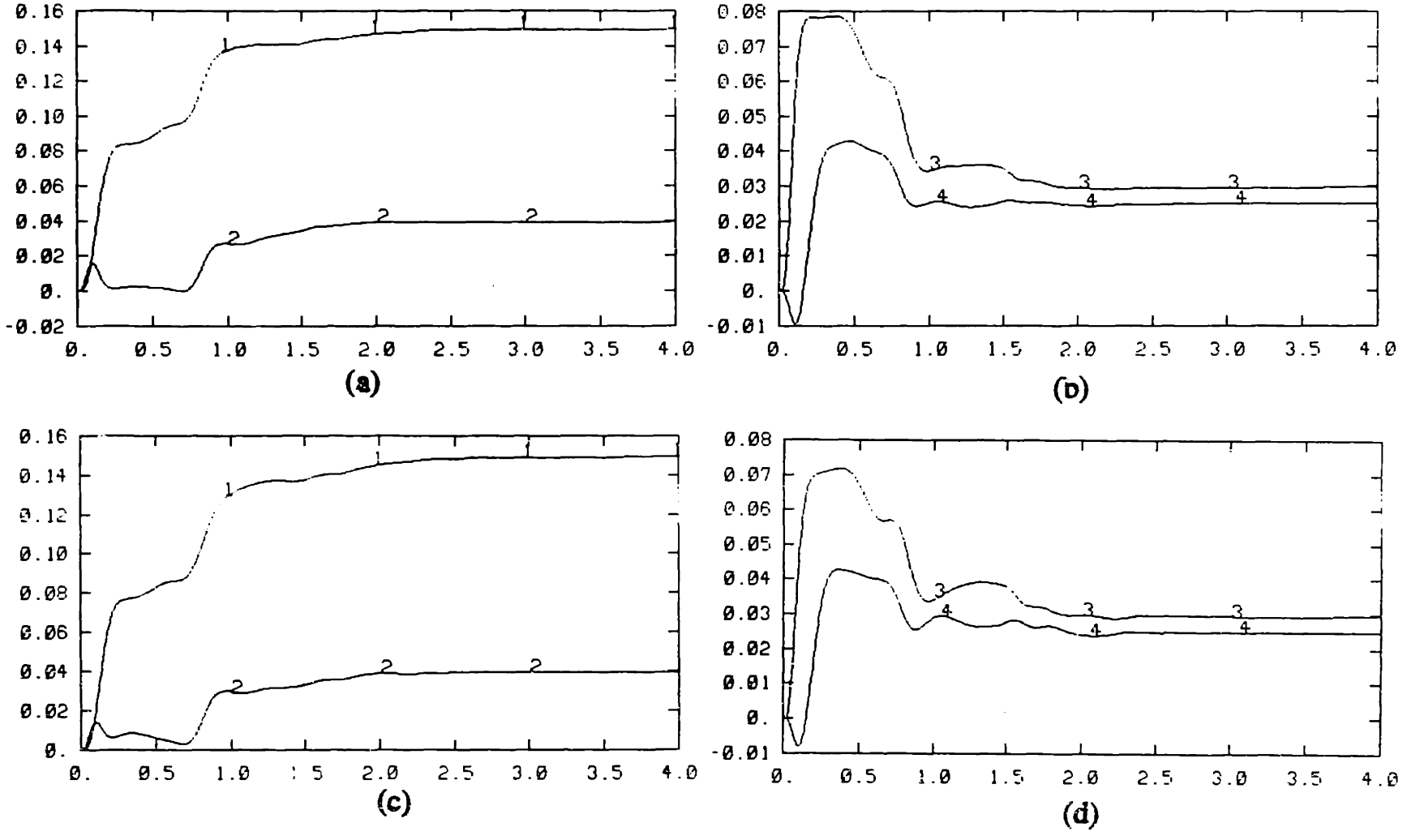


Figure 6-6: smaller gain (a) (b) BGF Estimator; (c) (d) CF Estimator

Chapter 7
Composite Adaptation

For the adaptive controller developed in chapter 3, the adaptation law extracts information about the parameters from the joint tracking errors. However, the tracking errors are not the only source of parameter information. Prediction errors on the joint input also contain parameter information, as reflected in the parameter estimation schemes in chapter 6. The main questions in this chapter are the following: can the different sources of parameter information be combined for parameter adaptation while preserving the stability of the adaptive control system? Does such a combined use of information sources indeed improve the performance of the adaptive controller?

The answers to the previous questions are both positive. In section 7.1 we propose the concept of composite adaptation which means the use of both tracking errors in joint motion and prediction errors in joint torques for parameter adaptation (Figure 7.1). Section 7.2 is devoted to the convergence analysis of the

composite adaptive control systems. Section 7.3 provides simulation results, showing the improved performance of the composite adaptive controllers.

Figure 7-1: Composite Adaptation

7.1 Composite Adaptation

The key to extracting information from both the tracking errors s and prediction errors e is to create a parameter update law which is driven by both s and e. In this chapter, we propose a so-called "composite adaptation law" for this purpose, which is defined to be

$$\dot{\hat{a}}(t) = -P(t)[\, Y^T s + W^T R(t) e \,] \tag{7-1}$$

where $R(t)$ is a uniformly *p.d.* weighting matrix indicating how much attention the adaptation law should pay to the parameter information in the prediction error, and the adaptation gain $P(t)$ is a uniformly *p.d.* gain matrix determined by the techniques to be described below. The quantities Y and s are the same as those in chapter 3, while W and e the same as those in chapter 6. In filtering the dynamics to avoid the joint acceleration (*i.e.*, in generating the matrix W), we use a first-order filter, *i.e.*,

$$W(q, \dot{q}) = \frac{\lambda_f}{p + \lambda_f}[\, Y_1(q, \dot{q}, \ddot{q})] \tag{7-2}$$

with Y_1 defined by (6-4). We thus obtain a new class of adaptive controllers. Note that we propose to use the same form of control law as (3-11), and thus the closed-loop tracking error dynamics have the same form as (3-15). Note that the adaptive controller in chapter 3 is a special case of the above proposed composite adaptive controller, with $R(t)=0$ and P being constant. That adaptive controller will be called tracking-error-based (TEB) adaptive controller. For simplicity, we shall take $R(t)$ to be the unity matrix in the following.

In the composite adaptive controller, we may use the gain update techniques of various parameter estimators in chapter 6 to generate the gain matrix P. Let us summarize some gain update laws and their relevant properties:

The constant gain (CG) method

$$P(t)=P_0 \tag{7-3}$$

where P_0 is a constant symmetric *p.d.* matrix.

The "bounded-gain-forgetting" (BGF) method

The BGF method uses the least-squares gain update

$$\frac{d}{dt}P^{-1}(t)=-\lambda(t)P^{-1}+W^TW \tag{7-4}$$

with the following variable forgetting factor

$$\lambda(t)=\lambda_0(1-\|P\|/k_0) \tag{7-5}$$

where k_0 and λ_0 are two positive constants specifying the upper bound of the gain matrix norm and the maximum forgetting rate. This gain update has been shown to guarantee that $\forall t \geq 0$, $\lambda(t) \geq 0$, and $P(t) \leq k_0 I$ for any signal W, and $\exists \lambda_1 > 0$, $\forall t \geq 0$, $\lambda(t) \geq \lambda_1$ if W is *p.e.*.

The "cushioned-floor" (CF) method

The gain-update is

$$\frac{d}{dt}(\mathbf{P}^{-1})(t) = -\lambda(t)(\mathbf{P}^{-1} - \mathbf{K}_0^{-1}) + \mathbf{W}^T\mathbf{W} \tag{7-6}$$

where $\mathbf{K}_0$ is a constant *p.d.* matrix specifying an upper bound of the gain matrix, and $\lambda(t)$ is a uniformly positive design parameter, *i.e.*, $\lambda(t) \geq \lambda_2 > 0$. If W is *p.e.* as defined by (6-14), then for all $t \geq \delta_0$,

$$\mathbf{P}^{-1}(t) - \mathbf{K}_0^{-1} \geq \alpha_1 e^{-\lambda_2 \delta_0}\mathbf{I} \tag{7-7}$$

Let us also point out some results on persistency of excitation relevant to our later discussion. If tracking errors $\tilde{\mathbf{q}}$ and $\dot{\tilde{\mathbf{q}}}$ converge to zero, then the persistent excitation of $\mathbf{W}_d = \mathbf{W}(\mathbf{q}_d, \dot{\mathbf{q}}_d)$ implies the *p.e.* of W, as shown in chapter 4. Also, as explained in section 6.4, the *p.e.* of $\mathbf{W}_d$ is itself guaranteed by the *p.e.* and uniform continuity of the matrix $\mathbf{Y}_{1d}$, which is defined by

$$\mathbf{Y}_{1d} = \mathbf{Y}_1(\mathbf{q}_d, \dot{\mathbf{q}}_d, \ddot{\mathbf{q}}_d)$$

Interpretation of Composite Adaptation

To give an intuitive interpretation of the composite adaptation law, let us consider, for simplicity, the gain matrix to be $\mathbf{P}(t) = \gamma\mathbf{I}$. The TEB adaptation law is of a kind of gradient nature because it can be written as

$$\dot{\hat{\mathbf{a}}} = -\gamma \frac{\partial \tau}{\partial \hat{\mathbf{a}}}\mathbf{s}$$

This gradient nature of the search direction is the reason for the oscillatory parameter convergence when large adaptation gain is used.

Using (6-8), one can rewrite the composite adaptation law as

$$\dot{\tilde{a}}(t)+\gamma W^T W \tilde{a}=-\gamma\, Y^T s$$

This means that the parameters errors are now a filtered version of the gradient direction, with the filter being time-varying. Thus, the parameter search in the composite adaptation goes along a filtered, or averaged, direction. This can be used to explain why parameter and tracking error convergence in composite adaptive controller is smoother and faster than the TEB adaptive controller, as will be seen in simulations in section 7.3.

7.2 Global Asymptotic and Exponential Convergence

In the following, we show the global asymptotic and exponential convergence of the tracking errors and parameter errors for the adaptive controllers based on the composite adaptation law (7-1), using a Lyapunov-like analysis.

The convergence analysis for the composite adaptive controllers based on various gain-update techniques will all rely on the following scalar function

$$V(t)=\frac{1}{2}[\, s^T H\, s+\tilde{a}^T P^{-1}\tilde{a}\,] \qquad (7\text{–}8)$$

This is the same function as that used for the tracking-error-based (TEB) controller in chapter 3, though we now allow P to be possibly time-varying.

7.2.1 The CG Composite Adaptive Controller

The constant gain (CG) adaptive controller is defined by (7-1), (7-3) and (3-11). The derivative of Lyapunov function can be obtained, using (3-15), (7-1), and the skew-symmetry of $(\dot{H}-2C)$, as

$$\dot{V}(t) = -\mathbf{s}^T \mathbf{K}_D \mathbf{s} - \tilde{\mathbf{a}}^T \mathbf{W}^T \mathbf{W} \tilde{\mathbf{a}} \tag{7-9}$$

This implies that $V(t) \leq V(0)$ and, therefore, that s and $\tilde{\mathbf{a}}$ are upper bounded from the construction of V, since $\mathbf{H}$ is uniformly *p.d.* and $\mathbf{P}_0$ is *p.d.*. It is interesting to note that the scalar function here is the same as that in (3-8) used for the earlier tracking error-based adaptive controller, while its derivative now contains an additional negative term $(-\mathbf{e}^T \mathbf{e})$, indicating that $V(t)$ will decrease as long as either the tracking error or the prediction error is not zero.

Theorem 7.1: *If the desired joint trajectories are bounded, then the tracking errors* $\tilde{\mathbf{q}}$ *and* $\dot{\tilde{\mathbf{q}}}$ *and the prediction errors* e *in the CG composite adaptive controller all globally converge to zero, and the parameter error* $\tilde{\mathbf{a}}$ *remains bounded. If, in addition, the trajectories are persistently exciting and uniformly continuous, the estimated parameters asymptotically converge to the true parameters.*

Proof: The boundedness of s and $\tilde{\mathbf{a}}$ has already been pointed out. The boundedness of $\tilde{\mathbf{q}}$ and $\dot{\tilde{\mathbf{q}}}$ follows from the boundedness of s. We now prove the convergence of the tracking error measure s and prediction error e by showing the convergence of $\dot{V}$ to zero for bounded desired trajectories through the use of Corollary 2.1.

Let us first show the boundedness of $\ddot{V}(t)$ which is

$$\ddot{V} = -2\mathbf{s}^T \mathbf{K}_d \dot{\mathbf{s}} - 2\mathbf{e}^T \dot{\mathbf{e}}$$

since this in turn guarantees the uniform continuity of $\dot{V}(t)$. The boundedness of $\tilde{\mathbf{q}}$, $\dot{\tilde{\mathbf{q}}}$, $\mathbf{q}_d$, $\dot{\mathbf{q}}_d$ and $\ddot{\mathbf{q}}_d$ implies that of $\mathbf{q}$, $\dot{\mathbf{q}}$, $\dot{\mathbf{q}}_r$ and $\ddot{\mathbf{q}}_r$. Examination of the terms in $\mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \dot{\mathbf{q}}_r, \ddot{\mathbf{q}}_r)$ and $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ reveals that they are all bounded, reflecting the physical fact that, for a mechanical manipulator, bounded motion quantities cannot

correspond to unbounded forces. Given the closed-loop dynamics (3-15),

$$\dot{s}=H^{-1}[Y\tilde{a}-(K_D+C)s] \tag{7-10}$$

and the upper boundedness of H^{-1}, $\dot{s}$ is bounded. This also implies that $\ddot{q}$ is bounded.

From (6-8), we have

$$\dot{e}=\dot{W}\tilde{a}+W\dot{\tilde{a}}$$

The second right-hand term is bounded because W, Y, $\tilde{a}$ and s are all bounded. Note that the boundedness of q, $\dot{q}$ and $\ddot{q}$ implies the boundedness of the torque τ and that of the matrix Y_1. Based on the facts that the filter in (7-2) is exponentially stable and strictly proper, and that Y_1 is bounded, one can easily show the boundedness of $\dot{W}$. This in turn guarantees the boundedness of $\dot{e}$.

The boundedness of s, e, $\dot{e}$ and $\dot{s}$ implies the boundedness of $\ddot{V}(t)$. Straightforward application of Corollary 2.1 then leads to $\dot{V}(t)\rightarrow 0$ *as* $t\rightarrow\infty$. Therefore, both the tracking error s and the prediction error e asymptotically converge to zero. The convergence of s to zero in turn guarantees the convergence of $\dot{\tilde{q}}$ and $\tilde{q}$ to zero, according to lemma 3.1.

