Adaptive Control of NTV Plants without Persistent Excitations. An Application in Robotics

by

Jing Yuan
B.S. North Jiao-Tong University, 1982
M.S. University of Akron, 1987

A Dissertation Submitted in Partial Fulfillment of the Requirements for the Degree of

DOCTOR OF PHILOSOPHY

in the Department of Mechanical Engineering

ACCEPTED
FACULTY OF GRADUATE STUDIES

______________________ DEAN

DATE 91/07/23

We accept this dissertation as conforming to the required standard.

____
Dr. Y. Stepanenko, Supervisor (Mechanical Engineering)

____
Dr. Z. Dong, Department Member (Mechanical Engineering)

____
Dr. G. Mclean, Department Member (Mechanical Engineering)

____
Dr. M. van Emden, Outside Member (Computer Science)

____
Dr. D. Cherchas, External Examiner (Mechanical Enginerring)

University of Victoria

Supervisor: Dr. Yury Stepenanko

Abstract

Adaptive control of a nonlinear time varying (NTV) plant, such as a robotic manipulator, is intended to tolerate the unmodeled disturbances and the uncertain parameters of the dynamic model. Most of the previous research has been focused on NTV plants with bounded and "slowly-varying" plant terms. Almost all adaptive controllers require persistent excitations to guarantee stable tracking in the presence of unmodeled disturbances.

The new adaptive controllers developed in this work provide stable and robust performance without persistent excitations and the "slowly-varying" assumption. Moreover, the uncertainties of a NTV plant model are not required to be bounded. This allows one to treat some potentially unbounded dynamics as disturbances. Stability and robustness analysis of adaptive controllers under the relaxed conditions is an essential part of this study.

A major problem arising in robotic control is parameter uncertainty. The linear parameterization approach is also implemented in this work to deal with the parameter uncertainty. An innovative algorithm for determining the manipulator "regressor" (a coefficient matrix in parameter-linearized form of robot dynamics) is developed. Based on this algorithm a robust self-tuning controller is designed. The control law is proved to be robust with respect to parameter errors and disturbances. The robustness of the controller relaxes the requirement for the parameter estimator, and leads to a stable system without persistent excitations.

Examiners:

Dr. Y. Stepanenko, Supervisor (Mechanical Engineering)

Dr. Z. Dong, Department Member (Mechanical Engineering)

Dr. G. Mclean, Department Member (Mechanical Engineering)

Dr. M. van Emden, Outside Member (Computer Science)

Dr. D. Cherchas, External Examiner (Mechanical Enginerring)

Acknowledgements

The author would like to thank his supervisor, Professor Yury Stepanenko of Department of Mechanical Engineering, for his encouragement, patience and advice during the course of this research, and for his help in preparation of this dissertation.

Financial assistance received from Professor Yury Stepanenko (through the National Science and Enginnering Research Councile of Canada, the Institute of Robotic and Intelligent Systems and Precarn Association Incorporation) is also gratefully acknowledged.

To my parents Yulin Hwang and Xiaosheng Yuan
For their love and constant encouragement.

Contents

1 Introduction 6

1.1 Problem Statement 6

1.2 Problems to be Solved 9

1.3 Organization of the Thesis 13

I Background 15

2 Model Reference Adaptive Controllers (MRAC) 16

2.1 MRAC for First Order LTI Plants 16

2.2 MRAC for Multi-variable LTI Plants: 18

2.3 MRAC for High Order LTI Plants 20

2.4 MRAC for LTV and NTV Plants 24

2.5 Summary 26

3 Manipulator Dynamics 27

3.1 Th Euler–Lagrange Equation 28

3.2 The Newton–Euler Algorithm 33

3.2.1 The outward iterations 35

3.2.2 The inward iterations 35

3.3 Summary 36

4 Manipulator Controllers **38**

4.1 PD Controller with Gravity Compensation 40

4.2 Computed Torque Method 41

4.3 Variable Structure Control 41

4.4 Adaptive Control of Manipulators 45

4.4.1 MRAC applied to robots 45

4.4.2 Linear parameterization 46

4.4.3 Adaptive control by linear parameterization 47

4.5 Summary 48

II Contributions **50**

5 Adaptive Control of LTV and NTV Plants **51**

5.1 Single Input/Single-Output Plants 52

5.1.1 Adaptive controller design 53

5.1.2 Stability analysis 55

5.1.3 Simulation examples 56

5.2 Multi-variable Plants 57

5.2.1 Adaptive controller design 59

5.2.2 Stability analysis 60

5.2.3 Discussions 61

5.3 An Application Example 62

5.4 LTV Plants of Relative Degree 1 72

5.4.1 System analysis 72

5.4.2 The effect of θ_g^* 76

5.5 Summary 80

6 Adaptive Control of Robots with Uncertain Models 81

6.1 Some Dynamic Properties 82

6.2 The Adaptive Controller Design 87

6.3 Stability Analysis 89

6.3.1 Robustness with respect to unmodeled dynamics 90

6.3.2 Robustness with respect to measurement noises 94

6.4 Discussion 99

6.5 Simulation Results 100

6.5.1 Step response of the system 101

6.5.2 Effect of the K matrix 101

6.5.3 Tracking at different speeds 101

6.5.4 Tracking with a varying load 104

6.5.5 Tracking in measurement noise 104

6.6 Summary 108

7 A Robust Self-Tuning Computed-Torque Controller 113

7.1 Robustness of the Controller 114

7.2 Parameter Uncertainties 117

7.3 Linear Parameterization 122

7.3.1 Linearizing $A(\dot{q}, q, \zeta_r)$ with respect to ζ_r 124

7.3.2 Linearizing $B(\dot{q}, q, \zeta_r)$ with respect to ζ_r 127

7.4 Parameter Identification 128
7.4.1 Identification under ideal conditions 129
7.4.2 Identification in the presence of external disturbances . . . 130
7.4.3 The QR algorithm 131
7.4.4 The Householder transforms 133
7.5 A Simulation Example 135
7.6 Summary 140

8 Conclusions **141**

List of Figures

2.1 Convergence of parameter adaptation. 25

3.1 The i-th joint frame of a robot. 29

5.1 Tracking error of an NTV plant with bounded coefficients. 57

5.2 Tracking error of an NTV plant with unbounded coefficients. . . . 58

5.3 Tracking errors of a manipulator (without training). 67

5.4 Tracking errors of a manipulator (with training). 68

5.5 Torques applied to the manipulator. (with training). 69

5.6 Torques applied to the manipulator (without training). 70

5.7 Tracking error of a LTV plant with θ_g. 78

5.8 Tracking error of a LTV plant without θ_g. 79

6.1 Block diagram of the control system. 88

6.2 Step responses of the two joints. 102

6.3 Effect of the feedback gain matrix K. 103

6.4 Slow tracking performance. 105

6.5 Fast tracking performance. 106

6.6 Tracking with a varying load. 107

6.7 Slow tracking with measurement noise. 109

6.8 Fast tracking with measurement noise. 110

7.1 A two link planar robot. 136

7.2 Performance of the self-tuning controller. 139

Nomenclature

Symbols associated with the system dynamic equation (1.1):

$q \in R^n$	Generalized coordinates of a mechanical system.
$M(q), V(\dot{q}, q) \in R^{n \times n}$	Nonlinear matrix functions of the system dynamic model.
$V_m(q) \in R^{n \times n^2}$	A matrix function such that $V(\dot{q}, q)x = V_m(q)[\dot{q}x] \ \forall \ x \in R^n$.
$[\dot{q}x] \in R^{n^2}$	A vector: $[\dot{q}_1 x_1, \ldots, \dot{q}_1 x_n, \ldots, \ldots, \dot{q}_n x_1, \ldots, \dot{q}_n x_n]^T$.
$G(q) \in R^n$	The gravity force/torque vector.
$\tau \in R^n$	The generalized control torque vector.

Symbols used in Chapter 2:

x_p	Plant output.
u	Control input.
r	Reference model input.
a_p, b_p	Constant plant coefficients.
θ_a^*, θ_b^*	Ideal controller parameters.
θ_a, θ_b	Adaptive controller parameters.
$\phi_a = \theta_a - \theta_a^*, \phi_b = \theta_b - \theta_b^*$	Compensation errors.
y_p	Plant state vector.
u	Control input vector.
r	Reference model input vector.
A_p, B_p, g_p	Plant coefficient matrices.
Θ^*	Ideal controller parameter matrix.
Θ	Adaptive controller parameter matrix.
$\Phi = \Theta - \Theta^*$	Compensation error matrix.
ω	Intermediate feedback vector.

Symbols used in Chapter 3:

${}^{i}_{k}R \in R^{3\times3}$	A rotation matrix between frame i and frame k.
$v_{c_i} \in R^3$	Linear velocity of the mass center of the i-th link.
$\omega_i \in R^3$	Angular velocity of the i-th link.
$J_{c_i}(q) \in R^{6\times n}$	Jacobian matrix with respect to the mass center of the i-th link.
$z_i \in R^3$	Axis of the i-th joint (unit vector).
$P_{ik} \in R^3$	Vector points from the i-th frame origin to the k-th frame origin.
c_k	The mass center vector with respect to the k-th frame.
$Q_i \in R^{6\times6}$	Mass matrix of the i-th link.
$m_{ij}(q)$	The ij-th entry of matrix $M(q)$.
$T_i \in R^{3\times3}$	Inertia tensor of the i-th link.
${}^{i+1}\omega_i \in R^3$	Angular velocity of the i-th link expressed in the $(i+1)$ -th frame.
${}^{c_i}T_i \in R^{3\times3}$	Matrix T_i expressed in the i-th mass center.
${}^{i}F_i \in R^3$	Dynamic force of the i-th link acting on the i-th joint.
${}^{i}N_i \in R^3$	Dynamic torque of the i-th link acting on the i-th joint.

Symbols used in Chapter 4:

$K_v, K_p \in R^{n\times n}$	Velocity and position feedback matrices.
$e = q - q_d \in R^n$	Tracking error vector.
$\hat{M}, \hat{V}$ and $\hat{G}$	Estimates of $M(q)$, $V(\dot{q}, q)$ and $G(q)$ respectively.
$\epsilon = [\dot{e}^T, e^T]^T \in R^{2n}$	Error state vector.
$\rho(\epsilon, t)$	A scalar function bound.
$w \in R^{2n}$	Intermediate vector.

Symbols used in Chapter 5:

$a_p(y_p,t), b_p(y_p,t)$	NTV plant coefficients.
θ_a^*, θ_b^*	NTV ideal controller parameters.
θ_a, θ_b	Adaptive controller parameters.
$A_p(y_p,t), B_p(y_p,t), g_p(y_p,t)$	NTV plant coefficient matrices.
Θ^*	NTV ideal controller parameter matrix.
Θ	Adaptive controller parameter matrix
$\Phi = \Theta - \Theta^*$	Compensation error matrix.
ω	Intermediate feedback vector.

Symbols used in Chapter 6:

$K > 0, \Lambda > 0 \in R^{n \times n}$	Constant positive definite gain matrices.
$\lambda_k, \lambda_\Lambda \in R$	Smallest eigenvalues of K and Λ respectively.
A_M, A_V, A_G	Adaptive matrices.
$\Phi_M = A_M - M(q)$	
$\Phi_V = A_V - V_m(q)$	
$\Phi_G = A_G - G(q)$	
$\Phi = [\Phi_M, \Phi_V, \Phi_G]$	System compensation error matrices.
$e = q - q_d \in R^n$	The tracking error vector.
$s = \dot{e} + \Lambda e \in R^n$	
$\psi = \dot{q}_d - \Lambda e \in R^n$	Intermediate vectors.
$[\dot{q}\psi] \in R^{n^2}$	An intermediate vector similar to $[\dot{q}x]$.
$\alpha,\ \sigma_0,\ \sigma_1$	Positive parameters in the adaptive controller.
$n_p, n_v \in R^n$	Position and velocity measurement noise vectors.
$e_n = q + n_p - q_d \in R^n$	Noisy position error vector.
$\dot{e}_n = \dot{q} + n_v - \dot{q}_d \in R^n$	Noisy velocity error vector.
$[\dot{q}_n\psi_n] \in R^{n^2}$	A noisy version of $[\dot{q}\psi]$.

Symbols used in the robustness study (Theorems 6.1 and 6.2):

$\tau_d \in R^n$	Unmodeled dynamics.
$\tilde{\tau}_d \in R^n$	Equivalent disturbance with both unmodeled dynamics and measurement noise.
Γ	An intermediate matrix defined in (6.20).
$\rho,\ \beta$	Scalar bounds related to Γ and defined in (6.22).
σ_n	A noisy version of σ computed with $\lVert\dot{q}_n\rVert$.
$\{c_i\},\ \{d_i\},\ \{\gamma_i\}$	
$\{a_i\},\ \{p_i\},\ \gamma$	Positive constant scalar bounds:

Symbols used in Chapter 7:

$F(\dot{q}, q) \in R^n$	Low-pass filtered robot dynamics.
$\frac{\partial}{\partial\zeta}F(\dot{q}, q) \in R^{n\times l}$	The regressor matrix.
$\zeta_r \in R^l$	Robot inertia parameter vector.
$A(\dot{q}, q, \zeta) \in R^n$	
$B(\dot{q}, q, \zeta) \in R^n$	Component vector functions of $F(\dot{q}, q)$.
a_i	The i-th element of $A(\dot{q}, q, \zeta)$.
$\tilde{T}_i$	Equivalent inertia tensor of the i-th link.
$S(y) \in R^{3\times 3}$	Skew-symmetric matrix such that $S(y)x = y \times x$.
$\Lambda({}^n\omega_n) \in R^{3\times 6}$	Intermediate matrix such that. $\tilde{T}_i\ {}^n\omega_n = \Lambda({}^n\omega_n)\rho$ where $\rho = [\tilde{T}_{xx}, \tilde{T}_{yy}, \tilde{T}_{zz}, \tilde{T}_{xy}, \tilde{T}_{xz}, \tilde{T}_{yz}]^T$.
$\Gamma(z) \in R^{n\times 10}$	Intermediate matrix function such that $A(\dot{q}, q, \zeta) = \Gamma(z)\zeta$.
$\aleph \in R^{n\times 10}$	Intermediate matrix function such that $B(\dot{q}, q, \zeta) = [\Gamma(\dot{z}) - \aleph]\zeta$.
$u = \frac{1}{D+\sigma}\tau \in R^n$	Low-pass filtered torque where D is a differential operator; σ is a positive constant.
$\zeta_m \in R^l$	Estimate of ζ_r.
$\mathcal{Q}, \mathcal{R} \in R^{m\times m}$	Matrices involved in the QR algorithm.
$\mathcal{H} \in R^{m\times m}$	Housholder reflector.

Chapter 1

Introduction

1.1 Problem Statement

In control theory, a physical system to be controlled is called a "plant". The controller design problem is to determine a proper input to a given plant such that the plant output follows some prescribed reference signal or the output of a prescribed reference model. The mathematical relationship between the input and output is called the plant model. As suggested by the title of this thesis, the focus of this study is a class of plants whose mathematic models are described by nonlinear differential equations. A particular example is the dynamics of a mechanical manipulator or a robot, which is a most frequently studied subject/example of nonlinear control theory. The main purpose of this study is to develop adaptive controllers that apply to a class of nonlinear plants. These nonlinear plants include almost all mechanical systems satisfying the general Lagrange equation. They are best represented by a rigid-body robotic manipulator. For this reason, the focus will be frequently directed towards a robot control problem without lossing generality.

An important problem in robotic research and development is accurate control of robotic manipulators. The function of a controller is to apply an appropriate torque vector τ to the joints of a robot such that the manipulator follows a desired trajectory. The dynamics of a general n-link robotic manipulator is represented by a second-order multi-variable differential equation

$$M(q)\ddot{q} + V(\dot{q}, q)\dot{q} + G(q) + \tau_d = \tau \tag{1.1}$$

where $q \in R^n$ denotes the joint position of the robot, $M(q)$, $V(\dot{q}, q) \in R^{n\times n}$ and $G(q) \in R^n$ are nonlinear functions of the joint position q and the joint velocity $\dot{q}$. $M(q)$ is the inertia matrix, $V(\dot{q}, q)$ represents the effects of centrifugal and coriolis forces while $G(q)$ represents the gravity force; $\tau_d \in R^n$ is a vector representing the external disturbances which may includes some unmodeled dynamics.

Generally, it is impossible to build a perfect dynamic model for a physical system. There are two main limitations on the modelling problem. First, the dynamics of many physical systems depend on some system parameters which may not be measured with sufficient accuracy. Sometimes, the system parameters may vary in some unpredictable ways. The parameter error of a physical system could affect the accuracy of the corresponding model, (which is exactly the case of robotic manipulators). Secondly, the interactions between a physical system and the environment may involve a number of unknown factors. These effects may not be modeled mathematically because of their "small" magnitudes, or because of the lack of information about their physical nature. The resulting model is always a trade-off between completeness and feasibility.

Because of the above factors, the mathematic model of a physical system should be considered as uncertain. In this study, the uncertainties are classified into two groups: the "uncertain but modeled dynamics" (*u.m.d.*) and the "unmodeled disturbances" (*u.d.*). The first group implies some degree of knowledge about the structure of the system dynamics. For example, if one neglects τ_d in (1.1), then the *u.m.d.* is represented by $M(q)\ddot{q} + V(\dot{q}, q)\dot{q} + G(q) = \tau$ which provides information about the order of the differential equation, the linearity with respect to the acceleration $\ddot{q}$ and input torque τ as well as the indication that $M(q)$, $V(\dot{q}, q)$ and $G(q)$ are functions of a set of system parameters. The system parameters of a robot consist of the mass, center of mass and inertia tensor of each link. The main source of *u.m.d.* is the uncertainty of the inertia parameters.

The second group includes completely unknown disturbances subject to some assumptions, such as all bounded disturbances. Their effects are considered as a whole, and denoted by a simple symbol τ_d in this thesis.

Almost all available high performance controllers for robotic manipulators try to deal with the nonlinear dynamics by either feed-forward compensation or feedback linearization. Examples of these controllers include the well known "Computed torque" [1]–[4], "Resolved acceleration" [5], "Direct design" [6] and "Interception" [7] methods. All of them require an exact dynamic model of the manipulator. Because of the *u.m.d.*, however, it may be impossible to obtain accurate plant terms $M(q)$, $V(\dot{q},q)$ and $G(q)$ even though there exist some algorithms to compute them. If approximate values of robot parameters are used, inaccurate dynamic terms $\hat{M}$ $\hat{V}$ and $\hat{G}$ are computed by the control algorithm. The controller will attempt to control an artificial robot whose equation of motion is

$$\hat{M}(q)\ddot{q} + \hat{V}(\dot{q},q)\dot{q} + \hat{G}(q) = \tau$$

instead of (1.1). A stable feedback controller designed for the artificial robot may not perform satisfactorily with the real robot (1.1). This is especially true when a controller uses feedback linearization $\hat{M}\ddot{q}_d + \hat{V}\dot{q} + \hat{G}$ to compensate the nonlinear dynamics. The stability of the closed-loop system may be at stake because of the inaccurate compensations.

An adaptive controller designed for an uncertain plant should be able to tolerate both *u.m.d.* and *u.d.*. In many practical applications, it is very difficult, if not impossible, to achieve asymptotic stability for adaptive control of a general nonlinear time-varying uncertain plant. The adaptation law depends on the feedback information provided by the tracking errors to update the controller parameters. However, the controller should provide stable tracking within a certain computable tolerance. A Lyapunov-type stability analysis is the focal point of the current research.

1.2 Problems to be Solved

In order to explain the essence and importance of the problem to be addressed in the thesis, it would be helpful to provide an outline of the most significant results in this area. Detailed reviews of the works mentioned here will be given later in Chapters 2 and 4 respectively.

In the last few years, significant progress has been made on adaptive control of linear time-invariant (LTI) plants [10]–[16]. Stable and robust adaptive laws have been developed which guarantee zero tracking error even when the plant terms are completely unknown and some plant states are not available. Input-to-state stability of nonlinear systems is investigated in [17] for a general nonlinear model, and conditions for the existence of co-prime factorization are established. The feedback linearization problem in the presence of unknown parameters and unmodeled dynamics is considered in [18]; conditions are given for global stability of reduced-order models. Stability of a specific class of "pure-feedback" systems has been analyzed in [19]. Other researchers have focused on linear-time-varying (LTV) plants [20]–[23]. Middleton and Goodwin [20] establish global robust stability for LTV plants with slowly-varying parameters, or bounded parameters with infrequent jumps; Parameter-adaptive control of LTV plants whose parameters belong to a convex region is presented in [21] for a case where unmodeled dynamic effects can be bounded by some known functions. Tsakalis and Ioannou proposed a new model reference control (MRC) structure. It is able to compensate fast varying plant terms as long as they are completely known to the designer. When the plant terms are unknown, the slowly varying assumption is relaxed by making use of some "structural knowledge" about the plant terms [24, 25]. This means the plant terms are known functions of time multiplied with slowly varying coefficients. (For example, an unknown plant term $a_p(x,t)$ may be assumed to be $a_p(x,t) = \sum_{i=1}^{n} a_i(x,t)p_i$ where $\{a_i(x,t)\}_{i=1}^{n}$ are assumed to be known, fast varying functions whereas $\{p_i\}_{i=1}^{n}$ are unknown, slowly varying coefficients.) Only the slowly varying coefficients (such as $\{p_i\}_{i=1}^{n}$) can be estimated and adjusted by the adaptive laws while the information on possible fast varying effects must be provided by some known functions (such as $\{a_i(x,t)\}_{i=1}^{n}$). It is commonly recognized that in order to ensure stability in the presence of unmodeled disturbances, an additional condition called "persistent excitation" is required [26]. (The persistent excitations are system dependent. For the particular case of adaptive robot control, the persistent excitation will be explained in page 16, Eq.(1.3).)

The development of adaptive controllers for robotic manipulators follows a similar pattern. Many early researchers treat the nonlinear dynamics of (1.1) as a multi-dimensional LTI plant by assuming $M(q)$, $V(\dot{q},q)$ and $G(q)$ to be constant. Adaptive controllers developed for LTI plants, such as the "modeled reference adaptive controllers" (MRAC) are directly applied to control robotic manipulators [27]–[34]. The stability analysis of these adaptive controllers is based on an unrealistic assumption that the plant terms change "very slowly" or equivalently $\dot{M}(q) = 0$, $\dot{V}(\dot{q},q) = 0$ and $\dot{G}(q) = 0$. Lim and Eslami [35] tried to relax the "slowly-varying" assumption by some mini-max analysis which assumes a bound on a $||\dot{M}(q)||$, $||\dot{V}(\dot{q},q)||$ and $||\dot{G}(q)||$. Seraji [36] rewrites the plant model (1.1) in n independent equations with uncertain terms representing dynamic coupling between the manipulator links. In the stability analysis he assumed that the inertia and the coupling terms are slowly-varying, and the latter satisfies an inequality condition, involving the negative part of the derivative of Lyapunov function.

The "slowly-varying" assumption ignores some potentially unstable effects. For example, the fact that $\dot{f}(x) = \frac{df}{dx}\dot{x}$ suggests that the norms of $\dot{M}(q)$ and $\dot{G}(q)$ could be proportional to $||\dot{q}||$. The entries of matrix $V(\dot{q},q)$ contain the products of $\dot{q}_i$, the components of the velocity vector $\dot{q}$. Thus $||V(\dot{q},q)||$ itself could be proportional to $||\dot{q}||$ and the expression of $||\dot{V}(\dot{q},q)||$ even involves the components of $\ddot{q}$ and could be proportional to $||\ddot{q}||$. Pre-assuming bounds on $||\dot{M}(q)||$, $||\dot{V}(\dot{q},q)||$ and $||\dot{G}(q)||$ is equivalent to assuming that $\dot{q}$ is bounded before one attempts to prove it. However, adaptive controllers based on the slowly-varying assumption perform well in simulations or experiments [36].

While simulations or experiments serve as an important way to evaluate the performance of a controller design, they can not prove whether the closed-loop system is stable or not. It is impossible to implement a simulation or experiment that exhausts all possible trajectories under all working conditions. An unstable controller could demonstrate excellent performance under certain conditions and/or along some specific trajectories. From this point of view, it is equally, if not more, important to study the stability problem analytically and find solid conditions that guarantee stable tracking for a closed-loop system. Craig, Hsu and Sastri [37, 38] were perhaps the first to design an analytically stable adaptive controller for robotic manipulators. They introduced the idea of linear parameterization and make use of the fact that

$$M(q)\ddot{q} + V(\dot{q}, q)\dot{q} + G(q) = Y(\ddot{q}, \dot{q}, q)\zeta \tag{1.2}$$

where $Y(\ddot{q}, \dot{q}, q)$ is called a "regressor", which is a known matrix function of q, $\dot{q}$ and $\ddot{q}$; ζ is a set of constant parameters. The components of vector ζ are combinations of mass, center of mass and inertia tensor of the last link. Slotine and Li [39, 40] improved Craig's idea with an elegant control law that avoided the acceleration feedback and the inversion of the estimated inertia matrix $\hat{M}(q)$. Another way to avoid $\ddot{q}$ was suggested by Hsu *et. al.* [41], Middleton and Goodwin [42]. They proposed to filter $Y(\ddot{q}, \dot{q}, q)$ and get another regressor $W(\dot{q}, q)$.

Although the computation of $Y(\ddot{q}, \dot{q}, q)$ has been studied by Atkeson, An and Hollerbach [43] as well as Khosla and Kanade [44], the regressor $Y(\ddot{q}, \dot{q}, q)$ is seldom used because of its dependence on the acceleration feedback $\ddot{q}$. Slotine and Li use an alternative $Y(\ddot{q}_d, \dot{q}_d, \dot{q}, q)$ which uses the desired acceleration $\ddot{q}_d$ and desired velocity $\dot{q}_d$ as part of its arguments. However, there exist some difficulties separating the arguments $\dot{q}$ and $\dot{q}_d$. No solutions to these difficulties have ever been reported. An algorithm to compute $W(\dot{q}, q)$ for a typical six-joint robot is also not available yet. The stability analysis of many linear parameterization adaptive controllers ignores the unmodeled disturbances τ_d. As in the case of general adaptive control, some additional condition called the persistent excitation is required to ensure stability if $\tau_d \neq 0$. For robotic dynamics, the persistent excitation is represented by [37, 45]

$$0 < d_1 I \leq \int_t^{t+\Delta t} Y(\ddot{q}, \dot{q}, q) Y^T(\ddot{q}, \dot{q}, q) dt \leq d_2 I \tag{1.3}$$

where Y is the regressor; $0 < d_1 < d_2$ and $0 < \Delta t$. It is very difficult to plan a desired trajectory that satisfies (1.3) without exhaustive trial-and-error computations. In reality, the actual trajectory measurements $\ddot{q}$, $\dot{q}$ and q are inevitably different from their desired counterparts $\ddot{q}_d$, $\dot{q}_d$ and q_d. This fact makes it difficult to predict, in advance, whether the actual trajectory provides persistent excitation or not.