If W_d is persistently exciting, then W is also *p.e.*. The convergence of the estimated parameters to the true parameters can then be shown easily by noting that the adaptation law

$$\dot{\tilde{a}}=-P_0W^TW\tilde{a}-P_0Y^Ts \tag{7-11}$$

represents an exponentially stable dynamics [Anderson, 1977; Morgan and Narendra, 1977] with convergent input $Y^T s$. □

It is interesting to note that composite adaptive control guarantees the convergence to zero of *both* tracking error and prediction error, while direct adaptive control only guarantees that of the tracking error. This is a reflection of the fact that composite adaptation explicitly pays attention to both tracking error and prediction error.

7.2.2 The BGF Composite Adaptive Controllers

The adaptive controller with P determined by expressions (7-4) and (7-5) is called the bounded-gain forgetting (BGF) composite adaptive controller. For convenience of exponential convergence analysis, we require the simple modification corresponding to (3-33) (3-34), *i.e.*, replacing Y by Y_m.

Theorem 7.2: *The BGF composite adaptive controller has globally convergent tracking errors* $\tilde{\mathbf{q}}$ *and* $\dot{\tilde{\mathbf{q}}}$ *and prediction error* e *if the desired trajectories are bounded. Furthermore, if the desired trajectories are persistently exciting and uniformly continuous, the parameter estimation errors and tracking errors are globally exponentially convergent to zero.*

Proof: For the BGF adaptive controller defined by (3-33), (3-34), (7-1) and (7-4), one has

$$\dot{V}(t) = -\lambda_s s^T \mathbf{H} s - (\lambda(t)/2)\, \tilde{\mathbf{a}}^T \mathbf{P}^{-1} \tilde{\mathbf{a}} - \tilde{\mathbf{a}}^T \mathbf{W}^T \mathbf{W} \tilde{\mathbf{a}} \le 0 \qquad (7\text{-}12)$$

The convergence of $\tilde{\mathbf{q}}$, $\dot{\tilde{\mathbf{q}}}$ and e can be shown as before. Note that, additionally, the convergence of $\dot{V}(t)$ to zero leads to that of $\lambda(t)\tilde{\mathbf{a}}^T \mathbf{P}^{-1}\tilde{\mathbf{a}}$ to zero.

As pointed out in section 6.3, $\mathbf{P}(t) \le k_0 \mathbf{I}$, and the persistent excitation of $\mathbf{W}_d$ (and consequently, that of W) guarantees that $\lambda(t) \ge \lambda_1 > 0$. These imply

$$\lambda(t)\tilde{\mathbf{a}}^T \mathbf{P}^{-1} \tilde{\mathbf{a}} \ge \lambda_1 \tilde{\mathbf{a}}^T \tilde{\mathbf{a}}/k_0 \qquad (7\text{-}13)$$

Therefore the convergence of $\lambda(t)\tilde{\mathbf{a}}^T\mathbf{P}^{-1}\tilde{\mathbf{a}}$ to zero implies that of $\tilde{\mathbf{a}}$. In fact, we can more precisely show the *exponential* convergence of the tracking and estimation errors. Indeed, let γ_0 be the strictly positive constant defined by $\gamma_0=\min(2\lambda_s,\lambda_1)$. In view of (7-11), we can write

$$\dot{V}(t) + \gamma_0 V(t) \leq 0$$

Therefore, $V(t) \leq V(0)\, e^{-\gamma_0 t}$. This, in turn, implies the exponential convergence of s and $\tilde{\mathbf{a}}$ to zero. The exponential convergence of $\tilde{\mathbf{q}}$ and $\dot{\tilde{\mathbf{q}}}$ to zero follows as a result of exponential convergence of s, according to lemma 3.1. □

The exponential convergence of an adaptive controller is an attractive property, since it favors robustness to noise and other disturbances, as indicated by the so-called total stability theory [Slotine and Li, 1990]. Note that only global asymptotic convergence can be shown for the standard least squares gain update.

7.2.3 The CF Composite Adaptive Controller

The cushioned-floor CF composite adaptive controller, obtained by using the CF gain update (7-6), has properties similar to those of the BGF adaptive controller.

Theorem 7.3: *The CF composite adaptive controller has globally asymptotically convergent tracking errors* $\tilde{\mathbf{q}}$ *and* $\dot{\tilde{\mathbf{q}}}$ *and prediction error* e *if the desired trajectories are bounded. If the desired trajectories are persistently exciting and uniformly continuous, the tracking errors and parameter estimation errors exponentially converge to zero.*

Proof: For the CF adaptive controller defined by (3-33), (3-34), (7-1) and (7-6), we can obtain

$$\dot{V}(t) = -\lambda_s s^T H s - (\lambda(t)/2)\,\tilde{a}^T[P^{-1} - K_0^{-1}]\tilde{a} - \tilde{a}^T W^T W \tilde{a} \le 0$$

since the CF gain update equation guarantees that $P^{-1} - K_0^{-1} \ge 0$. This indicates the boundedness of s and $\tilde{a}$. Similar reasoning as before shows the global asymptotic convergence of $\dot{V}(t)$, and accordingly, of the tracking errors $\tilde{q}$, $\dot{\tilde{q}}$, and the prediction error e.

If W_d is persistently exciting, then the parameter estimation errors and tracking errors can also be shown, by using (7-14) and (7-7), to be exponentially convergent to zero with a rate equal to $\min(2\lambda_s, \lambda_2 \alpha_1 e^{-\lambda_2 \delta_0})$. □

7.3 Simulation of the Composite Adaptive Controllers

The main objectives of the simulations here are to verify the results in the above analysis and demonstrate the advantages of the composite adaptive controllers over the tracking-error-based adaptive controller in chapter 3. The robot is the same as that in the simulations in chapter 3 and the first-order filter used to avoid joint acceleration in torque prediction is chosen to have the bandwidth 30 (rad/sec). Three sets of simulation results are presented in this section.

7.3.1 Basic Features of Composite Adaptive Controllers

This set of simulation results is intended to illustrate the basic features of the composite adaptive controllers. We have used the same K_d, Λ matrices as those for Figure 3.6.

For the CG composite controller, the adaptation gain $P_0 = 0.03\,I$ is the same as that for Figure 3.6. Thus, the only difference from the adaptive controller in

chapter 3 lies in the addition of the prediction error term to the adaptation law. The simulation results are shown in Figure 7.1. It is noted that the the tracking errors and the parameter errors all converge better than the TEB controller. Less oscillation is noted in these plots.

For the GAF composite controllers, the gain bound is chosen to be $k_0 = 0.03$ and the maximum forgetting factor is $\lambda_0 = 4$ $(1/sec)$. Therefore, $P(t) \leq 0.03\,I$. The simulation results are shown in Figure 7.2. It is seen that the signals are all very smooth and converge fast. The CF adaptive controller is also simulated with $K_0 = 0.03\,I$ and $\lambda(t) = 4.(1/sec)$. The results, shown in Figure 7.3, are essentially undistinuguishable from those of the GAF controller.

To show the performance of the composite adaptive controllers on non-*p.e.* trajectories, the BGF controller is simulated on the polynomial+rest trajectories in Figure 3.12, with results shown in Figure 7.4 (the gain bound is still 0.03). It is seen that the results are very similar but somewhat better than those in 3-14.

7.3.2 Results With Large Adaptation Gain

The fundamental advantage of the composite adaptation is that it allows large adaptation gains to be used with good results, unlike the TEB adaptation which has poor performance when large adaptation gain is used, as shown in Figure 3.9.

For the CG controller, the adaptation gain is now increased to $P_0 = 0.4\,I$ and the results are plotted in Figure 7.5. One sees that the convergence is now much better than those in Figure 7.1 from the smaller adaptation gain. For the BGF controller, the gain bound is changed to $k_0 = 0.4$, with the maximum forgetting factor unchanged. The results are plotted in Figure 7.6, with smooth convergence clearly seen. The results for CF controller are very similar. One interesting thing

about to note is that the CG composite adaptive controller leads to quite smooth estimated parameters despite the large adaptation gain, unlike the gradient estimator in Figure 6.1. If fact, the variation of the gain matrix does not seem to introduce too much performance improvement in the composite adaptation context.

7.3.3 Performance in the Presence of Unmodeled Dynamics

In the presence of unmodeled dynamics, the composite adaptive controllers perform much better than the TEB controller. To demonstrate this, we included a first-order filter in each joint motor, representing motor dynamics. The first order filters have bandwidths 30 (rad/sec).

With an adaptation gain of $\mathbf{P} = 0.2\mathbf{I}$, the TEB adaptive controller have poor results, as shown in Figure 7.7. Further increase of the adaptation gain will lead to instability of the adaptive controller (the simulation crashes).

With the adaptation gain $\mathbf{P} = 0.2\mathbf{I}$ for the CG adaptive controller, however, the control results are still quite quite good, as shown in Figure 7.8. The tracking errors after the initial adaptation process stay within one degree for both joints. The estimated parameters are also close to the true parameters.

For the BGF adaptive controller, the results are shown in Figure 7.9. It is seen that the results are similar to but somewhat better than those of the CG controller. The CF controller again is found to have very similar results.

7.4 Summary

The composite adaptive controller is obtained by properly combining the tracking error and prediction error for parameter adaptation. It is shown to retain the desirable characteristics of the tracking-error-based adaptive controller in chapter 3, such as global asymptotic convergence of tracking error and avoidance of measuring joint acceleration or of inverting the estimated inertia matrix. Furthermore, the composite adaptive controller has global *exponential* convergence in the presence of persistently exciting trajectories, and its improved performance is reflected by faster parameter convergence and better tracking accuracy in simulations. The incorporation of the prediction error into the adaptation allows the adaptation gain to be increased much higher than the tracking-error-based adaptation, leading to better tracking accuracy and stronger robustness to unmodeled dynamics. The possibility of using large gain is very useful for the adaptive control of time-varying plants. The composite adaptive manipulator control is demonstrated experimentally in [Niemeyer and Slotine, 1989]. Finally, we point out that the idea of composite adaptation can be straightforwardly extended to adaptive control of general linear systems.