Two major aspects, regarding to the above mentioned problems, will be concerned in this study:

1. It appears that the only available way to remove the slowly-varying assumption is by means of linear parameterization which implies heavy computations. In this study, a simple alternative is sought. The new adaptive controllers adjust some adaptive matrices A_M, A_V and A_G to compensate the effect of $M(q)$, $V(\dot{q}, q)$ and $G(q)$. The resulting adaptive controller designs no longer require the slowly-varying assumption to prove their robustness in the presence of both kinds of uncertainties as well as the measurement noise introduced in the feedback $\dot{q}$ and q. The robustness results are independent of the linear parameterization and its computational demanding regressors $Y(\ddot{q}, \dot{q}, q)$ or $W(\dot{q}, q)$. This approach allows one to design economic adaptive controllers.

2. An innovative algorithm to compute $W(\dot{q}, q)$ for a general n-link robot is developed. It is computational efficient in that many of the variables are directly available from the Newton-Euler algorithm, which is implemented to synthesize the control law. In order to avoid the restrictive condition of persistent excitation, the robustness of a computed-torque controller is studied. It is proved that the computed-torque controllers are able to tolerate both kinds of uncertainties while maintaining stable tracking. The robustness of the controller reduces the requirement of the adaptive law, and provides sufficient time for parameter identification. If the actual trajectory provides persistent excitation, then the adaptive law will identify the exact parameters; otherwise, it will solve a set of bounded parameters that minimize both kinds of uncertainties in the least-square sense.

1.3 Organization of the Thesis

The rest of the thesis consists of two major parts: background and contributions.

The background contains three chapters. Chapter 2 explains the different versions of available adaptive controllers. It covers all typical MRAC schemes for general LTI plants and some recent development on LTV and NTV plants. This chapter provides some fundamental knowledge about adaptive controllers in order to understand the design of new adaptive controllers for LTV, NTV plants and robotic manipulators to be presented in Chapters 5 and 6. Chapter 3 is devoted to the robotic dynamic model (1.1). It will explain two different approaches for evaluating the seemingly simple dynamic terms $M(q)$, $V(\dot{q}, q)$ and $G(q)$ which actually involve large amounts of computations. It also addresses the crucial influence of inertia parameters on the accuracy of the robot model. This chapter is closely related to the new self-tuning controller developed later in Chapter 7 and the algorithm for computing the regressor $W(\dot{q}, q)$. Some typical controllers currently available for robotic manipulators are presented in Chapter 4. Their stability analysis and the corresponding conditions and assumptions will be reviewed in detail. Such a review provides a base to compare the performance of the new adaptive controllers with that of the existing controllers.

The contribution is also organized into three chapters. In Chapter 5, an improved MRAC design is presented. It can be applied to a large class of LTV or NTV plants with fast varying and potentially unbounded plant terms. This is a significant improvement on those MRAC designs reviewed in Chapter 2. While the new MRAC controller can be applied directly to the robotic tracking problem as demonstrated by simulation examples, it does not take advantage of the special structural properties of robot dynamics. These properties are explored in Chapter 6 and a new adaptive controller is presented on such ground. The new adaptive controller does not depend on linear parameterization to avoid the slowly-varying assumption. Instead, it takes advantage of the general properties of the open-chain articulated mechanisms. The improved control and adaptation laws enable the closed-loop system to track trajectories at full speed. The controller is robust with respect to some potentially unbounded disturbances and possible measurement noise introduced in the feedback q and $\dot{q}$. Both position and velocity errors are proved to converge into a computable tolerance range within finite time. Finally, a robust self tuning controller is presented in Chapter 7. It is aimed at improving the linear parameterization approach. The robustness study is conducted on a closed-loop without involving the adaptive law. The control law is proved to be robust even when the inertia parameters are inaccurate. This arrangement allows the adaptive law sufficient time to identify the correct inertia parameters. The restrictive persistent excitation condition is thus eliminated. An algorithm is also developed to compute the regressor $W(\dot{q}, q)$ for a general n-link robot, which is the first reported algorithm for this purpose.

Part I

Background

Chapter 2

Model Reference Adaptive Controllers (MRAC)

MRAC is originally developed for LTI plants with unknown plant terms. It is closely related to a class of non-adaptive controllers, the model reference controllers (MRC). The design of a MRC requires complete knowledge about a LTI plant in order to compensate the plant dynamics such that the plant follows a prescribed model. Adaptive strategies are needed when the necessary knowledge about a plant is not available. Thus a MRAC is actually a MRC plus a proper adaptation law. The adaptation law provides additional freedom to adjust the coefficients of a MRC. It determines the stability of the overall closed-loop system. Therefore it is very important to design a proper adaptation law for a MRAC system. In this chapter, several typical adaptation laws are reviewed.

2.1 MRAC for First Order LTI Plants

Consider a first order LTI plant given by

$$\dot{x}_p = a_p x_p + b_p u \tag{2.1}$$

where u and x_p denote the input and output of the plant; a_p and b_p are the unknown constant plant parameter. It is commonly assumed that the sign of b_p

is known. Thus without loss of generality, one can assume $b_p > 0$.

The objective is to determine input u such that x_p is as close to a reference model output x_m as possible. The prescribed model satisfies a first order equation:

$$\dot{x}_m = -a_m x_m + b_m r \tag{2.2}$$

where $a_m > 0$, $b_m > 0$ are constant and r is the input signal that controls x_m to follow a desired trajectory. The name of "model reference adaptive control" is easily understood from this simple system — a given plant is required to behave like a prescribed model.

Let $e = x_p - x_m$ be the tracking error. Subtracting (2.2) from (2.1), one obtains

$$\dot{e} = -a_m e + b_p(\theta_a^* x_p + u - \theta_b^* r), \tag{2.3}$$

where $\theta_a^* = b_p^{-1}(a_m - a_p)$ and $\theta_b^* = b_p^{-1} b_m$.

Clearly, if u is calculated in such a way that the second term in the right side of (2.3) is zero, then e will eventually converge to zero. This can be realized when both θ_a^* and θ_b^* are known, and the controller designed in such a way is a MRC. When the exact values of θ_a^* and θ_b^* are not available to the designer, one can only write

$$u = \theta_b r - \theta_a x_p \tag{2.4}$$

where θ_a and θ_b are to be determined by an adaptive law

$$\dot{\theta}_a = e x_p, \quad \text{and} \quad \dot{\theta}_b = -er. \tag{2.5}$$

Substituting (2.4) into (2.3) leads to

$$\dot{e} = -a_m e + b_p(\phi_a x_p + \phi_b r) \tag{2.6}$$

where $\phi_a = \theta_a^* - \theta_a$ and $\phi_b = \theta_b - \theta_b^*$.

Theorem 2.1 *The tracking error of closed-loop system (2.6) and (2.5) is asymptotically stable and the tracking error will eventually converge to zero.*

Proof: Let us consider a Lyapunov function candidate

$$L = \frac{1}{2}[e^2 + b_p(\phi_a^2 + \phi_b^2)].$$

The time derivative of L is written as

$$\dot{L} = e\dot{e} + b_p(\phi_a\dot{\phi}_a + \phi_b\dot{\phi}_b).$$

Substituting (2.5), (2.6) and noting that $\dot{\phi}_a = -\dot{\theta}_a$ $\dot{\phi}_b = \dot{\theta}_b$, the above expression becomes

$$\dot{L} = -a_m e^2 \leq 0.$$

where $\dot{L} = 0$ if and only if $e = 0$. Thus L will keep on decreasing until $e = 0$. **Q.E.D.**

The adaptation law (2.5) consists of two integrators. The sign of integration is determined by the feedback information e. For unknown constant parameters like θ_a^* and θ_b^*, pure integration operations are sufficient to force the corresponding compensation errors ϕ_a and ϕ_b to converge to zero.

2.2 MRAC for Multi-variable LTI Plants:

The MRAC scheme can be applied to multi-variable plants given by

$$\dot{y}_p = -A_p y_p + B_p u + g_p$$

such that the plant follows a prescribed model

$$\dot{y}_m = -A_m y_m + B_m r.$$

Similar to the case of single variable systems, $A_p, A_m \in R^{n\times n}$, $B_p, B_m \in R^{n\times l}$ are constant matrices; $y_p, y_m \in R^n$, and $u, r \in R^l$ with $l \leq n$. The plant-model pair is assumed to satisfy the perfect matching condition [16]

$$(I - B_p B_p^\dagger)(A_p - A_m) = 0 \quad \text{and} \quad (I - B_p B_p^\dagger)B_m = 0 \quad \forall\, y_p, t$$

where $B_p^{\dagger} = (B_p^T B_p)^{-1} B_p^T$. The control law is given by

$$u = \Theta_b u + \Theta_a y_p + r = \Theta \omega + r \tag{2.7}$$

where $\Theta = [\Theta_b, \Theta_a]$ and $\omega^T = [u^T, y_p^T]$. There exist $\Theta_b^* = I - (B_p^{\dagger} B_m)^{-1}$, $\Theta_a^* = B_p^{\dagger}(A_p - A_m)$, and $\Theta^* = [\Theta_b^*, \Theta_a^*]$ such that the plant is expressed as

$$\dot{y}_p = -A_m y_p + B_m(\Phi\omega + r) \tag{2.8}$$

where $\Phi = \Theta - \Theta^*$. Like the previous problem, a proper MRC can be obtained by directly substituting $\Theta = \Theta^*$, in which case (2.8) becomes exactly the same as the prescribed reference model because Φ becomes an all zero vector.

A MRAC is needed when Θ^* is unknown. Let $e = y_p - y_m$ denote the tracking error. It satisfies

$$\dot{e} = -A_m e + B_m \Phi \omega. \tag{2.9}$$

The adaptation law is given by

$$\dot{\Theta} = \dot{\Phi} = -B_m^T P e \omega^T \tag{2.10}$$

where $P = P^T > 0$ satisfies $0.5(A_m^T P + P A_m) = Q = Q^T > 0$.

Theorem 2.2 *The closed-loop system (2.9) and (2.10) is asymptotically stable and the tracking error e will eventually converge to zero.*

Proof: Consider a Lyapunov function candidate

$$L = e^T P e + Tr\{\Phi^T \Phi\}$$

where $Tr\{X\}$ denotes the trace of a matrix X. The time derivative of L evaluated along (2.9) is given by

$$\dot{L} = -e^T Q e + 2e^T P B_m \Phi \omega + 2Tr\{\dot{\Phi}^T \Phi\}.$$

Using the fact that $e^T P B_m \Phi \omega = Tr\{\omega e^T P B_m \Phi\}$ and substituting (2.10), one can cancel the last two terms in the above equation to write

$$\dot{L} = -e^T Q e \leq 0.$$

This means that L will be decreasing until $e = 0$. **Q.E.D.**

2.3 MRAC for High Order LTI Plants

For LTI plants of order n, the general dynamic equation is

$$y_p + \sum_{k=1}^{n} \alpha_k \frac{d^k y_p}{dt^k} = k_p(u + \sum_{i=1}^{m} \beta_i \frac{d^i u}{dt^i})$$

where $\{\alpha_k\}$, $\{\beta_i\}$ and k_p are unknown constant parameters. The sign of k_p is assumed to be known *à priori*. Without loss of generality, it is assumed that $k_p > 0$. The coefficients $\{\beta_i\}_{i=0}^{m}$ must satisfy a necessary condition that all roots of polynomial

$$\beta(s) = \sum_{i=0} \beta_i s^i$$

are located in the negative half of the complex plane (such polynomials are sometimes called "Hurwitz" polynomials). The orders of the two sides are assumed to satisfy $n^* = n - m > 0$ and n^* is called the "relative degree" of the plant. The above equation can be expressed in state space form as

$$\begin{cases} \dot{x}_p = A_p x + b_p u \\ y_p = h_p^T x_p \end{cases} . \tag{2.11}$$

The objective is to compute u such that $|y_p - y_m|$ is within a prescribed tolerance range, where y_m is the output of reference model with exactly the same relative degree $n^* = n - m$:

$$\begin{cases} \dot{x}_m = A_m x_m + b_m r \\ y_m = h_m^T x_m \end{cases} . \tag{2.12}$$

The controller structure is described by [11, 15, 24]

$$\begin{cases} \dot{\omega}_1 = F\omega_1 + gu \\ \dot{\omega}_2 = F\omega_2 + gy_p \\ u = \theta^T \omega + r + \theta_g \end{cases} \tag{2.13}$$

where $\omega^T = [y_p, \omega_1^T, \omega_2^T]$, ω_1 and ω_2 are $(n-1)$-dimensional vectors; F is an arbitrary asymptotically stable matrix; (F, g) is controllable; and $\theta^T = [\theta_o, \theta_1^T, \theta_2^T]$

is the controller parameter vector. There exists a set of "ideal" parameters θ_0^*, θ_1^* and θ_2^* such that the dynamics of (2.11) and (2.13) can be combined as

$$\dot{z}_p = A_* z_p + b_*(\phi^T \omega + r), \qquad y_p = h_*^T z_p \tag{2.14}$$

where $z_p^T = [x_p^T, \omega^T]$, $\phi = \theta - \theta^*$ and

$$h_*^T(sI - A_*)^{-1} b_* = h_m^T(sI - A_m)^{-1} b_m = W_m(s). \tag{2.15}$$

When $\theta = \theta^*$, the plant plus the controller will match the reference model as (2.15) indicates. The controller in this case is a MRC. The following example demonstrates how a MRC works.

Example 2.1 *Consider a second-order LTI plant described by*

$$(s^2 + a_1 s + a_0) y_p = (s + b_0) u \tag{2.16}$$

where a_0, a_1 and b_0 are unknown constant coefficients; $s + b_0$ is a Hurwitz polynomial. The relative degree of this plant is $n^* = 1$ while the objective is to compute u such that $y_p = y_m = (s + a_m)^{-1} r$. Note that the reference model has the same relative degree of 1.

The MRC is given as

$$\begin{aligned} u &= \theta_1^* \omega_1 + \theta_2^* \omega_2 + \theta_0^* y_p + r \\ &= \theta_1^*(s+1)^{-1} u + \theta_2^*(s+1)^{-1} y_p + \theta_0^* y_p + r. \end{aligned}$$

Suppose all initial conditions are set to zero, then the above expression means

$$(s + 1 - \theta_1^*)\omega_1 = \theta_0^* y_p + \theta_2^* \omega_2 + r.$$

If one chooses $\theta_1^* = 1 - b_0$, then $(s + 1 - \theta_1^*) = (s + b_0)$ and

$$\omega_1 = (s + b_0)^{-1}(\theta_0^* y_p + \theta_2^* \omega_2 + r).$$

By substituting $(s^2 + a_1 s + a_0) y_p = (s + b_0)(s + 1)\omega_1$ into the above equation, one obtains

$$\begin{aligned} (s^2 + a_1 s + a_0) y_p &= (s + b_0)(s + 1)(s + b_0)^{-1}(\theta_0^* y_p + \theta_2^* \omega_2 + r) \\ &= [(s+1)\theta_0^* + \theta_2^*] y_p + (s + 1) r. \end{aligned} \tag{2.17}$$

The MRC coefficients are determined by

$$\theta_0^* = a_1 - a_m - 1, \quad \theta_1^* = 1 - b_0 \quad \textit{and} \quad \theta_2^* = a_0 - a_1 + 1.$$

As the result, (2.17) becomes

$$(s+1)(s+a_m)y_p = (s+1)r.$$

The controlled output y_p behaves exactly as the output of the prescribed model system.

For the general case, one can specify the model in state space

$$\dot{z}_m = A_* z_m + b_* r, \qquad y_m = h_*^T z_m.$$

The state error $e = z_p - z_m$ and output error $e_1 = y_p - y_m$ are derived by subtracting the above equations from (2.14)

$$\dot{e} = A_* e + b_* \phi^T \omega, \quad e_1 = h_*^T e. \tag{2.18}$$

When $n^* = 1$, a stable reference model must be strictly positive real (SPR). There exist positive definite matrices $P = P^T$ and $Q = Q^T$ such that

$$A_*^T P + P A_* = -Q \quad \text{and} \quad P b_* = h_*. \tag{2.19}$$

The following adaptive law

$$\dot{\theta} = -e_1 \omega \tag{2.20}$$

is applied to adjust the controller parameter vector θ.

Theorem 2.3 *The closed-loop system (2.18) and (2.20) is asymptotically stable and the tracking error e will eventually converge to zero.*

Proof: Consider a Lyapunov function candidate

$$L = e^T P e + \phi^T \phi.$$

Its time derivative along the system trajectory (2.18) has the form of

$$\begin{aligned}\dot{L} &= -e^TQe + 2e^TPb_*\phi^T\omega + 2\phi^T\dot{\phi} \\ &= -e^TQe + 2(e_1\phi^T\omega + \phi^T\dot{\phi}).\end{aligned}$$

where $e_1 = e^TPb_* = e^Th_*$, a part of (2.19), has been substituted. The last two terms of the above equation can be cancelled by substituting (2.20). It then follows that

$$\dot{L} = -e^TQe \leq 0.$$

This proves the theorem. **Q.E.D.**

When $n^* > 1$, the control law is still (2.13) while the adaptation law has to be different. The closed-loop system is described by (2.14). It has an operator expression

$$y_p = W_m(s)(\phi^T\omega + r) \tag{2.21}$$

where $W_m(s)$ is an integral-differential operator with relative degree n^*. It describes the desired dynamics of the model reference as indicated by (2.15). Accordingly, the reference model has an operator expression of

$$y_m = W_m(s)r. \tag{2.22}$$

The closed-loop tracking error equation is the difference of (2.21) and (2.22)

$$e = W_m(s)\phi^T\omega. \tag{2.23}$$

The fact that $W_m(s)$ has a relative degree of more than 1 makes it impossible to find positive definite matrices $P > 0$ and $Q > 0$ such that $PA_m + A_m^TP = -Q$. That is the main reason to modify the adaptation law. The adaptation process relies on a new error signal

$$e_1(t) = [\theta^TW_m(s)I - W_m(s)\theta^T]\omega(t)$$

which adds to the tracking error $e(t)$ to form

$$e_2(t) = e(t) + e_1(t).$$

Now, it is not difficult to verify that $e_2 = \phi^T \xi$ where $\xi = W_m(s)I\omega$. The adaptation law is given by

$$\dot{\theta} = -\frac{e_2 \xi}{1 + \omega^T \omega + \xi^T \xi}. \tag{2.24}$$

Theorem 2.4 *The closed-loop system (2.23) and (2.24) is asymptotically stable.*

The proof of this theorem is rather involved and not included here. However, it can be found in [12] and[15].

2.4 MRAC for LTV and NTV Plants

The ideal parameter θ^* of a model reference controller for a LTI plant can be viewed as a fixed point in a proper dimensional space as Fig. 2.1 indicates; the adaptation laws presented in the above few sections are all pure integrators with the error signal e providing a proper adaptation direction. An important design issue of such adaptive laws is to make sure that the error signal can combine with some proper system feedback to provide a correct adaptation direction. As long as the adaptation direction is correct, the integral operations will guarantee that the control parameter vector θ eventually converges to θ^*.

The stability theorems presented in this chapter are all based on ideal plants with perfect mathematical models. Unfortunately, this is not true in many real applications. Almost every physical system is nonlinear and/or time varying. The LTI model of a plant is obtained by ignoring some dynamic effects which are relatively small. If one uses a simple symbol τ_d to denote the unmodeled dynamics and assumes τ_d to be uniformly bounded instead of being zero, then all the above reviewed theorems are not necessarily true because the Lyapunov functions used in the proofs will no longer be negative definite. In fact, Ioannou and Kokotovic demonstrated [14] through examples that adaptation laws based on pure integrations may be unstable in the presence of certain kinds of disturbances. They proposed a so called "σ-modification" to improve the robustness of adaptive controllers. Recently, an $|e|$-modified adaptive law was proposed by Narerndra and Annaswamy [15] which demonstrated better performance and robustness.

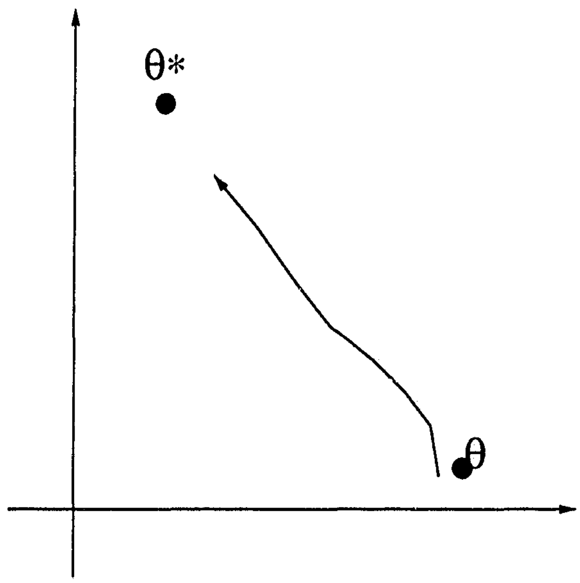

Figure 2.1: Convergence of parameter adaptation.

In order to minimize the effect of unmodeled dynamics, more precise models such as LTV or NTV plants are needed. Typical contributions of adaptive control for LTV and/or NTV plants were made by Middleton and Goodwin [20], Rotea and Khargonekar [22] as well as Tsakalis and Ioannou [24][25].

Many researchers [18]–[23] restrict their results to slowly varying LTV and/or NTV plants. Middleton and Goodwin assume that the plant parameters belong to a convex set [20]. Tsakalis and Ioannou [24] [25] separate MRC from MRAC for LTV plants. Their new MRC designs are asymptotically stable for any fast varying LTV plants as long as the plant terms are exactly known. Their MRAC schemes require some *à priori* knowledge about the plant terms. The corresponding adaptation law is able to adjust the MRC coefficients to some slowly varying coefficients of the plant terms, while the possible fast varying effect must be known *à priori* as functions of time. The MRAC designs presented in [24, 25] are the best results available yet for LTV plants. In Chapters 5 and 6, two new adaptive controller designs will be presented as an improvement to Tsakalis and Ioannou's schemes.

The stability analysis of adaptive controllers for LTV and/or NTV plants are much more complicated than those for LTI plants. For this reason, the analytical derivations are not reviewed here. Those who are interested in this subject are referred to [18]–[25] for details.

2.5 Summary

This chapter reviews the techniques of model reference adaptive control for LTI plants. The basic idea of MRAC can be best understood for first order systems where both the plant and the reference model satisfy first order linear differential equation. The corresponding Lyapunov analysis is easy to understand. Applications of MRAC to other LTI plants are extensions of the basic idea developed for first order systems. Adaptive control of LTV and NTV plants are much more complicated topics. Most of the available results requires the knowledge of "structural uncertainties" to design adaptive controllers. The sources of these reports are given in the last section for ease of reference.

Chapter 3

Manipulator Dynamics

The dynamic model of a robotic manipulator is closely related to the design of its controller. Starting from the late 1950s and early 1960s, a number of researchers made significant contributions to this problem. Based on Uicker's work [46] on linkages, Kahn and Roth [47] studied the particular problem of a multidegree-of-freedom mechanical manipulator; Renaud [48] and Liegois *et al.* [49] investigated the formulation of the mass-distribution descriptions of the links; Stepanenko [50] was the first to use a "Newton-Euler" approach to dynamics instead of the somewhat traditional Lagrangian approach. His work was improved in efficiency by Orin *et al.* [51]. It was discovered that the computation of dynamics can be simplified by some recursive formulations. Armstrong [52], and Luh, Walker and Paul [6] further contributed to the problem and reported an algorithm whose computational count is proportional to the number of links. The computational efficiency was further improved by Hollerbach [53], Silver [54], Hollerbach and Sahar [55] and many others [56]-[58].

This chapter reviews the currently available techniques for the computation of a manipulator dynamic model. The robot is assumed to consist of an open-chain with articulated rigid links. The elastic effects are assumed to be negligible. The main focus is how to compute the coefficient matrices $M(q)$, $V(\dot{q}, q)$ and the gravity force $G(q)$ given that the feedback information $\dot{q}$, q and the inertia parameters such as mass, center of mass and inertia of each link are available.

Although (1.1) looks rather simple, it is extremely difficult to obtain the analytical solutions of $M(q)$ $V(\dot{q}, q)$ and $G(q)$ for a general n-link robot. Instead, some numerical algorithms are developed. These algorithms belong to two main groups. One is the Euler–Lagrange method and the other the Newton-Euler method.

3.1 The Euler–Lagrange Equation

In classical mechanics, the Euler–Lagrange equation is an efficient way to derive the dynamics of a system of moving rigid bodies subject to certain constraints. The key step is to find the system Lagrangian function $L = K - W$ where K and W denote, respectively, the kinetic and potential energy of a system. Then the system dynamics will be governed by the following general equation

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right) - \frac{\partial L}{\partial q} = \tau$$

where $q \in R^n$ is a vector of generalized coordinates and $\dot{q}$ is the time derivative of q. The derivation of the Euler–Lagrange equation can be found in Griffiths [65].

A robotic manipulator is an open-chain articulated mechanical structure. Its links can be labeled along the chain structure in a consecutive order. The base of the robot is usually called the 0-th link. The connection between two neighboring links is called a "joint" which is either revolute or prismatic. It is a convention [63, 64] to attach a coordinate frame to each link (called the link frame). The z_i axis of the i-th link frame is assigned to the direction of motion of the i-th link. In other words, the i-th link either rotates or translates along the z_i axis of the i-th link frame. A schematic of the i-th joint frame is plotted in Fig. 3.1 to illustrate the frame system of a robot. In Fig. 3.1, the three coordinates of the base frame are denoted as x_0, y_0 and z_0 respectively. Only the $(i-1)$-th and i-th links are plotted. They are connected by a rotational joint to which the i-th joint frame is assigned. The coordinates of the i-th frame are denoted as x_i, y_i and z_i respectively. The z_i vector is not plotted explicitly because it is perpendicular to the picture.