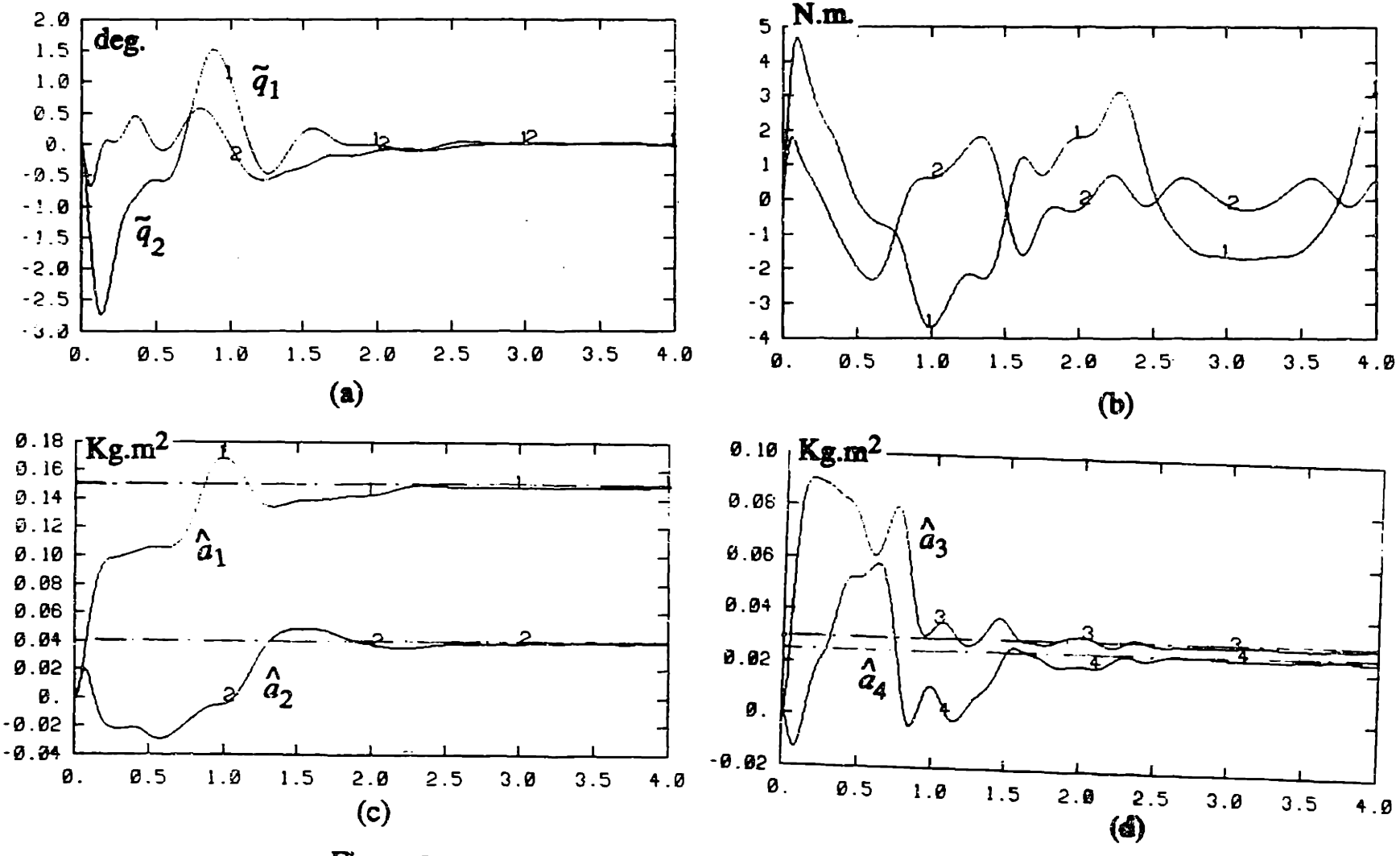


Figure 7-1: CG Composite Adaptive Controller, $\mathbf{P} = 0.03\,\mathbf{I}$

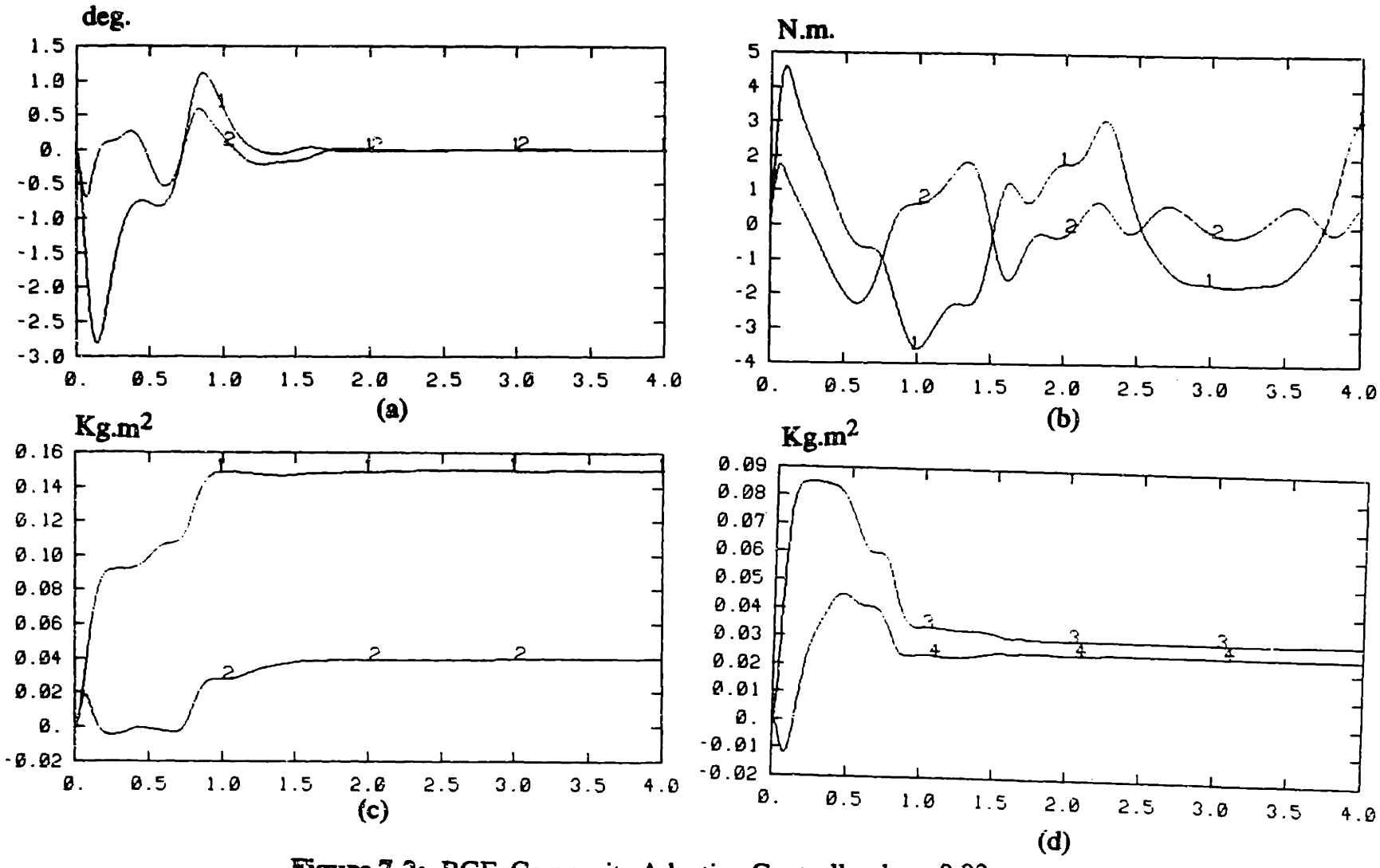


Figure 7-2: BGF Composite Adaptive Controller, $k_0 = 0.03$

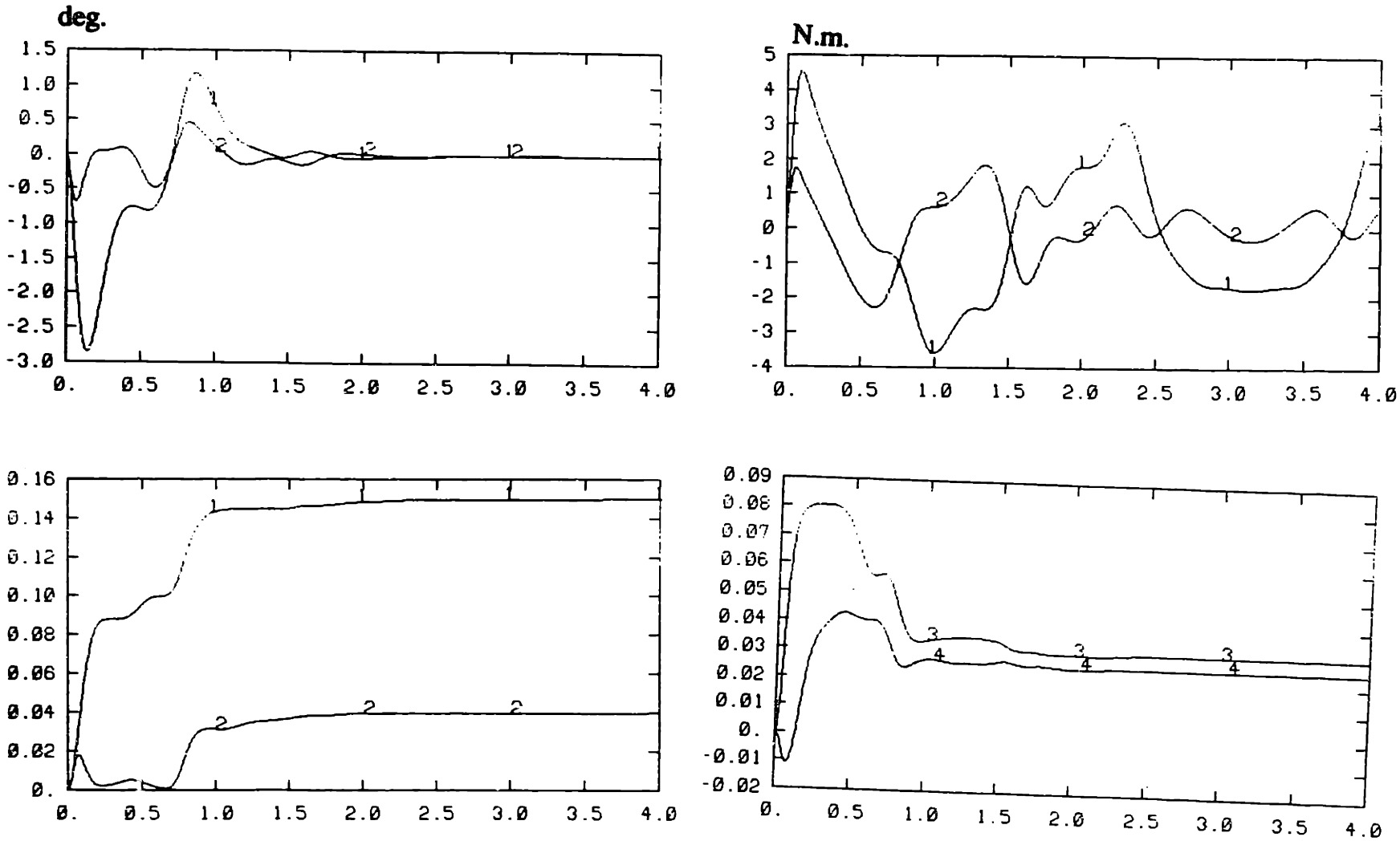


Figure 7-3: CF Composite Adaptive Controlier, $\mathbf{K}_0 = 0.03\ \mathbf{I}$

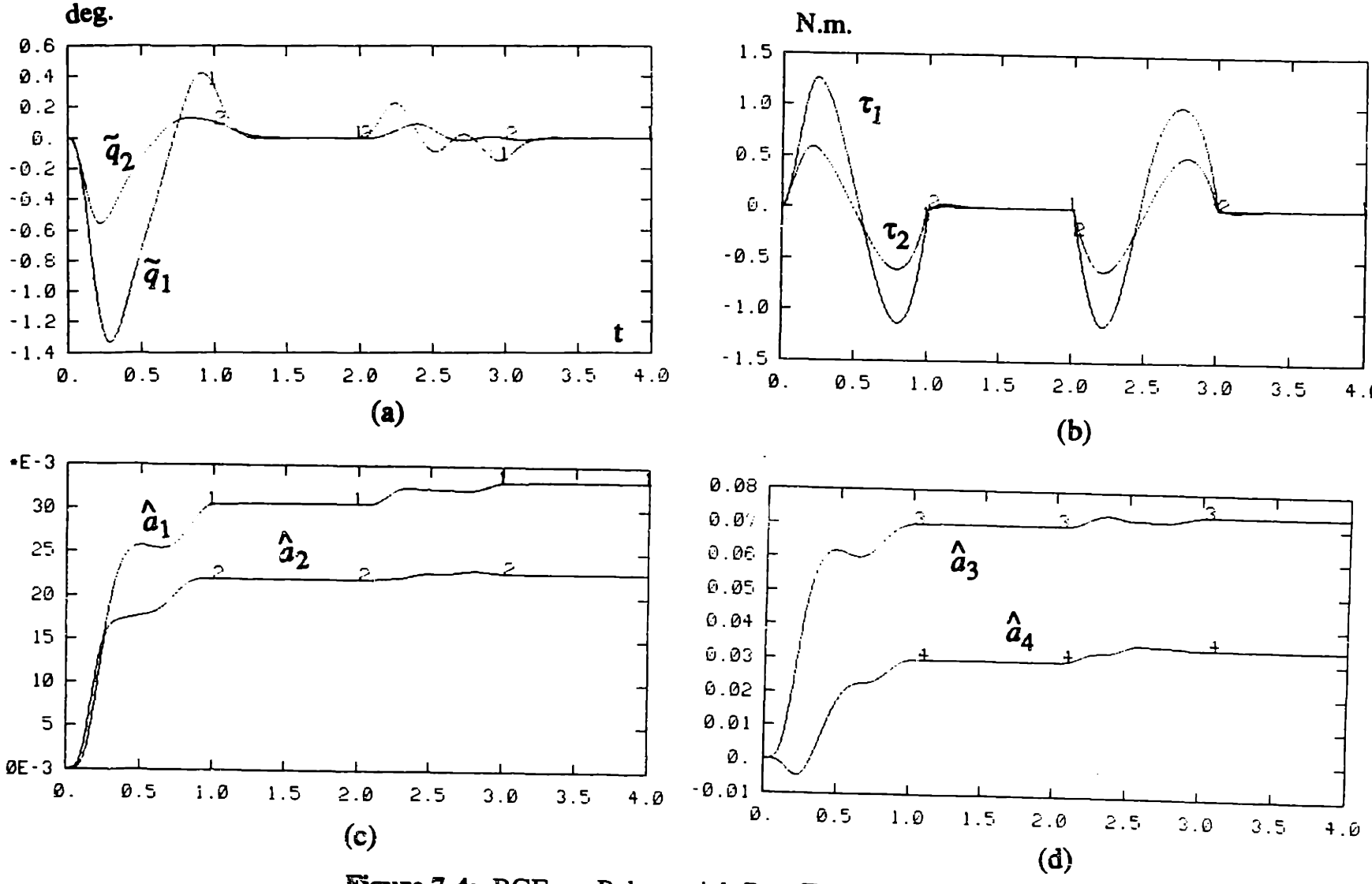


Figure 7-4: BGF on Polynomial+Rest Trajectories, $k_0 = 0.03$

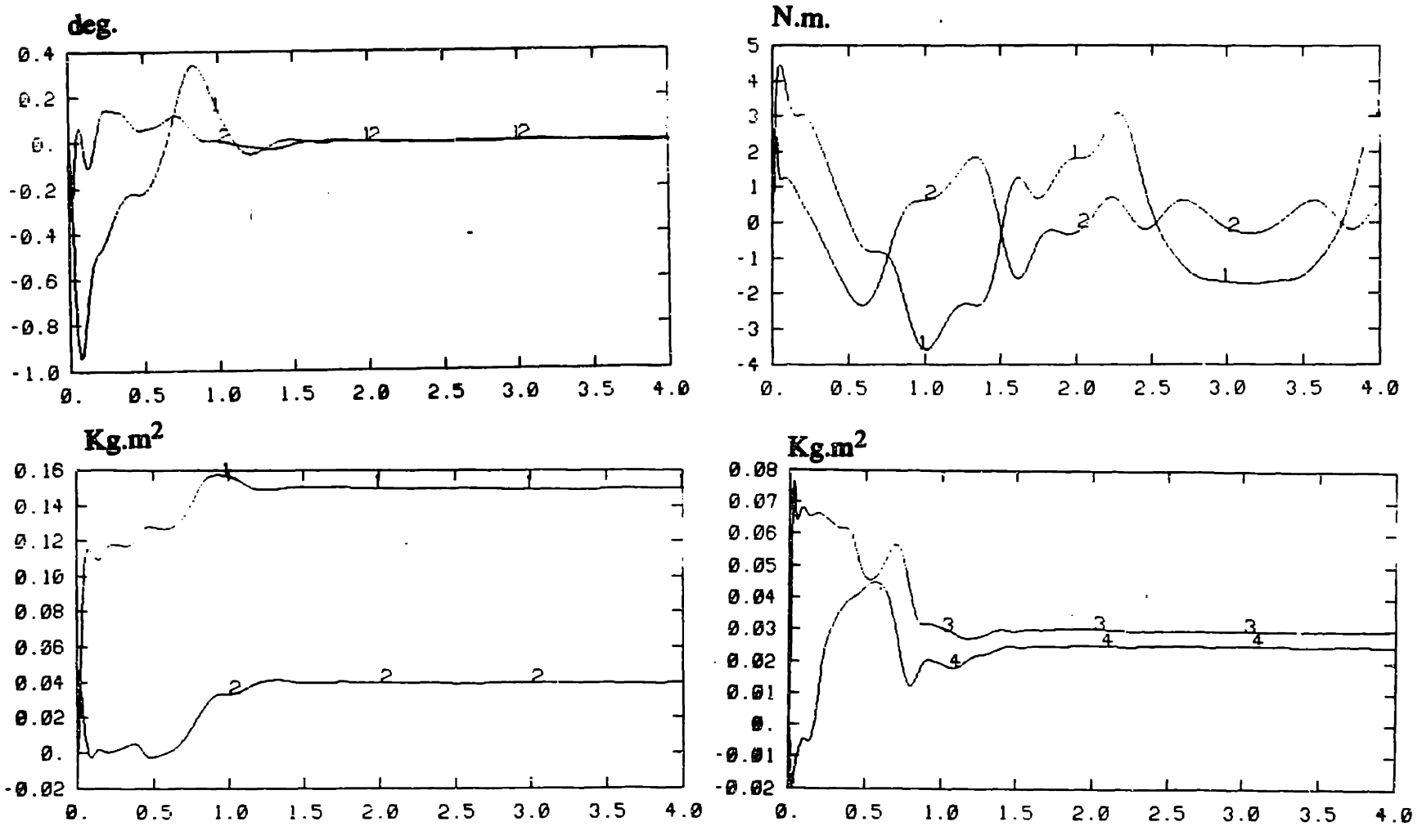


Figure 7-5 CG Composite Adaptive Controller, P = 0.4 I

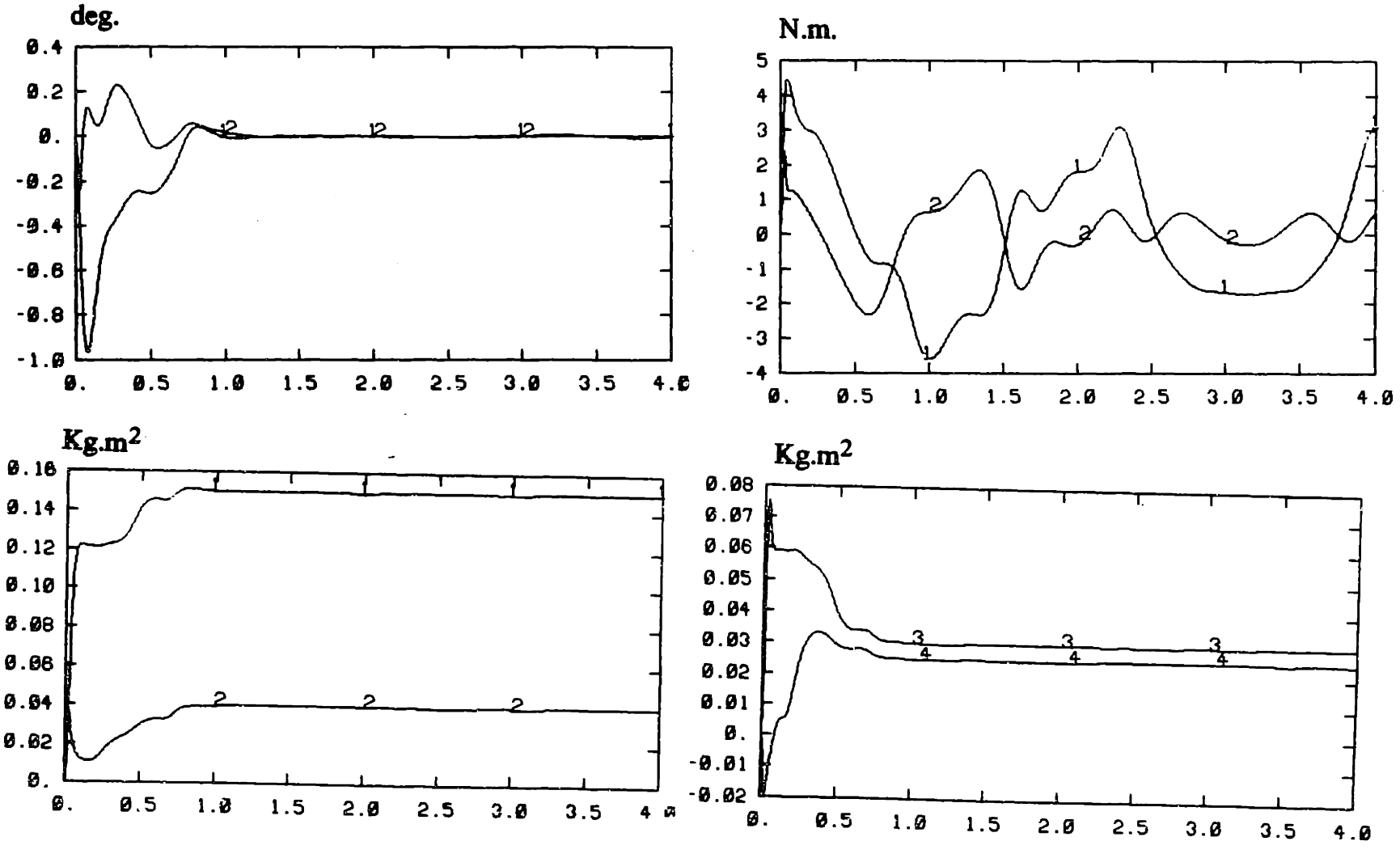


Figure 7-6 BGF Composite Adaptive Controller, $k_0 = 0.4$

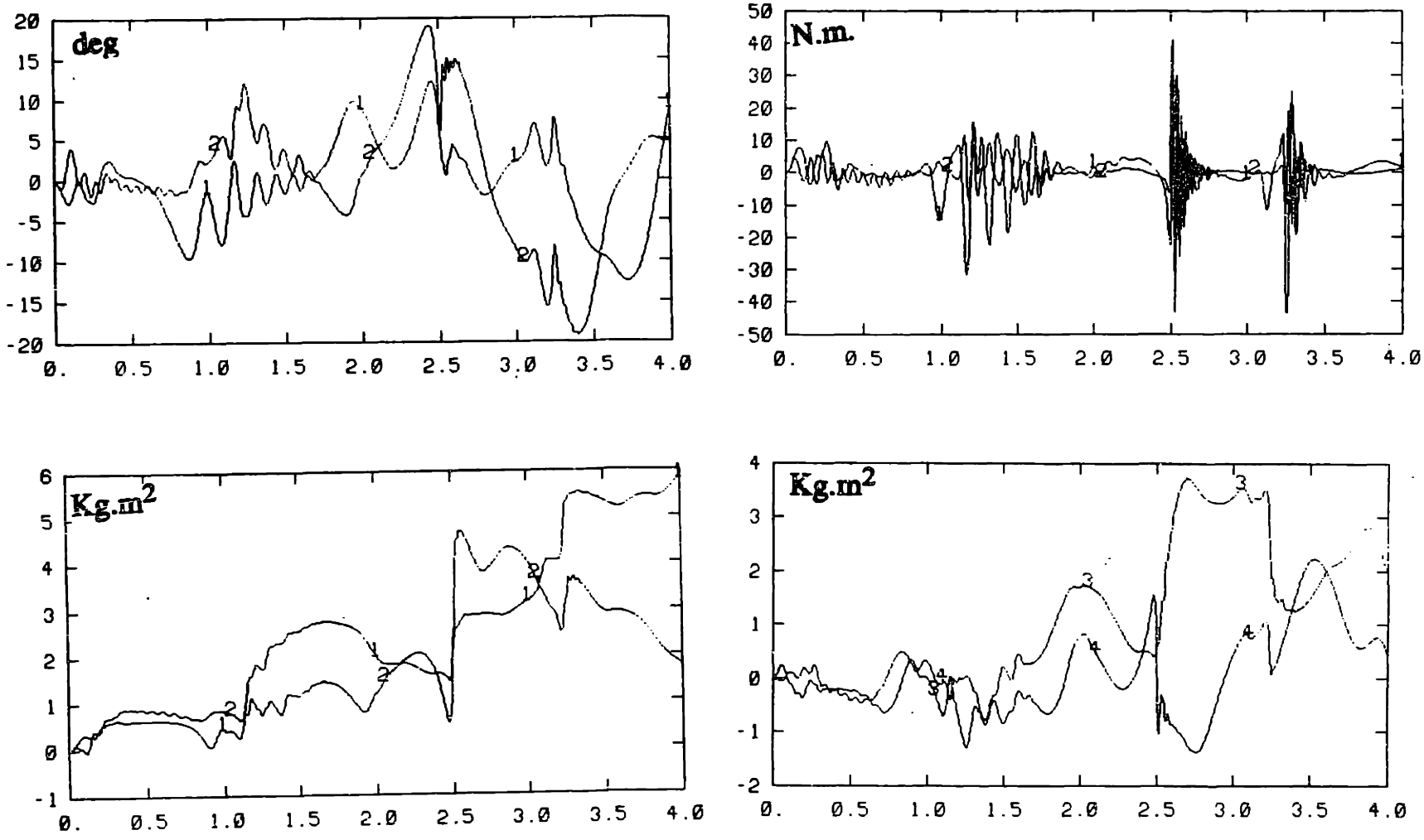


Figure 7-7: TEB Adaptive Controller with motor dynamics

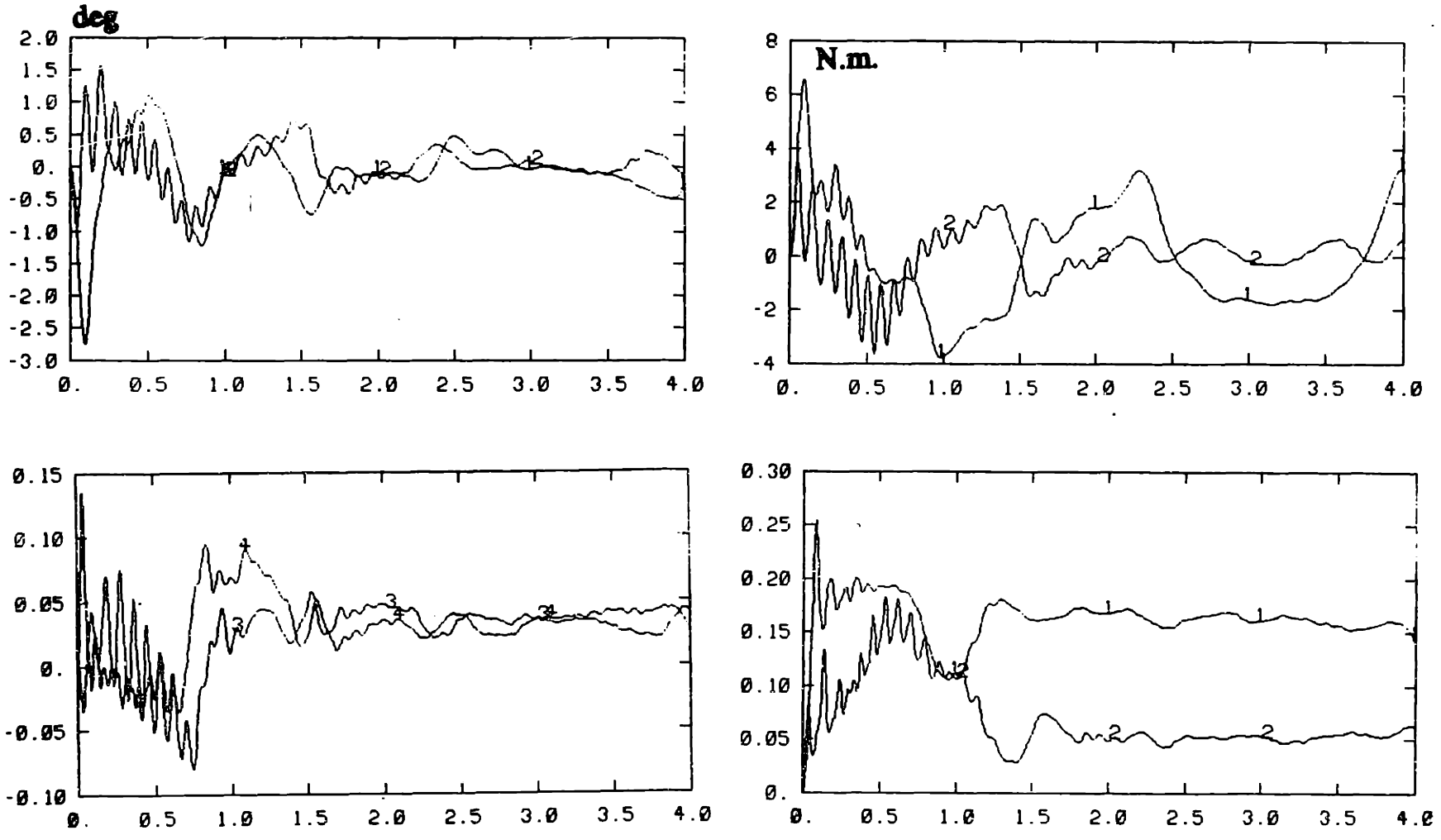


Figure 7-8: CG Adaptive Composite Controller with motor dynamics

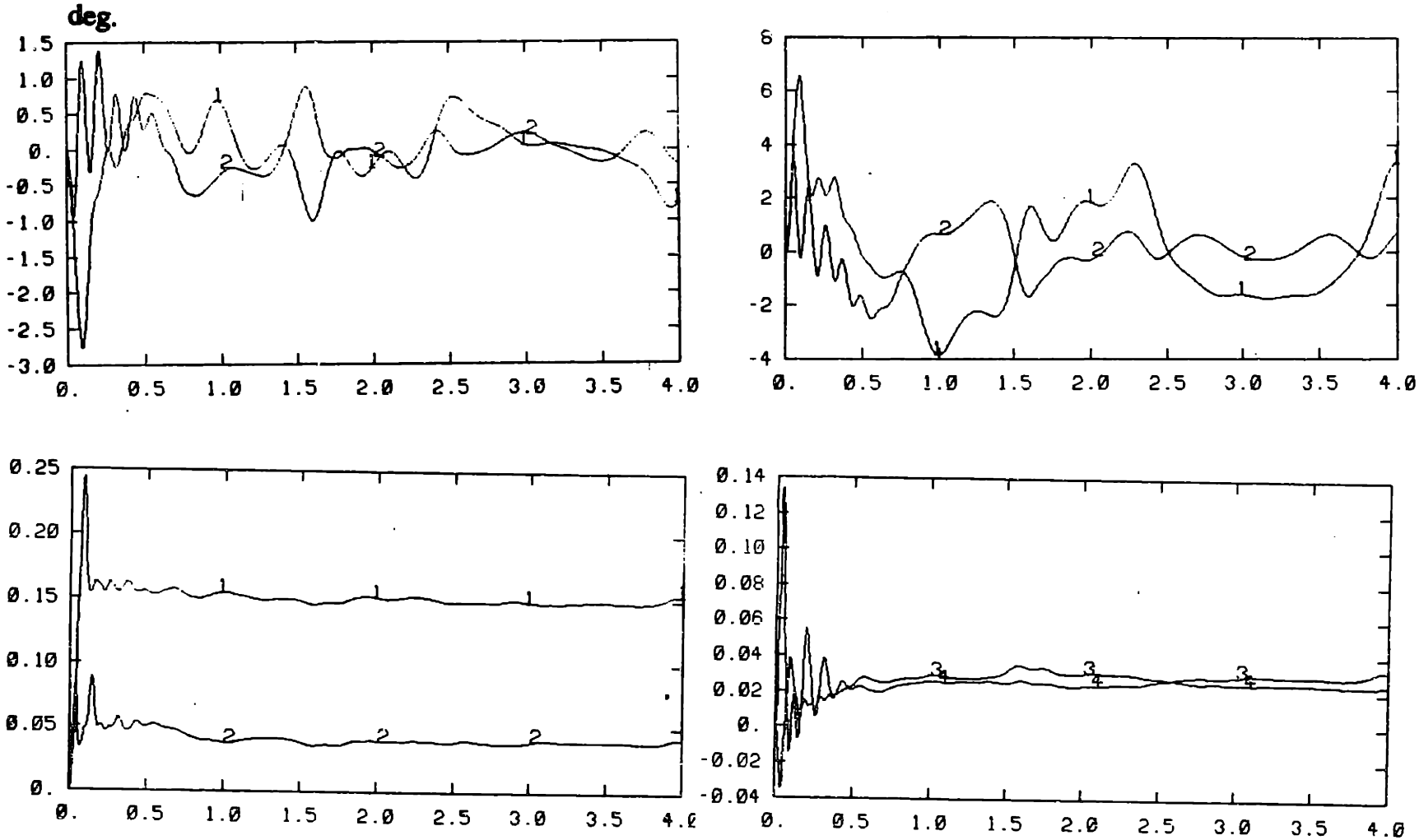


Figure 7-9: BGF Composite Adaptive Controller with motor dynamics

Chapter 8
Adaptive Constrained Motion

In chapter 3, we have developed an adaptive control method for robot free motion. But it is not clear whether and how the method can be used for adaptive control of constrained motion, the other category of robot motion. The basic intent of this chapter is to extend the adaptive free motion controller to the control of constrained motion in the presence of parameter uncertainty. Particularly interesting are constrained tasks with mobile environments, such as turning a crank, pushing an object along a surface, multiple manipulators carrying a common load, multiple robots pushing one object along a surface or turning one crank, and multiple finger manipulation. In such tasks, the dynamics of the environment are not negligible and uncertainty in the inertial parameters can cause uncertainty in dynamic forces.