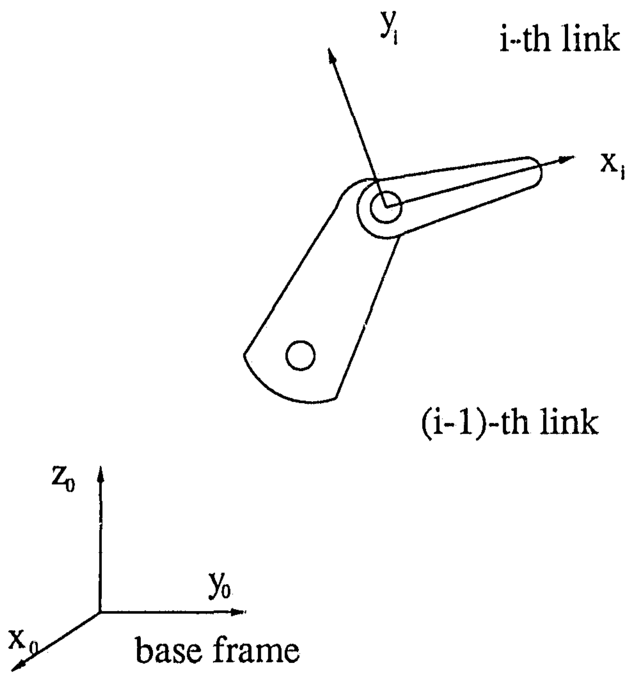

Figure 3.1: The i-th joint frame of a robot.

A vector can be expressed in any one of the link frames. Let P_{ik} denote the vector from the i-th link frame origin to the k-th link frame origin; then a vector ${}^k v$ expressed in the k-th link frame can be also expressed in the i-th link frame as

$$ {}^i v = {}^i_k R \, {}^k v + {}^i P_{ik} \tag{3.1} $$

where the leading superscript "i" indicates the particular frame in which the corresponding vector is expressed. A vector without the leading superscript can be assumed to be expressed in the base frame by default.

Each link can be viewed as a lump mass. When the robot moves, the i-th link is associated with a vector v_{c_i} representing the linear velocity of the i-th link mass center. Also associated with the i-th link is ω_i, the angular velocity of the i-th link. If the joint angles (displacements for prismatic joints) are specified as a vector of generalized coordinates $q = [q_1, \ldots, q_n]^T$, then v_{c_i} and ω_i are related to

q by

$$\begin{bmatrix} v_{c_i} \\ \omega_i \end{bmatrix} = J_{c_i}(q)\dot{q} \qquad J_{c_i}(q) \in R^{6\times n}$$

where $J_{c_i}(q)$ is the Jacobian matrix with respect to the mass center of the i-th link. For a robot with all revolute joints, the Jacobian matrix with respect to the mass center of the k-th link is given by

$$J_{c_k}(q) = \begin{bmatrix} z_1 \times (P_{1k} + c_k) & z_2 \times (P_{2k} + c_k) & \dots & z_k \times c_k & 0 & \dots & 0 \\ z_1 & z_2 & \dots & z_k & 0 & \dots & 0 \end{bmatrix}$$

where z_i is a unit vector representing the z_i axis (in the base frame); c_k is a vector representing the k-th link mass center. The $(k+1)$-th and higher ordered columns are zero because of the open-chain mechanical structure; the generalized velocities of higher joints do not affect the linear and angular velocities of lower links. In case the i-th joint of a robot is translational, the i-th column of the above Jacobian matrix will be changed from

$$\begin{bmatrix} z_i \times (P_{ik} + c_k) \\ z_i \end{bmatrix} \quad \text{to} \quad \begin{bmatrix} z_i \\ 0 \end{bmatrix}.$$

Let m_i and T_i denote, respectively, the mass and inertia of the i-th link, then the kinetic energy of the whole system can be written as

$$K = \frac{1}{2}\sum_{i=1}^{N}[v_{c_i}^T, \omega_i^T]Q_i \begin{bmatrix} v_{c_i} \\ \omega_i \end{bmatrix}, \qquad Q_i = \begin{bmatrix} m_i I & 0 \\ 0 & T_i \end{bmatrix}$$

where v_{c_i} is the linear velocity of the i-th link mass center; ω_i is the angular velocity of the i-th link body; m_i and T_i are, respectively, the mass and inertia tensor of the i-th link. Alternatively, the system kinatic energy can be expressed in the form of

$$K = \frac{1}{2}\dot{q}^T[\sum_{i=1}^{N} J_i^T(q)Q_iJ_i(q)]\dot{q} = \frac{1}{2}\dot{q}^T M(q)\dot{q}$$

where

$$M(q) = \sum_{i=1}^{n} J_{c_i}^T Q_i J_{c_i}$$

is the inertia matrix of the overall system. It is uniformly symmetric and positive definite. The potential energy of the overall system can also be expressed in the generalized coordinate space as $W = \int G^T(q)dq$ where $G(q) \in R^n$ represents the gravitational force (or equivalent torque) acting on the joints. The Lagrangian function of the system is given by $L = K - W = K - \int G(q)dq$. Applying the Euler–Lagrange principle, one can obtain

$$\begin{aligned} \tau &= \frac{d}{dt}\frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = \frac{d}{dt}\nabla_{\dot{q}} L - \nabla_q L \\ &= M(q)\ddot{q} + \frac{1}{2}\{\dot{M}(q)\dot{q} - \nabla_q[\dot{q}^T M(q)\dot{q}]\} + G(q) \end{aligned} \tag{3.2}$$

where ∇_q denotes the gradient with respect to q. By taking into account the unmodeled dynamics $\tau_d \in R^n$, one can write the dynamics of such a system as

$$M(q)\ddot{q} + V(\dot{q}, q)\dot{q} + G(q) + \tau_d = \tau \tag{3.3}$$

where the $V(\dot{q}, q) \in R^{n \times n}$ matrix is not unique. There are multiple forms of $V(\dot{q}, q)$ such that

$$V(\dot{q}, q)\dot{q} = \frac{1}{2}\{\dot{M}(q)\dot{q} - \nabla_q[\dot{q}^T M(q)\dot{q}]\}.$$

One possible way to determine $V(\dot{q}, q)$ is to express the kinetic energy as a quadratic form of the vector $\dot{q}$:

$$K = \frac{1}{2}\sum_{i,j}^{n} m_{ij}(q)\dot{q}_i\dot{q}_j = \frac{1}{2}\dot{q}^T M(q)\dot{q}$$

where m_{ij} is the i, jth entry of the $n \times n$ inertia matrix $M(q)$. The potential energy $W(q) = \int G(q)dq$ is independent of $\dot{q}$. Applying the Euler-Lagrange equations to the system Lagrangian

$$L = K - W = \frac{1}{2}\sum_{i,j} m_{ij}(q)\dot{q}_i\dot{q}_j - W(q), \tag{3.4}$$

one can write

$$\frac{\partial L}{\partial \dot{q}_k} = \sum_{j=1}^{n} m_{kj}(q)\dot{q}_j,$$

and

$$\begin{aligned}\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_k} &= \sum_{j=1}^{n} m_{kj}(q)\ddot{q}_j + \sum_{j=1}^{n} \frac{d}{dt} m_{kj}(q)\dot{q}_j \\ &= \sum_{j=1}^{n} m_{kj}(q)\ddot{q}_j + \sum_{i,j}^{n} \frac{\partial m_{kj}}{\partial q_j}\dot{q}_i\dot{q}_j.\end{aligned}$$

It follows (3.4) that

$$\frac{\partial L}{\partial q_k} = \frac{1}{2}\sum_{i,j}\frac{\partial m_{ij}}{\partial q_k}\dot{q}_i\dot{q}_j - \frac{\partial W}{\partial q_k}.$$

Thus the Euler-Lagrange equations can be written as

$$\sum_{j=1}^{n} m_{kj}\ddot{q}_j + \sum_{i,j}\{\frac{\partial m_{kj}}{\partial q_i} - \frac{1}{2}\frac{\partial m_{ij}}{\partial q_k}\}\dot{q}_i\dot{q}_j - \frac{\partial W}{\partial q_k} = \tau_k.$$

The order of summation can be inter-changed to take advantage of symmetry and obtain

$$\sum_{i,j}\frac{\partial m_{kj}}{\partial q_i}\dot{q}_i\dot{q}_j = \frac{1}{2}\sum_{i,j}\{\frac{\partial m_{kj}}{\partial q_i} + \frac{\partial m_{ki}}{\partial q_j}\}\dot{q}_i\dot{q}_j.$$

Hence

$$\sum_{i,j}\{\frac{\partial m_{kj}}{\partial q_i} - \frac{1}{2}\frac{\partial m_{ij}}{\partial q_k}\}\dot{q}_i\dot{q}_j = \frac{1}{2}\sum_{i,j}\{\frac{\partial m_{kj}}{\partial q_i} + \frac{\partial m_{ki}}{\partial q_j} - \frac{\partial m_{ij}}{\partial q_k}\}\dot{q}_i\dot{q}_j.$$

Introduce nonlinear scalar functions

$$v_{ijk} = \frac{1}{2}\{\frac{\partial m_{kj}}{\partial q_i} + \frac{\partial m_{ki}}{\partial q_j} - \frac{\partial m_{ij}}{\partial q_k}\}.$$

They represent the Christoffel symbols. Clearly, $v_{ijk} = v_{jik}$. This observation leads to a 50 percent computation reduction when evaluating the system dynamics. Since $G_k = \frac{\partial W}{\partial q_k}$ is the kth element of vector function $G(q)$, one can express the Euler–Lagrange equations as

$$\sum_{j=1}^{n} m_{kj}(q)\ddot{q}_j + \sum_{i,j} v_{ijk}\dot{q}_i\dot{q}_j + G_k(q) = \tau_k, \qquad 1 \le k \le n.$$

The matrix expression of the above equation is exactly (1.1) and (3.3) except that the unmodeled dynamics τ_d is missing. The k, jth element of matrix $V(\dot{q}, q)$ is determined by

$$\begin{aligned} v_{kj} &= \sum_{i=1}^{n} v_{ijk}(q)\dot{q}_i \\ &= \frac{1}{2}\sum_{i=1}^{n}\{\frac{\partial m_{kj}}{\partial q_i} + \frac{\partial m_{ki}}{\partial q_j} - \frac{\partial m_{ij}}{\partial q_k}\}\dot{q}_i. \end{aligned}$$

The Euler-Lagrange equation applies to general mechanical systems. It provides an easy way to represent the usually complicated form of dynamic equation for a mechanical syste. ` which consists of multiple moving bodies.

3.2 The Newton–Euler Algorithm

Another method to compute the system dynamics for an open-chain articulated mechanical structure is the so-called Newton–Euler algorithm. This method adds up the dynamics of individual links. For an n-link open-chain articulated structure, the overall dynamics can be expressed as a sum of

$$\tau = \begin{bmatrix} \tau_{11} \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \begin{bmatrix} \tau_{21} \\ \tau_{22} \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \ldots + \begin{bmatrix} \tau_{n1} \\ \tau_{n2} \\ \tau_{n3} \\ \vdots \\ \tau_{nn} \end{bmatrix} = \sum_{i=1}^{n} \tau_i \tag{3.5}$$

where $\tau_i = [\tau_{i1}, \ldots, \tau_{ii}, 0, \ldots, 0]^T$ is the dynamic of the i-th link; it only affects the i-th joint and the lower joints. In this section, the Newton–Euler algorithm is explained for a robot with rotational joints. The algorithm can be easily generalized to robots with translational joints.

According to (3.5), $\tau(k)$, the generalized torque applied to the k-th joint must satisfy the following equation:

$$\tau(k) = \sum_{i=k}^{n} \tau_{ki} = \sum_{i=k}^{n} z_k[T_i\dot{\omega}_i + \omega_i \times T_i\omega_i + m_i h_{ki} \times (\dot{v}_{c_i} + g)] \tag{3.6}$$

where z_k is an unit vector denoting the k-th joint axis; m_i denotes the mass of the i-th link; $h_{ki} = P_{ki} + c_i$ is a vector from the k-th joint frame origin to the mass-center of the i-th link; g is the gravity vector with $||g|| = 9.8$ and always pointing towards the earth center; v_{c_i} the linear velocity of the mass-center of the i-th link; and ω_i the angular velocity of the i-th link.

Physically, $\frac{d}{dt}Q_i\omega_i = T_i\dot{\omega}_i + \omega_i \times T_i\omega_i$ represents the inertia torque required for the angular movement of the i-th link while $m_i(\dot{v}_{c_i} + g)$ equals the force required for the linear movement of the i-th link. Consequently $m_i h_{ki} \times (\dot{v}_{c_i} + g)$ is the equivalent torque acting on the k-th joint.

Equation (3.6) looks very concise, yet it hides a lot of additional computations required to compute z_k, $\dot{v}_{c_i}$, ω_i, T_i and h_{ki}. But all the additional computations can be implemented by a recursive algorithm to take advantage of the articulated configurations. For example, the joint axes $\{z_k\}_{k=1}^{n}$ are related by a chain of rotation matrices ${}^{k+1}_{k}R$, or simply $z_{k+1} = {}^{k+1}_{k}Rz_k$. Here ${}^{k+1}_{k}R$ is an orthogonal matrix function of generalized coordinate q_k.

The vector h_{ki} can be breakdown as $h_{ki} = P_{ki} + P_{c_i}$ where P_{ki} points from the origin of k-th joint frame to that of the i-th joint frame while P_{c_i} is the vector of mass-center of the i-th link. Its expression in the i-th joint frame is denoted by ${}^{i}P_{c_i}$ and is a constant vector. The P_{ki} vector can be computed recursively by assembling the links from k-th joint to the i-th one. This process is called the "outward iterations" of a Newton–Euler algorithm.

After all the necessary variables in (3.6) are available, the algorithm will compute τ using (3.6). This process can also be implemented by a similar process called the "inward iterations".

There are many versions of the Newton–Euler algorithms. In this review, the algorithm given in [37] is listed because it is accompanied by a detailed explanation on its notations and the derivations.