In this chapter, the extension of the adaptive control method to constrained tasks is achieved by modeling the robot and its constraining environment as an

integrated mechanism and applying the adaptive control method on the dynamics of this mechanism. It is important to point out that the same modeling approach not only allows adaptive controller but also the non-adaptive controllers, like the computed-torque method, to be applied to constrained motion control. Although the focus of this chapter is on adaptive constrained motion control, we will also present the parallel results on non-adaptive constrained motion control, because it is convenient to do so and the non-adaptive results seems to be new and useful. Note that in extending the adaptive and non-adaptive controllers to constrained motion control, we assume that the geometry and dynamics of the constraining environment can be modeled (with geometry, such as the length of a crank, measured by a vision system, for example). This is different from impedance control method [Hogan, 1985] or active stiffness method [Salisbury, 1980] which do not require an explicit model of the environment. However, the added effort in modeling allows us to gain more accuracy in constrained motion.

8.1 Integrated Modeling

This section discusses some general issues in modeling interacting systems composed of robots and their environments. The motivation for the integrated modeling is explained in subsection 7.2.1, and the kinematic and dynamic modeling presented in subsection 7.2.2 and 7.2.3 provides the basis for the later control design.

8.1.1 Robot Motion: Constrained or Unconstrained ?

Whether a motion is constrained or unconstrained is subjective: it depends on the perspective one uses. Consider, for instance, a robot in free motion. If one looks at the individual links, the motion is definitely constrained because the

motion of a link is constrained by other links connected to it. Even if one treats the collection of links as a whole system, its motion is still constrained by the attachment to the base. But when one looks at the generalized coordinates (often chosen to be the joint angles) for the integrated system composed of all the links and the base, the system becomes unconstrained and allows us to label the motion as free motion, because the generalized coordinates can take arbitrary values (within hardware limits).

The same subjective distinction of constrained or unconstrained motion appears in the so called "constrained" robot motion. If one looks at the motion of a robot separately, it is a motion constrained by the interacting environment. If one looks at the generalized coordinates of the integrated system composed of all the interacting mechanical members including robots and environments, the motion is unconstrained. From a motion control point of view, it is advantageous to take this integrated approach, because the interaction forces between the robot and the environment become internal, just like the forces between the links. The same perspective can be used for multi-robot manipulation.

8.1.2 Kinematic Analysis

The kinematic analysis of an integrated mechanism involves the calculation of the number of degrees of freedom, the selection of generalized coordinates, and the determination of the geometrical relationship between the robot joint angles and the generalized coordinates.

The number of degrees of freedom (d.o.f.) of the integrated mechanism can be determined from the numbers of d.o.f. of the robots and mobile environment and the number of constraint equations at the contact point, or using a standard

procedure known as Gruebler's rule. After the number of d.o.f. of the integrated mechanism has been determined, one has the freedom of choosing a set of generalized coordinates among many possibilities. The choice should be based on a number of considerations such as the efficiency of control computation or the relevance to task specification. For the crank-tuning (through a rotatable handle at the crank tip) example of Figure 8.1(a), the integrated mechanism has 1 d.o.f. If the task is to turn the crank angle to follow a desired trajectory, it is convenient to choose the crank angle to be the generalized coordinate. For a two-link manipulator following a hard surface, as shown in Figure 8.2(b), there is 1 d.o.f., and the curvilinear position along of the surface may be used as the generalized coordinate.

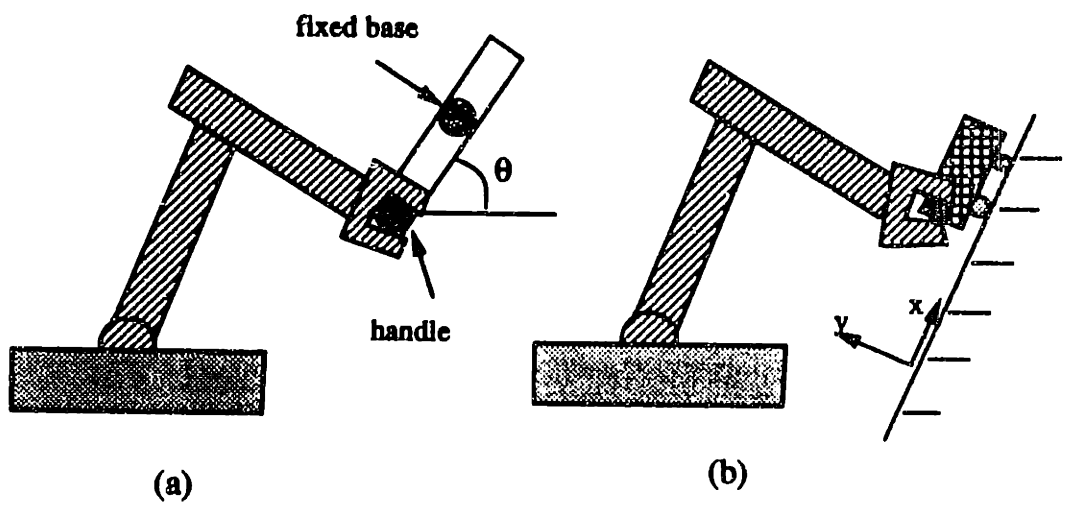


Figure 8-1: (a) manipulating a crank; (b) following a surface

Once the generalized coordinates have been chosen, the joint angles of the robots involved can be expressed as functions of them. Assume that the generalized coordinates are contained in a $n_1 \times 1$ vector $\boldsymbol{\theta}$. The joint positions of all the robots can be put into a $n \times 1$ vector $\mathbf{q}$,

$$\mathbf{q} = \mathbf{Q}(\boldsymbol{\theta}) \tag{8-1}$$

where $\mathbf{Q}$ is a $n\times 1$ nonlinear vector function. This implies that

$$\dot{\mathbf{q}} = \mathbf{Z}\,\dot{\boldsymbol{\theta}} \tag{8-2}$$

where $\mathbf{Z}$ is the $n \times n_1$ Jacobian matrix between the generalized coordinates and the joint coordinates. If the generalized coordinates and their derivatives are not measured directly, they can be determined from inverting (8-1) and (8-2), since the joint position and velocity can usually be conveniently measured by joint encoders and tachometers.

8.1.3 Dynamic Equations

Lagrange's equations can be used to determine the dynamics of the integrated mechanism. The total kinetic energy of an integrated mechanism is the sum of the kinetic energy of the individual robots and mobile environment, and can be written as $\mathbf{E} = (1/2)\,\dot{\boldsymbol{\theta}}^T\mathbf{M}\dot{\boldsymbol{\theta}}$, with $\mathbf{M}$ being the total inertial matrix which is easily determined by the inertia of the robots and the environment. For example, in the crank-turning task, the scalar total inertia is

$$M = I_c + \mathbf{z}^T\mathbf{H}_r\mathbf{z} \tag{8-3}$$

with $\mathbf{H}_r$ being the 2×2 joint space inertia matrix of the 2-d.o.f. robot, I_c the rotational moment of inertia of the crank, and $\mathbf{z}$ the 2×1 Jacobian matrix between the joint angles and the crank angle.

Lagrange's equations easily lead to the following nonlinear dynamics for an integrated mechanism,

$$\mathbf{M}\ddot{\boldsymbol{\theta}} + \mathbf{C}\dot{\boldsymbol{\theta}} + \mathbf{g} = \mathbf{Z}^T\boldsymbol{\tau} \tag{8-4}$$

where the $n_0 \times 1$ vector τ contains the joint torques of all involved robots, the vector $C(\theta,\dot{\theta})\dot{\theta}$ contains the centripetal and Coriolis forces, and g contains gravity (and possibly also friction) forces. The Jacobian matrix appears on the right-hand side of (8-4) because the virtual work by the joint torque τ corresponding to a virtual displacement in θ is $\tau^T \delta q = \tau^T Z \delta\theta$.

Note from (8-4) that the motion is determined by the generalized forces $Z^T\tau$. For convenience, let us define

$$F = Z^T\tau \tag{8-5}$$

and call F the *motion input*. Depending on the specific tasks and robots, the number of degrees of freedom of the integrated mechanism can be equal to, or larger than the number of joint actuators (though it is theoretically possible to have more d.o.f. than the number of joint actuators, such is the case for a two-link robot manipulating a triple pendulum, such a situation is rare in practice). For the first class, which appears most often in compliant tasks, the Jacobian matrix Z is non-square, and the joint input vectors in the null space of Z^T have no effect on the motion, though they affect the internal forces in the mechanism.

The unique properties of the robot dynamics, which are essential for adaptive free-motion control in chapter 3, are retained in the dynamics (8-4) of the integrated mechanism, and will be useful for adaptive compliant motion control design. First, the inertial, gravity, and Coriolis and centripetal forces are all linear in terms of a suitably selected set of robot and environment parameters, *i.e.*,

$$F = Y_1(\theta, \dot{\theta}, \ddot{\theta})\, a \tag{8-6}$$

where Y_1 is a nonlinear matrix function of the motion quantities and geometrical parameters, and the vector a contains the set of (possibly unknown)

constant inertial parameters of the robots and environment. To see this, note that the total inertia matrix is the sum of the transformed inertia of individual robots and the environment, *i.e.*,

$$\mathbf{M} = \sum \mathbf{J}_i^T \mathbf{H}_i \mathbf{J}_i + \mathbf{M}_e$$

where $\mathbf{H}_i$'s are the joint space inertia matrices of individual robots, $\mathbf{J}_i$'s are the Jacobian matrices between the joint positions of the robots and generalized coordinates, and $\mathbf{M}_e$ is the inertia matrix of the environment. Since the joint space inertia matrices $\mathbf{H}_i$'s, and similarly, $\mathbf{M}_e$, can be linearly parametrized [An *et al.* 1985; Khosla and Kanade, 1985], the total inertia, and therefore the motion equation (8-4), must be linearly parametrizable. For instance, this linearity can be seen in the crank-turning example by noting that the total inertia in (8-3) is linear in terms of the crank inertia I_c and of the robot link parameters (because the the robot inertia matrix $\mathbf{H}_r$ is linear in terms of them). Second, as pointed out by [Takegaki and Arimoto, 1981; Koditschek, 1984], the matrices M and C are not independent; specifically, the matrix $\dot{\mathbf{M}} - 2\mathbf{C}$ is skew-symmetric, as is simply shown in chapter 3.

8.2 Motion Control

In constrained motion control, the control design is decomposed into a motion control part and an input-resolution part. The motion control part focus on the accurate control of the constrained motion by $\mathbf{Z}^T\boldsymbol{\tau}$ while the input resolution part utilizes the remaining input freedom (in the null space of $\mathbf{Z}^T$) to maintain good internal forces between the robot end-effector and the mobile environment. This section is devoted to the motion control part.

The motion control design problem can be stated as follows: *given the*

desired motion trajectory $\theta_d(t)$ *and its first two derivatives, and given the joint measurements* $\mathbf{q}$ *and* $\dot{\mathbf{q}}$, *design a control law for joint torque* τ *so that the actual motion* $\theta(t)$ *closely track the desired motion* $\theta_d(t)$.

This control statement is quite similar to that of the free motion control. A few remarks should be made. First, the desired trajectory for the generalized coordinates has been assumed known. In the crank-turning example, this amounts to assuming that the desired history of the crank angle has been given. Secondly, the availability of joint position and velocity implies the knowledge of the generalized coordinates and its derivatives, from inverse kinematics. Third, we implicitly assume the availability of a model of the integrated mechanism, with or without inertial parameter uncertainty. Finally, our approach does *not* assume the availability of a force sensor. If a force sensor is available, its measurement can be used to for the accurate control of internal forces in the presence of model uncertainty, based on a new technique called "uncertainty cancellation" method, as willed be discussed later.

8.2.1 Exponentially Convergent Non-Adaptive Control

Given the knowledge of M, C and g in (8-4), one choice of the motion input F is

$$\mathbf{F} = \mathbf{M}\mathbf{u} + \mathbf{C}\dot{\boldsymbol{\theta}} + \mathbf{g} \tag{8-7}$$

where

$$\mathbf{u} = \ddot{\boldsymbol{\theta}}_d - \mathbf{K}_1\dot{\tilde{\boldsymbol{\theta}}} - \mathbf{K}_2\tilde{\boldsymbol{\theta}}$$

with $\mathbf{K}_1$ and $\mathbf{K}_2$ being constant positive definite matrices, and the tracking error $\tilde{\boldsymbol{\theta}}$ defined by $\tilde{\boldsymbol{\theta}} = \boldsymbol{\theta} - \boldsymbol{\theta}_d$. The choice (8-7) is in the same form as the computed-

torque method for free space motion [see e.g., Asada and Slotine, 1986]. Upon substitution of (8-7) into the dynamics (8-4), the resulting tracking error is found to verify $\ddot{\tilde{\theta}} = u$, *i.e.*,

$$\ddot{\tilde{\theta}} + K_1 \dot{\tilde{\theta}} + K_2 \tilde{\theta} = 0$$

The positive definiteness of K_1 and K_2 can be easily shown to guarantee the exponential stability of the error dynamics and, accordingly, the exponential convergence of $\tilde{\theta}$ and $\dot{\tilde{\theta}}$.

For position control, in which the desired trajectory is a constant point, proportional plus derivative feedback (in terms of any set of generalized coordinates) can be used to guarantee globally stable control, as can be shown using the total equivalent mechanical energy as a Lyapunov function. The steady-state error caused by gravity forces can be eliminated by the addition of either gravity compensation or an integral term.

8.2.2 Asymptotically Convergent Adaptive Control

The controller design in the last section assumes the knowledge of the the kinematic and inertial parameters of the mobile environment. In practice, it is possible that the kinematic parameters (such as the length of the crank) be known reasonably accurately (either *a priori*, or *e.g.* using a vision system), while the inertia parameters of the mobile environment (e.g, the rotational moment of inertia of the crank or the moments of inertia of the large load carried by multiple-robots) are not. Then the computed torque method can no longer be used because of unknown M, C and g, but the adaptive control method developed in chapter 3 can be used to achieve globally convergent tracking in the presence of parameter uncertainty.

Similarly to chapter 3, the "combined error" s is defined by

$$\mathbf{s} = \dot{\tilde{\boldsymbol{\theta}}} + \Lambda \tilde{\boldsymbol{\theta}} \tag{8-8}$$

with Λ being a constant positive definite matrix. The "reference velocity" $\dot{\boldsymbol{\theta}}_r$ is defined by adjusting the desired velocity according to the position tracking error, $\dot{\boldsymbol{\theta}}_r = \dot{\boldsymbol{\theta}}_d - \Lambda \tilde{\boldsymbol{\theta}}$. The reference acceleration $\ddot{\boldsymbol{\theta}}_r$ is the derivative of $\dot{\boldsymbol{\theta}}_r$, *i.e.*, $\ddot{\boldsymbol{\theta}}_r = \ddot{\boldsymbol{\theta}}_d - \Lambda \dot{\tilde{\boldsymbol{\theta}}}$. A nonlinear matrix function Y is defined by the relation

$$\mathbf{M}\ddot{\boldsymbol{\theta}}_r + \mathbf{C}\dot{\boldsymbol{\theta}}_r + \mathbf{g} = \mathbf{Y}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}, \dot{\boldsymbol{\theta}}_r, \ddot{\boldsymbol{\theta}}_r)\, \mathbf{a}$$

from the linear parametrization property.

The control law now is chosen to be

$$\mathbf{F} = \mathbf{Y}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}, \dot{\boldsymbol{\theta}}_r, \ddot{\boldsymbol{\theta}}_r)\, \hat{\mathbf{a}} - \mathbf{K}\mathbf{s} \tag{8-9}$$

where $\hat{\mathbf{a}}$ is the estimated parameter vector, and K is a (possibly time-varying) positive definite gain matrix. The estimated parameters $\hat{\mathbf{a}}$ are obtained from the adaptation law

$$\dot{\hat{\mathbf{a}}} = -\Gamma \mathbf{Y}^T \mathbf{s} \tag{8-10}$$

with Γ being a constant positive definite gain matrix. One can use the following Lyapunov function candidate to establish to global stability and tracking convergence of the above adaptive compliant motion controller

$$V = \mathbf{s}^T \mathbf{M} \mathbf{s} + \tilde{\mathbf{a}}^T \Gamma^{-1} \tilde{\mathbf{a}} \tag{8-11}$$

where $\tilde{\mathbf{a}}$ is the parameter estimation error, *i.e.*, $\tilde{\mathbf{a}} = \hat{\mathbf{a}} - \mathbf{a}$. The estimated parameters are guaranteed to be bounded in general, and converge to the exact parameter values if the matrix Y is persistently exciting, as for the free-motion adaptive control [Slotine and Li, 1987b]. Different ways of improving the basic

control and adaptation laws in (8-9) and (8-10) are discussed in chapters 3 and 7. We also remark that the control law (8-9) is a good alternative to the computed-torque control law (8-7) even when the parameter vector a is known (and therefore adaptation is not necessary).

8.3 Resolution of Input Redundancy

The discussion in section 8.3 indicates that as long as the motion input F is chosen in the form of (8-7) and (8-9), exact or convergent tracking of the desired motion can be achieved. In order to obtain the joint input τ from the relation $\mathbf{F} = \mathbf{Z}^T\tau$, the "same d.o.f." class and the "reduced d.o.f." class have to be considered separately.

In the "same d.o.f." case, the number of d.o.f. of the integrated system is the same as the total number of joint actuators, as is the case for a two link robot manipulating a double pendulum. Therefore, the $n_o \times n$ Jacobian matrix $\mathbf{Z}$ is square. The joint input τ is then solved from F uniquely, *i.e.*, $\tau = \mathbf{Z}^{-T}\mathbf{F}$, assuming non-singularity. In this case, the mobile environment does not really constrain the motion of the robot. Its presence only serves to modify the dynamics. In the "reduced d.o.f." case, the d.o.f. of the integrated system is smaller than the number of joint actuators, *i.e.*, $n < n_o$. The $n_o \times n$ Jacobian matrix is not square, and infinite sets of joint torques can satisfy (8-5), *i.e.*, guaranteeing the same convergent tracking motion but leading to different internal forces. This flexibility, or redundancy, in choosing the input, can be resolved in a number of ways depending on the structure of the environment. We now discuss two of them: minimizing the joint torques and specifying constraint forces.

8.3.1 Minimizing the Joint Torques

One simple way of resolving the joint torque redundancy is to minimize the joint input under the constraint (8-5). This can be used when there are no unidirectional constraint surfaces, as is the case for the crank turning problem and multiple arms rigidly holding a common load. In order to account for the possible difference in the actuator saturation limits among the joints, a weighting matrix can be incorporated into the minimization, leading to the performance index $P_o = \boldsymbol{\tau}^T \mathbf{Q}\, \boldsymbol{\tau}$, where $\mathbf{Q} = \text{diag}\,(1/c_i^2)$, with the c_i's representing the saturation limits of the joints. The solution of the above constrained minimization problem is

$$\boldsymbol{\tau} = \mathbf{Q}^{-1}\,\mathbf{Z}\,[\mathbf{Z}^T\mathbf{Q}^{-1}\mathbf{Z}]^{-1}\;\mathbf{F} \qquad (8\text{-}12)$$

where the matrix $\mathbf{Q}^{-1}\,\mathbf{Z}\,[\mathbf{Z}^T\mathbf{Q}^{-1}\mathbf{Z}]^{-1}$ is the weighted pseudo-inverse of the matrix $\mathbf{Z}^T$. The motion input $\mathbf{F}$ is determined by the various methods in section 8.2.

8.3.2 Specifying Forces in Constraint Directions

If there are unidirectional constraint surfaces in the compliant motion task, as is the case for contour following and multi-finger manipulation, the above method of resolving input redundancy may possibly result in the detachment of the mobile components from the contacting surfaces. In this kind of situation, the input redundancy can be resolved by specifying the reaction force components on the surfaces to be preselected values. Let the reaction force components in the constrained directions be $\mathbf{R}_n$. then $\mathbf{R}_n$ must be a function of the motion quantities and the joint input, *i.e.*, $\mathbf{R}_n = \mathbf{R}_n(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}, \ddot{\boldsymbol{\theta}}, \boldsymbol{\tau})$. Since the acceleration $\ddot{\boldsymbol{\theta}}$ can be expressed as a function of the position, velocity and joint input, and because $\mathbf{R}_n$ must be linearly dependent on the joint input from equilibrium consideration (virtual work principle), the reaction force can be rewritten as

$$\mathbf{R}_n = \mathbf{c}_o(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + \mathbf{C}_1(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})\, \boldsymbol{\tau} \qquad (8\text{-}13)$$

where $\mathbf{c}_0$ is a vector function and $\mathbf{C}_1$ a matrix function. Part or all of the flexibility in choosing the joint torque can be used to set the reaction forces to be equal to (possibly time-varying) nominal value $\boldsymbol{\varepsilon}$

$$\mathbf{R}_n = \boldsymbol{\varepsilon} \qquad (8\text{-}14)$$

The joint torque $\boldsymbol{\tau}$ can then be solved from (8-14) and (8-7) or (8-9).

If there is no model uncertainty, the real reaction force will equal the nominal value. In practice, there will be some unknown terms in (8-13) due to model uncertainty, and the real reaction force will be different from the nominal force. Suggested below are two ways of specifying the nominal forces so as to guarantee contact in the presence of model uncertainty.

The first way is simply to choose a reasonably large nominal force to overcome the unmodeled uncertainty effects, similarly to the idea of bias force in the active stiffness method of [Salisbury, 1980]. The resulting interaction force in the constrained direction will then be larger than zero. An example of this way of specification is given in the contour-following example in section 8.4.3. Humans seem to use this force feedforward method in many compliant motion tasks, such as sweeping floors, *i.e.*, the person simply pushes with a certain bias force normal to floor while sweeping it.

The second way, which requires a force sensor, is an on-line specification method which uses force measurements to infer the uncertain forces in the static equilibrium relation (8-13), and incorporate the detected uncertainty in a periodic modification of the nominal force so as to cancel the unmodeled low-frequency forces. This "uncertainty cancellation" method can be used to achieve accurate

control of contact force around a prespecified desired force, as we now explain using the 1 d.o.f. force control problem in Figure 8.4.

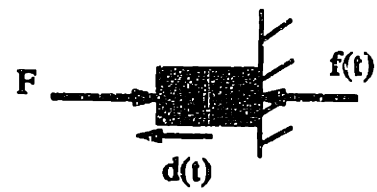


Figure 8-2: force balance

The mass is pushed against a stiff surface by a motor force F and also acted upon by an low frequency unknown disturbance force d. The disturbance force may be due to the unmodeled friction force at the motor joint and/or, in the case of multi-d.o.f. compliant motion, due to the motion in other directions. The objective is to design the motor force F to control the contact force f between the mass and the surface around a desired value f_d in the presence of the uncertain force d. In this case, the nominal model (8-13) is

$$F = f \tag{8-15a}$$

But, with the presence of the disturbance d(t), the forces satisfy the equilibrium condition

$$F - d - f = 0 \tag{8-15b}$$

If the input is designed based on the nominal model (8-15a), *i.e.*, $F = f_d$, the real interaction force will force will be $f = f_d - d$. Contact will be lost if f_d is specified to be a value smaller than d. In order for the contact force f to be f_d. the input F should be $F = f_d + d$. With a force sensor, the unknown disturbance d can be detected and used to compute the above input value. If d is a constant disturbance

force, one can simply try a relatively large input F and measure the corresponding contact force f, which will lead to the detection of the value of d (=F - f). This disturbance force can then be combined with f_d as input to obtain the desired contact force f_d. In case, d is time-varying, as is the case in multi-joint robots, such computation should be updated periodically. In order to avoid exciting and amplifying unmodeled high frequency dynamics, it is useful to average a few samples of (or filter) force measurement to remove noise and vibration before computing the disturbance value. We remark that the basis of this "uncertainty cancellation" method is a *static-equilibrium* view of the interaction in the constraint direction which assumes that the high stiffness of the surface and the arm allows the high frequency vibration to be a negligible factor in the modeling.

Note that, in the hybrid motion/force control method, the force control is achieved using proportional feedback, which is equivalent to taking the nominal force to be $\varepsilon = f_d - K(R_n - f_d)$, where f_d is the desired force and K a gain matrix. However, this method leads to steady-state force control error which in the mass-pushing case is $d/(1+k)$, where k is the force feedback gain. Furthermore, proportional force feedback may excite and amplify the unmodeled high frequency vibration in the constraint directions, [e.g., Eppinger and Seering, 1986], unless soft force sensors are used or filtering techniques are employed [An *et al*, 1988].

8.4 Design Examples

In the previous sections, we have proposed a three-step procedure of designing constrained motion controllers: *i.e.*, modeling the integrated mechanism, designing motion input and then resolving the input redundancy. In this section, the proposed design approach is illustrated on three constrained motion problems. The three problems represent three subclasses of constrained motion: interaction with a

mobile environment, interaction with a fixed environment, and multi-robot cooperation. Using the integrated control design approach, controllers are designed for all of them in the same straightforward fashion.

8.4.1 Manipulating A Mobile Environment

When a robot manipulates a mobile environment, the environment not only introduces geometrical constraints but also dynamic forces. A representative task is the crank-turning problem in Figure 8.1(a) where the crank is turned by a two-d.o.f. robot and the specification of the task can be stated as manipulating the crank such that it rotate with its rotational angle θ accurately following a given motion $\theta_d(t)$. With the crank angle θ chosen as the generalized coordinate, the dynamics is found to be

$$M\ddot{\theta} + (1/2)\dot{M}\dot{\theta} + g = Z^T\tau \tag{8-16}$$

with M defined in (8-3), where τ contains the two joint inputs and g contains gravity (and possibly also friction) forces.

In the absence of parameter uncertainty, the computed-torque method given by (8-7) can be used to achieve exponentially convergent tracking, *i.e.*,

$$F = Mu + (1/2)\dot{M}\dot{\theta} + g \tag{8-17}$$

with $u = \ddot{\theta}_d - 2\lambda\dot{\tilde{\theta}} - \lambda^2\tilde{\theta}$. In the presence of parameter uncertainty, say that the moment of inertia of the crank I_c is unknown, the adaptive controller can be used to guarantee global tracking convergence, with control law having the form in (8-9) and the adaptation law (8-10) for estimating the unknown I_c having the following explicit form

$$\dot{\hat{I}}_c = -\gamma u s$$

It is reasonable to use torque minimization to resolve the input redundancy since there are no unidirectional constraints involved in this task. Then the joint torque is determined by (8-12) and (8-17) or (8-9). Note that, if hybrid motion/force controller is used for this problem [Mason, 1981], time-varying constraints have to be explicitly considered and larger joint inputs have to be used than the minimal here.

8.4.2 Controlling Multiple Robots

Multiple robot cooperation (or multi-finger manipulation) is useful for handling large and heavy loads. Recent contributions by [Zheng and Luh, 1987; Tarn, *et al.*, 1988; Kreutz and Lokshin, 1988] on multiple-robots have used a similar integrated modeling for non-adaptive control of load-carrying. The approach presented here, while being motivated and applicable in a more general context, also allows globally stable and convergent adaptive control to be achieved in the presence of parameter uncertainty.

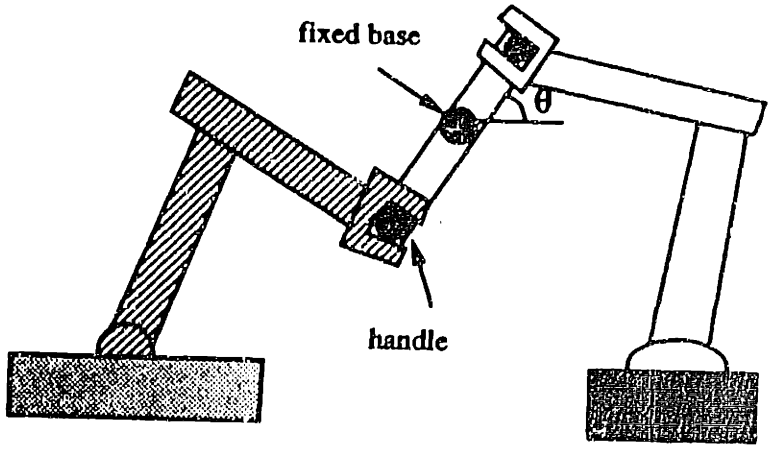


Figure 8-3: Two Robots Manipulating One Crank

For the multi-robot cooperation problem in Figure 8.3, using the integrated control design method, the dynamic equation of the (still 1-d.o.f.) integrated mechanism, the adaptive or non-adaptive design of motion input and torque minimization resolution of input redundancy are all identical to the single robot case in section 8.5.1 except for the difference in total inertia which is now

$$M = I_c + z_1^T H_{r1} z_1 + z_2^T H_{r2} z_2$$

where H_{r1} is the joint space inertia matrix of the first robot, H_{r2} is that of the second, and z_1 and z_2 are the Jacobian vectors between the joint positions and generalized coordinate for the two robots, respectively. We remark that, in multi-robot cooperation involving non-rigid contacts, the input redundancy should be resolved partly or fully using force specification.