3.2.1 The outward iterations

This part computes the velocities and accelerations for all links. In the whole mechanical configuration, the $(i+1)$th link is assembled at the end of ith link. As a result, its velocity and acceleration are affected by the velocities and accelerations of the first i links as well as by the joint velocity $\dot{q}_{i+1}$ which is the $(i+1)$th component of the vector $\dot{q}$.

The "propagation" effect of the velocities and accelerations can be efficiently computed by an outward iteration algorithm given by the following equations

$$
\begin{aligned}
{}^{i+1}\omega_{i+1} &= {}^{i+1}_{i}R\,{}^{i}\omega_i + \dot{q}_{i+1}\,{}^{i+1}z_{i+1}, &(3.7)\\
{}^{i+1}\dot{\omega}_{i+1} &= {}^{i+1}_{i}R\,{}^{i}\dot{\omega}_i + {}^{i+1}_{i}R\,{}^{i}\omega_i \times \dot{q}_{i+1}\,{}^{i+1}z_{i+1} + \ddot{q}_{i+1}\,{}^{i+1}z_{i+1}, &(3.8)\\
{}^{i+1}\dot{v}_{i+1} &= {}^{i+1}_{i}R\,{}^{i}\dot{\omega}_i \times {}^{i}P_{i+1} + {}^{i+1}_{i}R\,{}^{i}\omega_i \times ({}^{i}\omega_i \times {}^{i}P_{i+1}) + {}^{i+1}_{i}R\,{}^{i}\dot{v}_i. &(3.9)
\end{aligned}
$$

In the above formulations, we use ${}^{i}\omega_i$, ${}^{i}\dot{\omega}_i$ and ${}^{i}\dot{v}_i$ to denote the angular velocity, angular acceleration and linear acceleration of the ith link. They are all expressed with respect to the attached frame. Matrix ${}^{i+1}_{i}R$ is a rotation that relates the $(i+1)$th frame to the ith frame. Vector P_i points from the origin of the $(i-1)$-th attached frame to that of the i-th attached frame. z_i is a unit vector representing the axis of the ith link.

The velocities and accelerations are then transformed to the corresponding frame attached at the center of mass of each link. The external force ${}^{i}F_i$ and torque ${}^{i}N_i$ needed to support the motion are computed ıo: the ith link with the following formulations

$$
\begin{aligned}
{}^{i}\dot{v}_{c_i} &= {}^{i}\dot{\omega}_i \times {}^{i}P_{c_i} + {}^{i}\omega_i \times ({}^{i}\omega_i \times {}^{i}P_{c_i}) + {}^{i}\dot{v}_i, &(3.10)\\
{}^{i}F_i &= m_i\,{}^{i}\dot{v}_{c_i}, &(3.11)\\
{}^{i}N_i &= {}^{c_i}T_i\,{}^{i}\dot{\omega}_i + {}^{i}\omega_i \times {}^{c_i}T_i\,{}^{i}\omega_i. &(3.12)
\end{aligned}
$$

3.2.2 The inward iterations

The interaction between the links is computed by an inward iteration scheme. For a robotic manipulator, the ith link is assembled on the $(i-1)$th link and it supports the $(i+1)$th link. Such a relationship can be expressed mathematically as

$$
\begin{aligned}
{}^{i}f_i &= {}^{i}_{i+1}R\,{}^{i+1}f_{i+1} + {}^{i}F_i, && (3.13)\\
{}^{i}n_i &= {}^{i}N_i + {}^{i}_{i+1}R\,{}^{i+1}n_{i+1} + {}^{i}P_{c_i} \times {}^{i}F_i \\
&\quad + {}^{i}P_{i+1} \times {}^{i}_{i+1}R\,{}^{i+1}f_{i+1}, && (3.14)\\
\tau_i &= {}^{i}n_i^T\,{}^{i}z_i. && (3.15)
\end{aligned}
$$

In the above formulas, the gravity effect is absent. But it can be included quite simply by letting ${}^{0}\dot{v}_0 = g$, where g represents the gravity acceleration vector. This is equivalent to an imaginary upward acceleration of the base frame. The fictitious upward acceleration causes exactly the same effect on the whole system as the real gravity force does. Therefore no extra computation is needed.

3.3 Summary

The Euler–Lagrange and Newton–Euler methods for evaluating the dynamics of a robotic manipulator are reviewed in this chapter. It must be emphasized that the Euler–Lagrange and Newton–Euler methods are equivalent as far as the final result is concerned. However, the Euler-Lagrange equation can be applied to practically any mechanical systems as long as their Lagrangian functions can be derived. When the Newton–Euler algorithm is applied to a closed-chain configuration, many operations, such as the computation of linear velocities and accelerations, must be modified. Particularly, ${}^{k+1}_{k}R$ is not necessarily a matrix function of a single variable q_k. Such differences are due to the reduction of degree of freedom because of the closed-chain configurations.

The evaluation of robot dynamics is closely associated with the current trend of model-based or "parameter linearization" adaptive controllers [37] [39]. In comparing the two algorithms, more attention will be paid to the possible development of general algorithms for parameter linearization.

In the development of Newton–Euler algorithms, the focus bas been on reducing computations as much as possible. The $M(q)$, $V(\dot{q}, q)$ and $G(q)$ matrices in the system dynamics do not appear explicitly. It is rather difficult to determine which particular form of $V(\dot{q}, q)$ will be used to compute $V(\dot{q}, q)x$. In Chapter 6, it will be shown that only a proper form of $V(\dot{q}, q)$ satisfies the identity

$$x^T[\dot{M}(q) - 2V(\dot{q}, q)]x = 0 \quad \forall \quad x \neq 0. \tag{3.16}$$

Thus, the Newton–Euler algorithms do not seem to fit Slotine and Li's adaptive controller [39] which depends on (3.16) for its stability proof. On the other hand, the Newton–Euler algorithms use linear velocities and angular velocities of individual links as intermediate variables in the recursive computation. These variables are needed to linearize (3.6) with respect to the unknown parameters, such as mass, center of mass and inertia tensor of the last link. From this point of view, the Newton–Euler algorithm seems to have a higher potential to be modified into a general parameter linearization algorithm. This problem will be addressed in Chapter 7.

The Euler–Lagrange method can be applied only when the overall system inertia matrix $M(q)$ is available. The unknown parameters are hidden in the $M(q)$ matrix and it seems rather difficult to linearize $M(q)$ with respect to the unknown parameters. However, the Euler–Lagrange equation evaluates matrices $V(\dot{q}, q)$ and $G(q)$ and especially determines the suitable $V(\dot{q}, q)$ such that (3.16) holds. It seems that the Slotine and Li's controller [39] needs both algorithms to implement an adaptive controller for a general n-link robot. This fact motivates the design of a self-tuning regulator to be presented in Chapter 7.

Chapter 4

Manipulator Controllers

Numerous controller designs have been proposed for robotic manipulators. It is only possible to review some of the most representative designs in this chapter. The currently available controllers for robotic manipulators can be put into three main groups: the simple feed-back controllers; the fixed nonlinear compensators and the adaptive controllers.

The first group includes combinations of position (P), differential (D) and integral (I) feed-back controllers such as PD and PID controllers. The stability of feed-back controllers is well studied for LTI plants. For nonlinear or time-varying plants, the stability analysis becomes difficult and no general results are available. However, feed-back controllers, especially PD controllers, are very popular in industrial applications despite extensive research effort on other sophisticated controllers. The reason is simple: feed-back controllers are simple and economic. Particularly, it has been proved by Arimoto *et al.* [69] that a PD controller with exact gravity compensation is asymptotically stable for point-to-point control. In [37], a conjecture is stated that a PD controller, without gravity compensation, is stable for trajectory tracking if the velocity (differential) feed-back gain is sufficiently large. This conjecture is supported by many experimental and simulated results.

Although PD controllers are believed to be stable in trajectory tracking, their tracking errors are relatively large because of the nonlinear dynamics of a robotic manipulator. The only way to realize zero tracking error without infinitely high gain feed-back is to compensate the nonlinear dynamics. The computed-torque controller is a successful example of the second group (the fixed nonlinear compensators). It computes (according to the feed-back information) and compensates the nonlinear dynamics. If the robot parameters are exactly available, the system dynamics will reduce to a simple linear second order differential equation. The analysis of tracking performance becomes straightforward.

The performance of computed torque controllers is uncertain when the robot parameters are inaccurate. The robot dynamics computed using the inaccurate parameters can not compensate the real system dynamics exactly. The remaining dynamics still effect the stability of the closed-loop system. Different methods have been reported in order to deal with the parameter uncertainty. These methods form the third group, the adaptive controllers. One of them is the variable-structure-control technique. As will soon be discussed in detail, this method estimates the instant signs of the unknown dynamics and uses a sufficiently large switching signal to compensate them. In order for the tracking error to converge to zero, the ideal actuator must be able to switch with an infinitely high frequency. A modified version uses a linear area to replace the step switching function, as a result, the tracking error is forced into a computable tolerance range. Strictly speaking, a variable-structure controller does not belong to the class of adaptive controllers.

An adaptive controller does not merely estimate and compensate the sign of unknown dynamics. Instead, it attempts to compensate the whole nonlinear dynamics according to the feedback information. The early versions of adaptive controllers are more performance oriented. The robot dynamics are assumed to be LTI and MRAC designs are applied without much modification. Recently, the linear parameterization method has attracted growing attention because of its potential to force zero tracking errors. It is reviewed here in detail as an example of adaptive controllers and as a new trend for further study.

4.1 PD Controller with Gravity Compensation

This is the simplest controller for the robotic manipulators. It guarantees stable pointwise movement when $G(q)$, the gravity force vector, is available. The control law is given by

$$\tau = -K_v\dot{q} - K_p e + G(q) \tag{4.1}$$

where $e = q - q_d$ is the position error between the joint position vector q and the desired position q_d; $K_v = K_v^T$ and $K_p = K_p^T$ are two arbitrary positive definite matrices. Substituting (4.1) into the system dynamic equation (1.1) and neglecting the unmodeled dynamics, one can write

$$M(q)\ddot{q} = -V(\dot{q}, q)\dot{q} - K_v\dot{q} - K_p e. \tag{4.2}$$

Theorem 4.1 *The closed-loop system (4.2) is asymptotically stable and the position error vector e will eventually converge to zero.*

Proof: Consider a Lyapunov function candidate

$$v = \frac{1}{2}\{\dot{q}^T M(q)\dot{q} + e^T K_p e\}.$$

The time derivative of v, evaluated along (4.2), is given by

$$\dot{v} = -\dot{q}^T K_v \dot{q}$$

where we have substituted identity $x^T[\dot{M}(q) - 2V(\dot{q}, q)]x = 0$ and $\dot{e} = \dot{q}$ to arrive at the above equation. It follows that v will keep decreasing as long as $\dot{q} \neq 0$. v will converge to a constant when $\dot{q} = 0$. At that time, (4.2) reduces to $K_p e = 0$. Since K_p is positive definite, the only solution must be $e = 0$. **Q.E.D.**

Remarks: The above stability analysis is valid only when the gravity force $G(q)$ is perfectly cancelled and no unmodeled disturbance is present. In practice, such a perfect matching is almost impossible. The stability of PD controllers without perfect cancellation of $G(q)$ is still a conjecture which remains to be proved.

4.2 Computed Torque Method

The computed torque method is designed for position control along a prescribed trajectory. It is assumed that the dynamic parameters $M(q)$, $V(\dot{q}, q)$ and $G(q)$ are available and the unmodeled dynamics τ_d is neglected. The control law is given by

$$\tau = M(q)(\ddot{q}_d + K_v\dot{e} + K_p e) + V(\dot{q}, q)\dot{q} + G(q) \tag{4.3}$$

where q_d is the desired trajectory and $e = q_d - q$ the tracking error.

Substituting (4.3) into (1.1) and neglecting τ_d, one obtains

$$M(q)(\ddot{e} + K_v\dot{e} + K_p e) = 0.$$

Since $M(q)$ is uniformly positive definite, the above equation means

$$\ddot{e} + K_v\dot{e} + K_p e = 0.$$

When the two gain matrices K_v and K_p are chosen to be diagonal, the nonlinear dynamics of a manipulator become decoupled and the tracking error will be subjected to a second order stable linear equation as the above equation suggests.

Remarks: Also requires perfect matching of the nonlinear functions $M(q)$, $V(\dot{q}, q)$ and $G(q)$. But the controller is able to tolerate any bounded disturbances due to unmodeled dynamics when perfect knowledge of the system parameters are available.