8.4.3 Following A Contour

A large number of compliant motion problems, such as object-pushing, grinding and finishing, belong to the class of contour-following, in which the manipulator end-effector (or the object or tool in it) is required to move along a rigid surface while maintaining contact with the unidirectional constraint surface. To be specific, let us examine the application of the integrated control design for the 2-d.o.f. problem in Figure 4. There are a number of interesting issues involved in this problem though the motion input design is quite straightforward. The first is the generation of constraint equation, *i.e.*, how to obtain the static equilibrium equation (its symbolic form given in (8-13) based on reasoning) required for force specification. The second is the behavior of the controlled robot during a bouncing motion, *i.e.*, whether the controller designed for a robot in contact remains stable in case the contact is lost momentarily after a sudden impact. The third is the effect of unmodeled disturbance forces on robot motion and contact

force. These issues are discussed because they appear in all constrained motion tasks involving unidirectional constraints.

A. Dynamic model

Let us take the curvilinear position x along the surface, measured with respect to a certain starting point A, as the generalized coordinate (Figure 8.1(b)) for this 1 d.o.f. integrated mechanism. Though the dynamic equation may be derived from the Lagrangian method, it is preferable to obtain it from the robot dynamics by transformation because it also gives us the needed constraint force equilibrium equation.

It is well known that the dynamics of the robot under external forces r is in the following form:

$$\mathbf{H}_r\ddot{\mathbf{q}} + \mathbf{b}_r = \boldsymbol{\tau} - \mathbf{J}^T\mathbf{r} \tag{8-18}$$

where q is the joint position vector, J is the Jacobian matrix between the end-effector position and the joint position, r in this case includes the tangential and normal forces *on* the surface, and the second term on the left side represent the centripetal, Coriolis, gravity and friction forces. Then, the joint acceleration and the end-effector acceleration is related by

$$\mathbf{w} = \mathbf{J}\ddot{\mathbf{q}} + \dot{\mathbf{J}}\dot{\mathbf{q}}$$

where w is the end-effector acceleration $\mathbf{w} = (\ddot{x}, \ddot{y})^T$.

Solving $\ddot{\mathbf{q}}$ from above, substituting it into (8-18) and multiplying both sides of (8-18) by $\mathbf{J}^{-1}$ leads to

$$\mathbf{M}\,\mathbf{w} + \mathbf{b} = \mathbf{J}^{-T}\boldsymbol{\tau} - \mathbf{r} \tag{8-19}$$

where $M = J^{-T} H_r J^{-1}$ and the vector b is defined obviously. Since the surface and arm are assumed to be stiff, the motion along the direction normal to the surface can be neglected in our modeling, $\ddot{y} = 0$. The first equation in (8-19) is the motion equation along the surface,

$$M_{11}\ddot{x} + b_1 + r_f = z_1^T \tau \tag{8-20}$$

with r_f being the friction force along the surface and the vector z_1 being the first row of the matrix J^{-1}. The second equation is the static equilibrium equation in the direction normal to the surface,

$$M_{21}\ddot{x} + b_2 = z_2^T \tau - r_n \tag{8-21}$$

Using (8-20), this equation can be written in the form of (8-13)

$$r_n = z_2^T \tau - b_2 - M_{21} M_{11}^{-1} (z_1^T \tau - b_1 - r_f)$$

In case the surface is very soft, it should be modeled as a spring and an additional mass. Then the integrated mechanism has two degrees of freedom and a corresponding design should be made following the same three-step procedure.

B. Joint Input Control Design

If the following discussion, the robot model and parameters are assumed to be known, and therefore non-adaptive control design is considered only. If the parameters of the links (or the tool or load grasped to the endeffector) are unknown, one can linearly parametrize the dynamics in terms of these parameters and use adaptive control to achieve globally convergent tracking motion.

Given the dynamic model in (8-20), the motion input can be chosen to be

$$z_1^T \tau = M_{11} u + b_1 + r_f \tag{8-22}$$

according to subsection 8.2.1, where $u = \ddot{x}_d - k_1\dot{\tilde{x}} - k_2\tilde{x}$, with x_d being the desired position along the surface and $\tilde{x}$ the motion tracking error, $\tilde{x} = x - x_d$. With this motion input design, the motion tracking error of the robot along the surface is given by

$$\ddot{x} = u \tag{8-23}$$

The resulting motion tracking error exponentially converges to zero, and, from (8-21) and (8-23), the resulting normal contact force is simply

$$r_n = z_2{}^T\tau - M_{21}u - b_2 \tag{8-24}$$

Now, let us resolve the input redundancy by making the contact force to be equal to a nominal value ε (to be chosen)

$$z_2{}^T\tau = M_{21}u + b_2 + \varepsilon \tag{8-25}$$

The joint inputs can be easily solved from (8-22) and (8-25). The controller is quite similar to that of [Khatib and Burdick, 1986], but no force feedback is required here.

The above controller is designed for the robot in contact with the surface. However, as shown below, it can also suppress bouncing motion at impact without switching between compliant motion mode and free motion mode, provided that, similarly to [Khatib and Burdick, 1986], a velocity damping term ($-k_3\dot{y}$) with proper damping coefficient is added to the left side of (8-25). On the surface, this term will automatically be zero due to lack of motion in that direction. Off the surface, the $\ddot{y}$ is no longer zero, and, from (8-18), (8-22) and (8-25), the dynamic equations in x and y directions are

$$M_{11}\ddot{x} + M_{12}\ddot{y} = M_{11}u \tag{8-26a}$$

$$M_{21}\ddot{x} + M_{22}\ddot{y} = \varepsilon - k_3\dot{y} - M_{21}u \tag{8-26b}$$

Substituting (8-26a) into (8-26b), the motion off the surface in the y direction is obtained as

$$(M_{22} - M_{11}^{-1}M_{12}^{2})\,\ddot{y} + k_3\dot{y} = \varepsilon \tag{8-27}$$

Note that $(M_{22} - M_{11}^{-1}M_{12}^{2})$ is positive because of the positive definiteness of M. Since $(M_{22} - M_{11}^{-1}M_{12}^{2})$ is time-varying, the stability of the dynamics in (8-27) is unclear for general choices of damping coefficient k_3 and nominal force ε. However, if the damping coefficient k_3 and nominal force ε are chosen to be in the following time-varying forms

$$k_3 = (M_{22} - M_{11}^{-1}M_{12}^{2})\,k_0$$

$$\varepsilon = (M_{22} - M_{11}^{-1}M_{12}^{2})\,\varepsilon_0$$

with k_0 and ε_0 being two positive constants, the off-surface dynamic equation becomes

$$\ddot{y} + k_0\dot{y} = \varepsilon_0 \tag{8-28}$$

and the stability of $\dot{y}$ is guaranteed. At the instants of impact, $\dot{y}$ is abruptly reduced due to energy absorption by the rigid surface. By regarding (8-28) as describing a unit mass falling to a rigid floor in viscous air under "gravity" ε_0, one can visualize the bouncing mass settling down on the surface under the combined effects of viscous damping $(-k_0\dot{y})$ and the intermittent energy absorption. In the x direction, the tracking error during off-surface motion is determined by

$$\ddot{x} - u = M_{11}^{-1}M_{12}\ddot{y} = M_{11}^{-1}M_{12}(\varepsilon_0 - k_0\dot{y})$$

Since one can easily show that $\dot{y}$ is bounded from (8-28), the tracking error $\tilde{x}$ is

therefore bounded during the bouncing motion.

C. Model Uncertainty and Guaranteeing Contact

If all the forces in (8-18) are modeled exactly, the above controller will guarantee the tracking of desired motion along the surface to be exponentially convergent, and the contact force normal to the surface to be ϵ. Since in practice there is always be some modeling error, it is natural to wonder whether the existence of unmodeled disturbance forces may lead to loss of contact with the unidirectional constraint surface or loss of stability in the tangential motion. It is shown in the following that contact is guaranteed as long as the nominal contact force is specified to be larger than a certain value which reflects the model uncertainty. It is assumed that the robot links and contact surface are rigid, and that the joint position sensors can accurately locate the contact surface.

Assume that, after initial impact is settled down, there exist unmodeled forces d_x in the x direction, and d_y in the y direction. These may be due to inaccuracy in the friction model, gravity or due to actuator torque ripple. Then, in the presence of modeling uncertainty, the motion equation in the x direction and static equilibrium equation in the y direction are

$$M_{11}\ddot{x} + b_1 + r_f = z_1^T \tau + d_x \tag{8-29}$$

$$M_{21}\ddot{x} + b_2 + r_n = z_1^T \tau + d_y \tag{8-30}$$

With the previously designed controller in (8-22) and (8-25), the motion in the x direction is

$$\ddot{x} - u = M_{11}^{-1} d_x \tag{8-31}$$

The normal contact force is found from (8-30), (8-25) and (8-31)

$$r_n = \varepsilon + d_y - M_{11}^{-1} M_{21} d_x \tag{8-32}$$

It is seen from (8-31) and (8-32) that the tracking error in the x direction does not converge to zero any more due to the unmodeled force d_x, and the normal reaction force is equal to the sum of the nominal force value and terms due unmodeled disturbance forces in both x and y directions. However, if the unmodeled forces are bounded, the tracking errors and the normal force are bounded.

Equation (8-32) indicates that in order to maintain contact, the nominal force must be larger than the uncertainty terms in (8-32). This implies that, if the bounds of model uncertainty in the x and y directions are known,

$$|d_x| < D_x(t) \qquad |d_y| < D_y(t)$$

where D_x and D_y are known positive functions, the contact with the surface can be guaranteed without using force sensor and force feedback by choosing the nominal force to be

$$\varepsilon \geq D_y + M_{11}^{-1} M_{21} D_x$$

In the presence of a force sensor, one may use the uncertainty-cancellation method to achieve accurate force control based on the static equilibrium equation in (8-32).

8.5 Summary

This chapter provides a simple and general approach, integrated control design, for designing non-adaptive and adaptive controllers for various robot compliant motion problems (as well as free motion problems). It involves a three step procedure: integrated modeling of robots and environments, motion control

design and input redundancy resolution. The problems may involve a single robot or multiple robots, non-redundant or redundant manipulators, fixed or mobile environments, rigid contacts or plane contacts. The controllers designed from this approach can guarantee global stability and convergence of the compliant motion given a desired position or trajectory. Some particular contributions of the chapter are the extension of the free-motion adaptive control method to compliant motion, so that globally convergent tracking can be achieved in the presence of large parameter uncertainty, and the proposition of a new force control method, which uses force sensors to detect uncertain forces in constraint directions and allows the actuator inputs to cancel them.

Chapter 9
Conclusion

In this thesis, adaptive control design for robotic dynamics, an important class of multi-input-multi-output nonlinear dynamics, has been developed. This, unlike most previous studies on the topic, has been achieved based on the full nonlinear (rigid body) model of the robot system, without approximations or assumptions such as local linearization, time-invariance, or joint decoupling. Compared with the more recent adaptive control methods by [Craig and *et al*, 1986; Middleton and Goodwin, 1986], the proposed adaptive control design requires neither inversion of the estimated inertia matrix nor the measurement of joint accelerations. The controller design is made by using a Lyapunov-like analysis and exploiting the unique physical properties of the robot dynamics. The resulting adaptive control system is shown to have global stability and the global tracking error convergence under the mild condition that the desired trajectories are bounded. It is also shown that the parameter estimation error will be bounded in general, and convergent to zero if the desired trajectories are sufficiently rich or persistently exciting.

As demonstrated by the experimental results on a a 2-DOF high-speed robot, the adaptive controller has better accuracy than the PD and computed torque controllers. The advantages of on-line adaptation are particularly clear when the robot speeds and/or the parameter uncertainties are high. Even though the desired trajectories used in the experiments, being of polynomial type, have very little persistent excitation, the adaptive controller is found to work very well, indicating good robustness with respect to unmodeled dynamics. Though some amount of parameter drift is observed in the experiments, it is not a serious problem for short time operations of a few seconds. When the adaptive controller is run for long time on a non-*p.e.* trajectory, the use of a dead zone was seen to avoid parameter drift well.

The adaptive control is a practical approach to deal with parameter uncertainty in robotic manipulations also because it requires very reasonable amount of computation. Indeed, because the link parameters can be estimated before adaptive control operation, only ten parameters of the load have to be adjusted on-line by the adaptation law. Furthermore, the control law and adaptation law in the proposed adaptive controller can both be computed recursively using a modified version of the well-known Newton-Eular method, as shown in a recent paper by [Niemeyer and Slotine, 1988].

In some tasks, it is desirable to deal with the uncertainty in the load parameters by using on-line parameter estimation. We have shown that it is possible to modify least-square estimators so that both good parameter convergence and good estimator robustness can be achieved. Two new estimators, called BGF and CF estimators, have resulted from this effort. The data forgetting feature in these estimators allow them to deal with time-varying parameters.

Since both joint motion tracking error and the joint torque prediction error contain information of load parameters, it is desirable to extract parameter information from both sources. We have developed a novel technique called composite adaptation to achieve this purpose. As analysis and simulation suggest, such a combined use of information sources not only retains the global stability of the tracking-error-based adaptive controller, it also leads to faster and smoother convergence. The improved performance is obtained through the possibility of using much larger adaptation gains, without encountering the oscillatory behavior in the tracking-error-based adaptive controller. Note that the improved performance is obtained at the expense of increased computational effort. We also remark that the concept of composite adaptation can be straightforwardly extended to the adaptive control of general linear systems.

Most of this study, as well as the adaptive robot control literature, has focused on the free motion control. However, it is shown that the results can all be applied to adaptive control of constrained robot motion. This is made possible by modeling the robot and its environments as an integrated system, using generalized coordinates for description of its motion. In fact, the adaptive control of constrained motion is essentially the same, except for the possible need of using the input redundancy to guarantee the contact (between a robot end-effector and a uni-directional surface) in the presence model uncertainties.

There are still a number of unresolved issue in adaptive robot control. One such problem is the study of the robustness of the adaptive controller in the presence of disturbances and unmodeled dynamics. Although some simulations have been carried out, no systematic study of the robustness issues in the adaptive robot control has been carried out. Another problem is the generation of persistently exciting trajectories for robot. This is currently limited by the

computation requirements. We believe that use of physical insights or physical properties may be useful to simplify the computational difficulty. Integration of adaptive control with trajectory planning, *i.e.*, using the information gathered during adaptation to modify the planned trajectories, is also a very important topic of research. In adaptive control of constrained motion, many issues, such as experimental implementation, force measurement utilization, geometrical uncertainty and nominal force specification, deserve careful investigations. Finally, we point out that the adaptive control of flexible link robots is still a rarely studied topic, with existing work all relying on local linearization. It is interesting to see whether and how physical properties can be utilized to achieve adaptive flexible robot control based on nonlinear models.

References

An, C.H., C.G. Atkeson and J.M. Hollerbach (1985), Estimation of inertial parameters of rigid body links of manipulators, *Proc. IEEE Conf. Decision and Control*, Fort Lauderdale, FL.

Anderson, B.D.O. (1977), Exponential Stability of Linear Systems Arising From Adaptive Identification. *IEEE Trans. on Auto. Contr.*, AC 22-2.

Anderson, B.D.O., and C.R.Jr. Johnson (1981), Exponential Convergence of Adaptive Identification and Control Algorithms, *Automatica*, vol. 19, No 1.

Arimoto, S. and F. Miyazaki (1983), Stability and Robustness of P.I.D. Feedback Control for Robot Manipulators of Sensory Capability, *Proc. 1st Int. Symp. Robotics Res.*.

Armstrong, B. (1987), On Finding Exciting Trajectories for Identification Experiments Involving Systems With Nonlinear Dynamics, *Proc. IEEE Int. Conf. Robotics and Automation.*

Asada, H., Kanade, T., Takeyama, I. (1983), Control of A Direct-Drive Arm, *ASME Journal of Dynamic Systems, Measurement, and Control* No. 105.

Asada, H., and J.J.E. Slotine (1986), Robot Analysis and Control, *John Wiley and Sons.*

Astrom, K.J. (1980), Self-tuning Regulators-Theory and Applications, in *Applications of Adaptive Control*, Edited by R. Monopoli and K. Narendra, Academic Press, New York.

Astrom, K.J. and Wittenmark, B. (1989), Adaptive Control, *Addison-Wesley.*

Atkeson, C. G., An, C. M. and Hollerbach, J. M. (1985), Estimation of Inertial Parameters of Manipulator Loads and Links, *Proc. Int. Symp. Robotics Res.*, Gouvieux.

Bayard, D.S., and Wen, J.T. (1987), Simple Adaptive Control Laws for

Robotic Manipulators, *Proc. of Fifth Yale Workshop on the Applications of Adaptive Systems Theory.*

Balestrino, A., De Maria, G. and Zinober, A.S. I. (1983), An Adaptive Model Following Controller for Robotic Manipulators, *ASME J. Dyn, Sys. Meas. Contr.*, No. 105.

Bodson, M. (1986), Stability, Convergence and Robustness of Adaptive Systems, Ph.D thesis, University of California at Berkeley.

Canudas, C., Astrom, K. J. and Braun, K. (1986), Adaptive Friction Compensation in D.C. Motor Drives, *Proc. IEEE Int. Conf. Robotics and Automation* San Francisco.

Choi, Y.K., Chung M.J. and Bien, Z. (1986), An Adaptive Control Scheme for Robot Manipulators, *Int. J. Control*, vol. 44, pp1185-1191.

Craig, J.J. (1986), An Introduction To Robotics, *Addison-Wesley.*

Craig, J.J., P. Hsu and S. Sastry (1986), Adaptive Control of Mechanical Manipulators, *Proc. IEEE Int. Conf. Robotics and Automation*, San Francisco.

Desoer, C.A. and Vidyasagar, M. (1975), Feedback Systems: Input-Output Properties, *Academic Press.*

Dubowsky, S. and DesForges, D. T. (1979), The Application of Model-referenced Adaptive Control of Robotic Manipulators, *ASME J. Dynamic Syst., Meas., Contr.*, 101.

Goodwin, G.C. and Sin, K.S. (1984). Adaptive Prediction, Filtering and Control, *Prentice Hall.*

Hogan, N. (1981), Impedance Control of a Robotic Manipulator, *Proc. ASME Annual Winter Meeting*, Washington, D.C.

Hogan, N. (1985a), Impedance Control: An Approach to Manipulation, *ASME J. Dyn. Syst., Meas., and Control*, 107.

Hogan, N. (1985b), Control Strategies for Computer Movements Derived from Physical Systems Theory, *Proc. Int. Symp. Synergetics*, Bavaria.

Hollerbach, J.M. (1980), A Recursive Formulation of Lagrangian Manipulator Dynamics, *Proc. IEEE Trans. Systems, Man, Cybernetics*, vol. 10, No. 11.

Hsia, T.C. (1986), Adaptive Control of Robot Manipulators - A Review, *Proc. IEEE Int. Conf. Robotics and Automation*, San Francisco.

Hsu, P., S. Sastry, M. Bodson and B. Paden (1987), Adaptive Identification and Control of Manipulators Without Joint Acceleration Measurements, *Proc. IEEE Int. Conf. Robotics and Automation*, Raleigh, NC.

Irving, E. (1979), Unpublished.

Kanade, T., Khosla, P. K. and Tanaka, N. (1984), Real-Time Control of CMU Direct-Drive Arm II Using Customized Inverse Dynamics, *I.E.E.E. Conf. On Decision and Control*, Las Vegas.

Khatib, O. (1980), Commande Dynamique dans L'Espace Operationnel des Robots Manipulateurs en Presence D'obstacles, *Docteur Ingenieur Thesis, Ecole Nationale Superieure de L'Aeronautique et de L'Espace*, Toulouse, France.

Khatib, O. (1983), Dynamic Control of Manipulators in Operational Space, *Sixth IFTOMM Congress on Theory of Machines and Mechanisms, New-Delhi.*

Khatib, O. (1985), The Operational Space Formulation and the Analysis, Design, and Control of Robot Manipulators, *3rd Int. Symp. Robotics Res., Gouvieux.*

Khatib, O. and Burdick, J. (1985), Dynamic Optimization in Manipulator Design: The Operation Space Formulation, *Proc. ASME Annual Winter Meeting*, Miami.

Khosla, P.K. and Kanade, T. (1985), Parameter Identification of Robot Dynamics, *I.E.E.E. Conf. Decision and Control*, Fort Lauderdale.

Koditschek, D.E. (1984), Natural Motion of Robot Arms. *Proc. IEEE Conf. on Dec. and Contr.*, Les Vegas, Nev.

Koditschek, D.E. (1987), Adaptive Techniques for Mechanical Systems, *Proc. of Fifth Yale Workshop on the Applications of Adaptive Systems Theory.*

Kreutz, K. and Lokshin, A. (1988). Load Balancing and Closed-Chain Multiple Arm Control, *Proc. American Control Conference.*

Koivo, A.J. and Guo, T.K. (1981), Control of Robotic Manipulator With Adaptive Control, *Proc. IEEE Conf. Decision and Control.*

Landau, I. D. (1985), Adaptive Control Techniques for Robot Manipulators – The Status of the Art, *Proc. IFAC Symp. Robot Control*, Barcelona.

Lee, S.C.G. and Lee B.H. (1984), Resolved Motion Adaptive Control for Mechanical Manipulators, *ASME J. Dyn. Syst. Meas. Contro.*, 106.

Leininger, G.G., and Wang, S. (1982), Pole Placement Self-Timing Control of Manipulators, *I.F.A.C. Symp. Computer-Aided Design of Multivariable Technological Systems.*

Li, W. and Slotine, J.J.E. (1987), Parameter Estimation Strategies for Robot Manipulators, *Proc. of ASME Annual Winter Meeting*, Boston.

Li, W. and Slotine, J.J.E. (1988), Globally Convergent Indirect Adaptive Robot Control. *Proc. Int. Conf. Robotics and Automation*, Philadelphia, PA.

Li, W. and Slotine, J.J.E. (1989a), Indirect Adaptive Robot Control, *System and Control Letters*, vol. 12, pp 259-266.

Li, W. and Slotine, J.J.E. (1989b), Neural Network Control of Uncertain Nonlinear Systems, *Proc. American Control Conference*, Tuscon, Arizona.

Lim, K.Y. and Eslami, M. (1985), New Controller Designs for Robot Manipulators, *Proc. American Control Conference.*

Lozano-Leal, R., and Goodwin, G.C. (1985), A Globally Convergent

Adaptive Pole Placement Algorithm Without a Persistency of Excitation Requirement, *IEEE Trans. Auto. Contr.*, vol. 30, No 8.

Luh, J.Y.S., Walker, M. W. and Paul, R. P. C. (1980a), On-Line Computational Scheme for Mechanical Manipulators *A.S.M.E. J. Dyn. Syst. Meas. Contr.*, No. 102.

Luh, J.Y.S., Walker, M. H. and Paul, R. P. C. (1980b), Resolved Acceleration Control of Mechanical Manipulators, *IEEE Trans. Automatic Control*, vol. 25, No. 3.

Luh, J.Y.S., and Zheng, Y. F. (1985), Computation of Input Generalized Forces for Robots with Closed Kinematic Chain Mechanisms, *IEEE J. of Robotics and Automation*, vol. 1, No. 2.

Mason, M. T. (1981), Compliance and Force Control for Computer Controlled Manipulators, *IEEE Trans. Systems, Man and Cybernetics*, vol. 11, No. 6.

Mason, M.T. and Salisbury, J.K. (1985), Robot Hands and the Mechanics of Manipulation, *M.I.T. Press.*

Middleton, R.H. and G.C. Goodwin (1986), Adaptive Computed Torque Control for Rigid Link Manipulators, *Proc. IEEE Conf. on Dec. and Contr.*, Athens, Greece.

Morgan, A.P. and Narendra, K.S., (1977), On the Uniform Asymptotic Stability of Certain Linear Nonautonomous Differential Equations, *S.I.A.M. J. of Control and Optimization*, vol. 15.

Nicosia, S. and Tomei, P. (1984), Model-Reference Adaptive Control Algorithms for Industrial Robots, 20, pp 635-644.

Niemeyer, G., and Slotine, J.J.E. (1988), Performance in Adaptive Manipulator Control, *Proc. IEEE Conf. Decision and Control*, Austin, TX.

Popov, V.M. (1973), Hyperstability of Automatic Control Systems. *Springer*, New York.

Raibert, M. H. and Craig J. J. (1981), Hybrid Position/force Control of Manipulators, *ASME J. Dynamic Systems, Measurement, Control*, No. 102.

Sadegh, N., and R. Horowitz (1987), Stability Analysis of an Adaptive Controller for Robotic Manipulators. *Proc. IEEE Int. Conf. Robotics and Automation*, Raleigh, NC.

Seraji, H. (1987), Direct Adaptive Control of Manipulators in Cartesian Space, *J. Robotic Systems*, No. 4.

Salisbury, J.K. (1980), Active Stiffness Control of a Manipulator in Cartesian Coordinates, *Proc. IEEE Conf. Decision and Control*, Albuquerque, New Mexico.

Singh, S.N. (1985), Adaptive Model-Following Control of Nonlinear Robotic Systems, *IEEE Trans. Auto. Contr.., AC-30, pp1099-1100.*

Slotine, J.-J. E. *(1983*, Tracking Control of Nonlinear Systems using Sliding Surfaces, *Ph.D. Thesis, Massachusetts Institute of Technology.*

Slotine, J.-J. E. (1984), Sliding Controller Design for Nonlinear Systems, *Int. J. Control.*, vol 40, No 2.

Slotine, J.-J. E. (1985), The Robust Control of Robot Manipulators, *Int. J. Robotics Research*, vol. 4, No. 2.

Slotine, J.J.E., and W. Li (1986), On The Adaptive Control of Robot Manipulators, in Robotics: Theory and Applications (ed. F.W. Paul and K. Youcef-Toumi) from *ASME Winter Annual Meeting*, Anaheim, CA.

Slotine, J.J.E., and W. Li (1987a), Adaptive Robot Control, A Case Study, *Proc. IEEE Int. Conf. Robotics and Automation*, Raleigh, NC.

Slotine, J.J.E., and W. Li (1987b), Adaptive Strategies in Constrained Manipulation, *Proc. IEEE Int. Conf. Robotics and Automation*, Raleigh, NC.

Slotine, J.J.E. and W. Li (1987c), Theoretical Issues In Adaptive Manipulator Control, in *Proc. 5th Yale Workshop on Applications of Adaptive Systems*

Theory.

Slotine, J.J.E. and W. Li (1987d), Adaptive Robot Control - A New Perspective, *Proc. IEEE Conf. Decision and Control* L.A., CA.

Slotine, J.J.E., and W. Li (1987e), On the Adaptive Control of Robot Manipulators, *Int. J. Robotics Res.*, vol 6, No. 3.

Slotine, J.J.E., and W. Li (1988a), Parameter Convergence in Adaptive Robot Control, *Proc. American Control Conference.*

Slotine, J.J.E., and W. Li (1988b), Adaptive Manipulator Control: A Case Study, *IEEE Trans. on Auto. Control*, AC 33-11.

Slotine, J.-J. E. and Sastry S. S. (1983), Tracking Control of Nonlinear Systems Using Sliding Surfaces With Applications to Robot Manipulators, *Int. J. Control*, vol. 39, No. 2.

Slotine, J.-J.E. and Spong, M.W. (1985), Robust Robot Control with Bounded Input Torques, *Int. J. Robotics Systems*, vol. 2, No. 4.

Slotine, J.-J. E., and Yoerger, D.R. (1986), Inverse Kinematics Algorithm for Redundant Manipulators, *Int. J. Robotics and Automation*, vol. 1, No. 2.

Spong, M. (1989), Adaptive Control of Flexible Joint Manipulators, *System and Control Letters*, vol. 13, pp15-21.

Sundareshan, M.K. and Koenig, M.A. (1985), Decentralized Model Reference Adaptive Control of Robotic Manipulators, *Proc. American Control Conf.*, pp 44-49.

Suzuki, H. (1988), Adaptive Control of Welding Processes. Ph.D Thesis. Department of Mechanical Engineering, MIT.

Takegaki, M. and Arimoto, S. (1981), A New Feedback Method for Dynamic Control of Manipulators, *ASME J. Dynamic Systems, Measurement, Control*, No. 102.

Tarn, T.J., Bejczy, A.K. and Li, Z.F. (1988). Dynamic Workspace Analysis of Two Cooperating Robot Arms, *Proc. American Control Conference.*

Walker, M.W. (1988), An Efficient Algorithm for the Adaptive Control of a Manipulator, *Proc. IEEE Int. Conf. Robotics and Automation.*

Zheng, Y.F. and Luh, J.Y.S. (1988). Optimal Load Distribution for Two Industrial Robots Handling a Single Object, *Proc. IEEE Int. Conf. on Robotics and Automation.*