4.3 Variable Structure Control

An important feature of the computed torque method is its reliance on the exact knowledge about the system dynamics $M(q)$, $V(\dot{q}, q)$ and $G(q)$. The stability analysis relies solely on an assumption that these parameters are known exactly to the controller. As we discussed in the introduction, such an assumption is rarely true in practice. Many times, some inaccurate quantities $\hat{M}(q)$, $\hat{V}(\dot{q}, q)$ and $\hat{G}(q)$ are computed by the controller algorithm because of the imprecise knowledge about the robot parameters such as mass, center of mass and inertia for each link. As the result, (4.3) becomes

$$\tau = \hat{M}(q)(\ddot{q}_d + K_v\dot{e} + K_p e) + \hat{V}(\dot{q}, q)\dot{q} + \hat{G}(q).$$

Substituting into (1.1) and neglecting τ_d, one obtains

$$\hat{M}(\ddot{e} + K_v\dot{e} + K_p e) = \Delta M\ddot{q} + \Delta V\dot{q} + \Delta G$$

where $\Delta M = M(q) - \hat{M}(q)$, $\Delta V = V(\dot{q}, q) - \hat{V}(\dot{q}, q)$ and $\Delta G = G(q) - \hat{G}(q)$. The right side of the above equation involves the products of $\ddot{q}$ and $\dot{q}$ which represent some nonlinear time-varying feedback effects. It is difficult to predict if the feedback is positive or negative. Thus the stability of the closed-loop system is uncertain. In order to ensure stable tracking, the variable structure control is used. The control law is given by

$$\begin{aligned} \tau &= \hat{M}(\ddot{q}_d + K\epsilon + u) + \hat{V}\dot{q} + \hat{G} \\ &= M(\ddot{q}_d + K\epsilon + u) - \Delta M(\ddot{q}_d + K\epsilon + u) + \hat{V}\dot{q} + \hat{G} \end{aligned} \tag{4.4}$$

where $K = [K_v, K_p]$ and $\epsilon = [\dot{e}, e]^T$; u is an additional control input to be determined a bit later.

Substituting (4.4) into (1.1) and neglecting the effect of τ_d, we can write

$$\ddot{e} + K_v\dot{e} + K_p e = \eta - u \tag{4.5}$$

where

$$\eta = Hu + H\ddot{q}_d + HK\epsilon + M^{-1}(\Delta V\dot{q} + \Delta G), \tag{4.6}$$

and $H = M^{-1}\Delta M = I - M^{-1}\hat{M}$.

Spong and Vidyasagar [64] proposed a variable structure control law for the closed-loop system of (4.5). Their stability analysis is based on the following assumption:

$$\sup_{t\geq 0} \|\ddot{q}_d\| < c_1 < \infty. \tag{4.7}$$

$$\|H\| = \|I - M^{-1}\hat{M}\| \leq \alpha < 1 \qquad \forall q \in R^n. \tag{4.8}$$

$$\|M^{-1}(\Delta V\dot{q} + \Delta G)\| \leq \phi(\epsilon, t) \tag{4.9}$$

for some known function ϕ bounded in t.

The assumption of (4.7) comes from the fact that the desired acceleration should be assigned a bounded function of time. (4.8) seems to be rather restrictive. However, it is always true by some simple choice of $\hat{M}$. Since $M(q)$ is uniformly positive definite for all $q \in R^n$, there must exist positive constants $\lambda_{M1} < \lambda_{M2}$ such that

$$\lambda_{M1} \leq \|M(q)\| \leq \lambda_{M2} \quad \forall q \in R^n.$$

By choosing

$$\hat{M} = \frac{\lambda_{M1} + \lambda_{M2}}{2} I$$

it can be shown that

$$\|I - M^{-1}\hat{M}\| \leq \frac{\lambda_{M2} - \lambda_{M1}}{\lambda_{M2} + \lambda_{M1}} = \alpha < 1.$$

In practice, one could find other ways to compute $\hat{M}(q)$ such that α is as small as possible. In any case, the simple procedure provides an easy way to satisfy (4.8). As for (4.9), the $\phi(\epsilon, t)$ function can be a polynomial $a_0 + a_1\|\epsilon\| + a_2\|\epsilon\|^2$ as explained in [64].

Now, we try to find a scalar function $\rho(\epsilon, t)$ such that

$$\|u\| \leq \rho(\epsilon, t) \quad \text{and} \quad \|\eta\| \leq \rho(\epsilon, t). \tag{4.10}$$

Substituting (4.7)-(4.9) into (4.6), we can write

$$\begin{aligned} \|\eta\| &\leq \alpha\|u\| + \alpha c_1 + \alpha\|K\|\|\epsilon\| + \phi(\epsilon, t) \\ &\leq \alpha\rho(\epsilon, t) + \alpha c_1 + \alpha\|K\|\|\epsilon\| + \phi(\epsilon, t) \\ &= \rho(\epsilon, t) \end{aligned}$$

This enables us to write

$$\rho(\epsilon, t) = \frac{1}{1 - \alpha}[\alpha c_1 + \alpha\|K\|\|\epsilon\| + \phi(\epsilon, t)]$$

which will satisfy both inequalities of (4.10).

Let

$$A = \begin{bmatrix} -K_v & -K_p \\ I & 0 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} I \\ 0 \end{bmatrix}$$

then (4.5) can be expressed as

$$\dot{\epsilon} = A\epsilon + B(\eta - u). \tag{4.11}$$

By a proper choice of $K = [K_v, K_p]$, one can find positive definite matrices $P = P^T$ and $Q = Q^T$ such that

$$PA + A^T P = -Q.$$

Consider a Lyapunov function candidate $v = \epsilon^T P \epsilon$, the time derivative of v evaluated along (4.11) is given by

$$\begin{aligned} \dot{v} &= -\epsilon Q \epsilon + 2\epsilon P B(\eta - u) \\ &\leq -\epsilon^T Q \epsilon + w^T(\eta - u) \end{aligned}$$

where $w = B^T P \epsilon$.

If we specify

$$u = \begin{cases} \rho(\epsilon, t)\frac{w}{\|w\|} & \text{for } w \neq 0 \\ 0 & \text{for } w = 0 \end{cases},$$

then we can write

$$\dot{v} = \begin{cases} -\epsilon^T Q \epsilon + 2[w^T \eta - \|w\|\rho(\epsilon, t)] & \text{for } w \neq 0 \\ -\epsilon^T Q \epsilon & \text{for } w = 0 \end{cases}.$$

According to (4.10), $\dot{v} \leq 0$ in both cases and v will keep on decreasing until $\epsilon = 0$. Therefore the closed-loop system is asymptotically stable.

Remarks: The advantage of this type of controllers is its ability to control robots without perfect parameter match. They are also robust with respect to unmodeled dynamics as long as the disturbances are bounded in norm by the scalar function $\rho(\epsilon, t)$. But the control laws could switch at extremely high frequency because of the additional control u. In many applications the actuators are unable to provide such a high frequency switch of forces.

4.4 Adaptive Control of Manipulators

The stability analysis of the controller discussed in the previous section has a small flaw: the discontinuity of the definition for u. Such a control requires an infinitely high frequency switch which is difficult to implement in practice. In this section, we discuss the use of adaptive control strategy for manipulator control.

4.4.1 MRAC applied to robots

Under the assumption that $M(q)$, $V(\dot{q}, q)$ and $G(q)$ vary "slowly" compared with q (it is not clear how slow the variations are), then the robot dynamics (without the unmodeled dynamics τ_d) can be approximated by a second-order LTI plant

$$M_c\ddot{q} + V_c\dot{q} + G_c = \tau \tag{4.12}$$

where the subscripts "c" indicates that the corresponding matrix can be considered as constant. Denote as $y_p^T = [\dot{q}^T, q^T]$, $u_p = \tau - G_c$,

$$A_p = \begin{bmatrix} M_c^{-1}V_c & 0 \\ -I & 0 \end{bmatrix} \quad \text{and} \quad B_p = \begin{bmatrix} M_c^{-1} \\ 0 \end{bmatrix}.$$

Then (4.12) can be written in a state space form

$$\dot{y}_p = -A_p y_p + B_p u_p. \tag{4.13}$$

For a prescribed reference model system

$$\dot{y}_m = -A_m y_m + B_m r$$

where

$$A_m = \begin{bmatrix} K_v & K_p \\ -I & 0 \end{bmatrix} \quad \text{and} \quad B_m = \begin{bmatrix} I \\ 0 \end{bmatrix},$$

one can easily verify that the "perfect matching" condition [16] is satisfied. That is

$$(I - B_pB_p^\dagger)(A_p - A_m) = 0 \quad \text{and} \quad (I - B_pB_p^\dagger)B_m = 0 \quad \forall \;\; y_p, t$$

where $B_p^{\dagger} = (B_p^T B_p)^{-1} B_p^T$. According to Theorem 2.2, an adaptive controller with control law (2.7) and adaptation law (2.10) will render an asymptotically stable tracking system.

Remarks: The above algorithm belongs to the "performance-based" adaptive controllers [36]. These adaptive controllers are all based on an assumption of "slowly-varying" dynamics of robots. Many simulation and experiment tests show that these controllers are stable. But their theoretical stability analysis is weak because the "slow-varying" assumptions are observed to be violated in some experiments [36].

4.4.2 Linear parameterization

In chapter 3, we discussed several MRAC schemes for LTI plants. The stability analysis of all these MRAC schemes makes use of the important fact that the unknown plant coefficients are constant. For robotic manipulators, the dynamic coefficient matrices and vectors are functions of the joint positions which change as the manipulator moves. A direct application of MRAC to control the manipulators does not guarantee stable tracking for the closed-loop system. One way out of this is by means of linear parameterization.

Neglecting the unmodeled dynamics τ_d, the equation of motion of an n-link manipulator is described by

$$M(q)\ddot{q} + V(\dot{q}, q)\dot{q} + G(q) = N(\ddot{q}, \dot{q}, q, p) \tag{4.14}$$

where the right side represents the torque computed by a computed algorithm. The symbol p represents the effect of system parameters such as mass, center of mass and inertia of each link. By "parameter linearization", it is meant that there exist some algorithms such that (4.14) can be reduced to

$$M(q)\ddot{q} + V(\dot{q}, q)\dot{q} + G(q) = Y(\ddot{q}, \dot{q}, q)p \tag{4.15}$$

where the system dynamics are expressed as the linear product of an known function matrix $Y(\ddot{q}, \dot{q}, q)$ and a generalized parameter vector p whose components are some combinations of mass, center of mass and inertia of each link. For example, the dynamics of an ideal two-link configuration is described by

$$\begin{aligned} \tau_1 &= m_2 l_2^2(\ddot{q}_1 + \ddot{q}_2) + m_2 l_1 l_2 c_2 (2\ddot{q}_1 + \ddot{q}_2) + (m_1 + m_2) l_1^2 \ddot{q}_1 - m_2 l_1 l_2 s_2 \dot{q}_2^2 \\ & \quad -2 m_2 l_1 l_2 s_2 \dot{q}_1 \dot{q}_2 + m_2 l_2 g c_{12} + (m_1 + m_2) l_1 g c_1, \\ \tau_2 &= m_2 l_1 l_2 c_2 \ddot{q}_1 + m_2 l_1 l_2 s_2 \dot{q}_1^2 + m_2 l_2 g c_{12} + m_2 l_2^2 (\ddot{q}_1 + \ddot{q}_2). \end{aligned}$$

By defining $p = [p_1, \ldots, p_5]^T$ where $p_1 = m_2 l_2^2$, $p_2 = m_2 l_1 l_2$, $p_3 = m_2 l_2 g$, $p_4 = (m_1 + m_2) l_1^2$ and $p_5 = (m_1 + m_2) l_1 g$;

$$Y = \begin{bmatrix} (\ddot{q}_1 + \ddot{q}_2) & c_2(2\ddot{q}_1 + \ddot{q}_2) + s_2 \dot{q}_2 (\dot{q}_2 - 2\dot{q}_1) & c_{12} & \ddot{q}_1 & c_1 \\ (\ddot{q}_1 + \ddot{q}_2) & c_2 \ddot{q}_1 + s_2 \dot{q}_1^2 & c_{12} & 0 & 0 \end{bmatrix}$$

We can describe the dynamics by

$$\tau = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = Y(\ddot{q}, \dot{q}, q) p.$$

Linear parameterization enables us to express the parameter uncertainty in terms of an unknown constant parameter vector p. This makes it possible to apply some adaptation law to adjust p and ensure stable tracking for the closed-loop system. Although the $Y(\ddot{q}, \dot{q}, q)$ matrix looks simple for the ideal two-link configuration, it is extremely difficult to compute Y for a general n-link configuration. Nevertheless, adaptive control of manipulators by means of linear parameterization is perhaps the most popular adaptive controller in this area. It has the most rigorous stability analysis.

4.4.3 Adaptive control by linear parameterization

The most recent result on adaptive control of robotic manipulators is due to Slotine and Li [39]. In designing the adaptive control system, they do not try to linearize the closed-loop system. Instead, they make use of the Skew-symmetric property of $\dot{M} - 2V$ to cancel out the nonlinear effect. As a result, a globally stable adaptive controller is proposed.

The control law is given by

$$\begin{aligned} \tau &= -K\omega + \hat{M}(q)\dot{\psi} + \hat{V}(\dot{q},q)\psi + \hat{G}(q) \\ &= -K\omega + Y(\dot{\psi},\psi,\dot{q},q)\hat{P} \end{aligned} \tag{4.16}$$

where $\omega = \dot{e} + \Lambda e$, $\psi = \dot{q} - \omega = \dot{q}_d - \Lambda e$ and Λ is an arbitrary positive definite matrix. The objective of the control system is to force $\omega \to 0$ in a finite time. As long as $\omega = 0$, it follows that $\dot{e} + \Lambda e = 0$ and the tracking error should also converge to zero.

The adaptive law is described by

$$\dot{\hat{p}} = -Y^T(\dot{\psi},\psi,\dot{q},q)\omega. \tag{4.17}$$

The closed-loop system is described by

$$M(q)\dot{\omega} + V(\dot{q},q)\omega = -K\omega + Y(\dot{\psi},\psi,\dot{q},q)\phi_p \tag{4.18}$$

where $\phi_p = \hat{p} - p$ represents the parameter error.

Consider a Lyapunov function candidate $L = \frac{1}{2}(\omega^T M(q)\omega + \phi_p^T\phi_p)$. The time derivative of L evaluated along (4.18) is given by

$$\dot{L} = -\omega^T K\omega + [\omega^T Y(\dot{\psi},\psi,\dot{q},q) + \dot{\phi}_p^T]\phi_p$$

where the Skew-symmetric property $x^T(\dot{M} - 2V)x = 0 \; \forall \; x \neq 0$ has been substituted to eliminate the effect of $\dot{M}$ and $V(\dot{q},q)$. A further substitution of (4.17) into the above expression leads to

$$\dot{L} = -\omega^T K\omega \leq 0$$

which implies that the closed-loop system is asymptotically stable.

4.5 Summary

This chapter reviews a number of controllers designed for robotic manipulators. These controllers represent the numerous controllers reported by the previous researchers. The most simple controller is the PD controller. It is proved to be stable for point-to-point control under the condition that the gravitational force is exactly available to the controller and the external disturbances are negligible. The computed-torque controller use forward dynamics to compensate the nonlinear robot dynamics and PD feedback to keep good tracking. It transforms the closed-loop system into a LTI plant if the forward dynamics matches the robot dynamics exactly. However, perfect matching is generally not possible due to various practical reasons. In the most common case, the robot dynamics changes when it operates with an unknown payload. The most simple way to deal with the inaccurate matching effects is by means of variable structure control. Such controllers require that the actuator be able to switch at an infinitely high frequency to guarantee stable tracking. Adaptive controllers adjust the control gains automatically according to the feedback information. It is important to study the stability of an adaptive controller. Many early adaptive controllers are base on an assumption that the dynamic coefficient matrices are "slowly-varying" such that they can be approximated by constant matrices. The linear-parameterization based adaptive controllers do not require the "slowly-varying" assumption to obtain stability results.

Part II

Contributions

Chapter 5

Adaptive Control of LTV and NTV Plants

In chapter 3, several general MRAC schemes for LTI plants are reviewed and some results on adaptive control of LTV and NTV plants are noted. In reality, many physical problems involve LTV and NTV plants. The mathematic model of LTI plants only cover a small fraction of real physical problems. When the model is applied to real LTV and NTV plants, significant modelling error will be introduced. Until recently, the modelling error of fitting a LTI model to a LTV or NTV plant is simplified, as some bounded disturbances and robustness of adaptive controllers for LTI plants have been studied extensively.

It goes without saying that the best way to minimize the modelling error is to use LTV and NTV plants as models and develop new adaptive controllers for such plants. This problem was successfully studied by Middleton and Goodwin [20], Rotea and Khargonekar [22] as well as Tsakalis and Ioannou [24] [25]. Their results inspire the contributions made in this chapter as an application of MRAC to LTV and NTV plants with uncertain plant terms.

Unlike the adaptive controller developed in [24][25], the proposed MRAC does not require any *à priori* knowledge about the plant parameters. All the plant terms in this Chapter are assumed to be unknown and no linear parameterizations are available. Under this assumption, it is impossible to separate slowly-varying coefficients from some known fast-varying functions. Tsakalis and Ioannou's adaptive controller [24, 25] does not apply in such situations. Besides, the plant terms are potentially unbounded instead of belonging to a convex set as assumed by Middleton and Goodwin [20]. The assumption on the unknown terms is relaxed such that they are not subjected to any constant bound. Some functional bounds are imposed on the plant terms and the functional bounds could grow to infinity if the closed-loop system states are not proved to be bounded. A rigorous stability analysis proves that the proposed MRAC scheme works well under these relaxed assumptions.

5.1 Single Input/Single-Output Plants

The main idea of this research can be explained with a scalar nonlinear time varying plant. Its mathematic model is given by

$$\dot{y}_p = -a_p(y_p, t)y_p + g_p(y_p, t) + b_p(y_p, t)u. \tag{5.1}$$

The model satisfies the following assumption:

Assumption 5.1 *The magnitudes of coefficients a_p, b_p and g_p are bounded by a positive function $k\sqrt{y_p^2+1} = kp(y_p)$ for some positive constant k. Their partial derivatives with respect to both y_p and t are all bounded.*

Observation 5.1 *According to Assumption 5.1, the normalized plant coefficients $\tilde{a}_p = a_p/p(y_p)$, $\tilde{b}_p = b_p/p(y_p)$ and $\tilde{g}_p/p(y_p)$ are bounded by some constants. The time derivative of $\tilde{a}_p$ is given by*

$$\dot{\tilde{a}}_p = \frac{\dot{a}_p}{p(y_p)} - \frac{a_p}{p(y_p)}\frac{\dot{y}_p}{p^2(y_p)}.$$

Since (5.1) implies $|\dot{y}_p| \leq k_a p(y_p)|y_p|$ for some positive k_a, the second term in the above expression is bounded while the first term depends on the magnitude of $\dot{a}_p$. The fact that

$$\dot{a}_p = \frac{\partial a_p}{\partial t} + \frac{\partial a_p}{\partial y_p}\dot{y}_p$$

implies $|\dot{a}_p| \le k_d p(y_p)|y_p$ *for some positive* k_d*. This enables one to conclude that*

$$|\dot{\tilde{a}}_p| \le c_1|e| + c_0| \quad (5.2)$$

for some positive constants c_1 *and* c_0 *where* $|y_p| \le |y_m| + |e|$ *and the boundedness of* $|y_m|$ *have been substituted. The magnitudes of* $\dot{\tilde{b}}_p$ *and* $\dot{\tilde{g}}_p$ *can be bounded by similar positive functions.*

In Section 4, it will be shown that the plant coefficients of a large class of mechanical systems satisfy Assumption 5.1. The objective of this section is to design a control law such that y_p follows the output of a reference model

$$\dot{y}_m = -a_m y_m + r \quad (5.3)$$

as accurately as possible; or that the tracking error, $e = y_p - y_m$, is as small as possible.

5.1.1 Adaptive controller design

Consider a control law given by

$$u = \theta_b u + \theta_a y_p + \theta_g + r = \omega^T \theta + r \quad (5.4)$$

where $\theta^T = [\theta_b, \theta_a, \theta_g]$ and $\omega^T = [u, y_p, 1]$. This controller looks odd because the control input u appears on both sides of (5.4). Yet it can be implemented as

$$u = \frac{1}{1-\theta_b}(\theta_a y_p + \theta_g + r).$$

Equation (5.4) is written for the convenience of stability analysis to be presented later on. Some straightforward calculus will verify the existence of

$$\theta_b^* = 1 - b_p, \qquad \theta_a^* = a_p - a_m, \qquad \theta_g^* = -g_p \quad (5.5)$$

and hence $\theta^{*T} = [\theta_b^*, \theta_a^*, \theta_g^*]$ such that (5.1) controlled by (5.4) can be expressed as

$$(s + a_m)y_p = \omega^T \phi + r. \quad (5.6)$$

Subtracting (5.3) from (5.6), the dynamic equation of tracking error e is then derived as

$$\dot{e} = -a_m e + \omega^T \phi = -a_m e + p(y_p)\omega^T \tilde{\phi} \tag{5.7}$$

where $\phi = \theta - \theta^*$ and $\tilde{\phi} = \tilde{\theta} - \tilde{\theta}^* = (\theta - \theta^*)/p(y_p)$.

Equation (5.7) represents a stable system if either the compensation error ϕ is an all-zero vector or $|\phi^T \omega|$ is uniformly bounded. However, due to the unknown natural of θ^*, it is very unlikely that θ matches θ^* perfectly; and $|\omega^T \phi|$ could grow without bound if y_p is not proven to be bounded. In order to stablize the system, a σ-modified adaptive law is designed to adjust $\theta = \tilde{\theta} p(y_p)$ according to the feedback information. It is synthesized according to

$$\dot{\tilde{\theta}} + \sigma \tilde{\theta} = -\frac{e}{\alpha} \omega p(y_p) \tag{5.8}$$

where α is a positive scalar constant.

The above adaptive law creates a strong coupling effect between the tracking error e and the compensation error ϕ. This effect can be emphasized by subtracting

$$\gamma = \dot{\tilde{\theta}}^* + \sigma \tilde{\theta}^*$$

from both sides of (5.8), which leads to

$$\dot{\tilde{\phi}} + \frac{e}{\alpha} \omega p(y_p) = -\sigma \tilde{\phi} - \gamma. \tag{5.9}$$

In the stability analysis, one is recommended to focus on (5.7) and (5.9). These two equations describe the inter-coupling effect of e and ϕ. Besides, (5.2) and similar treatments on $|b_p(y_p, t)|$ and $|g_p(y_p, t)|$ enables one to write

$$\|\frac{1}{2\sigma}\gamma\| = \frac{1}{4\sigma^2}\|\dot{\tilde{\theta}}^* + \sigma \tilde{\theta}^*\|^2 \leq \gamma_0 + \gamma_1 |e| + \gamma_2 e^2 \tag{5.10}$$

for some positive constants γ_0 γ_1 and γ_2. The potentially unbounded and fast varying effect of θ^* is then modeled by (5.10` mathematically. The three equations (5.7), (5.9) and (5.10) play a key role in the following stability analysis.

5.1.2 Stability analysis

Like the traditional Lyapunov approach, the stability analysis of closed-loop system (5.7) and (5.9) is associated with a positive definite function $v = \frac{1}{2}(e^2+\alpha\tilde{\phi}^T\tilde{\phi})$. However, the objective here is not to establish $\dot{v} \leq 0$ along the system trajectory (5.7) and (5.9). Instead, a finite positive constant v^* is to be found such that $\dot{v} < 0$ whenever $v \geq v^*$. According to Spong and Vidyasagar [64], v is uniformly ultimately bounded if v^* does exist. The closed-loop system will evolve to converge into a state that $\dot{v} = 0$.

Therefore the stability analysis consists of two main steps: 1) find a finite v^* such that $\dot{v} < 0$ whenever $v \geq v^*$; and 2) reduce the area where $\dot{v} \geq 0$ as much as possible.

For the first step, one immediately writes $\dot{v} = e\dot{e} + \alpha\tilde{\phi}^T\dot{\tilde{\phi}}$. A substitution of (5.7) leads to

$$\dot{v} = -a_m e^2 + \alpha\tilde{\phi}^T[\omega p(y_p)\frac{e}{\alpha} + \dot{\tilde{\phi}}].$$

The second term of the right side can be replaced by a negative definite term by substituting (5.9), which yields

$$\begin{aligned} \dot{v} &= -a_m e^2 - \alpha\sigma\tilde{\phi}^T\tilde{\phi} - \alpha\gamma^T\tilde{\phi} \\ &= -a_m e^2 - \alpha\sigma\|\tilde{\phi} + \frac{\gamma}{2\sigma}\|^2 + \alpha\sigma\|\frac{\gamma}{2\sigma}\|^2. \end{aligned} \tag{5.11}$$

The last term in the right side is not negative definite. However, its effect can be conservatively bounded by substituting (5.10). This enables one to write

$$\dot{v} \leq -(a_m - \alpha\sigma\gamma_2)[|e| - \frac{\alpha\sigma\gamma_1}{2(a_m - \alpha\sigma\gamma_2)}]^2 - \alpha\sigma\|\tilde{\phi} + \frac{\gamma}{2\sigma}\|^2 + \tilde{\gamma} \tag{5.12}$$

where $\tilde{\gamma}$ is a positive constant scalar given by

$$\tilde{\gamma} = \alpha\sigma\gamma_0 + \frac{\alpha^2\sigma^2\gamma_1^2}{4(a_m - \alpha\sigma\gamma_2)}.$$

If σ and α are sufficiently small such that $a_m > \alpha\sigma\gamma_2$, then the right side of (5.12) represents an elliptic. Clearly, $\dot{v} < 0$ whenever the point $(|e|, \|\tilde{\phi}\|)$ is outside of this elliptic. Therefore there must exist a finite v^* such that $\dot{v} < 0$ whenever $v > v^*$.

Once the existence of v^* is established, v is proven to be uniformly ultimately bounded; and $\dot{v}$ will eventually converge to zero according to [64]. In the particular case of closed-loop system (5.7) and (5.9), the tracking error and compensation error will eventually converge into

$$(a_m - \alpha\sigma\gamma_2)[|e| - \frac{\alpha\sigma\gamma_1}{2(a_m - \alpha\sigma\gamma_2)}]^2 + \alpha\sigma\|\tilde{\phi} + \frac{\rho}{2\sigma}\|^2 \leq \tilde{\gamma}$$

as (5.12) indicates. This brings the stability analysis to the second step. Now, the tracking error can be controlled by properly adjusting the design parameters a_m, α and σ.

5.1.3 Simulation examples

Two simulation experiments are conducted to test the adaptive controller. In the first experiment. the plant dynamics and the reference model are given by

$$\dot{x}_p(2 + cos(v_p)) + 5x_p - (1 + cos(v_p))(2 + cos(v_p)) = u$$

where $\dot{v}_p = x_p$. The desired trajectory is given by

$$\dot{x}_m + 10x_m = 20 * (1 - cos(\pi t)).$$

All initial values of the control parameters are set to 0. The adaptation parameters are chosen to be $\sigma = 1.0$ and $\alpha = 0.001$. As shown in Fig. 5.1, the tracking error is controlled within $|e| < 0.04$, which is about 2 percent related to the reference model output.

In the second simulation experiment, the plant dynamics and the reference model are given by

$$(s^2 - 0.6|y_p|s + 0.2|y_p|\sin(4t))y_p = [s(2 + \sin(4t)) + 1]u$$

$$\dot{x}_m + 10x_m = 20 * (1 - cos(\pi t))$$

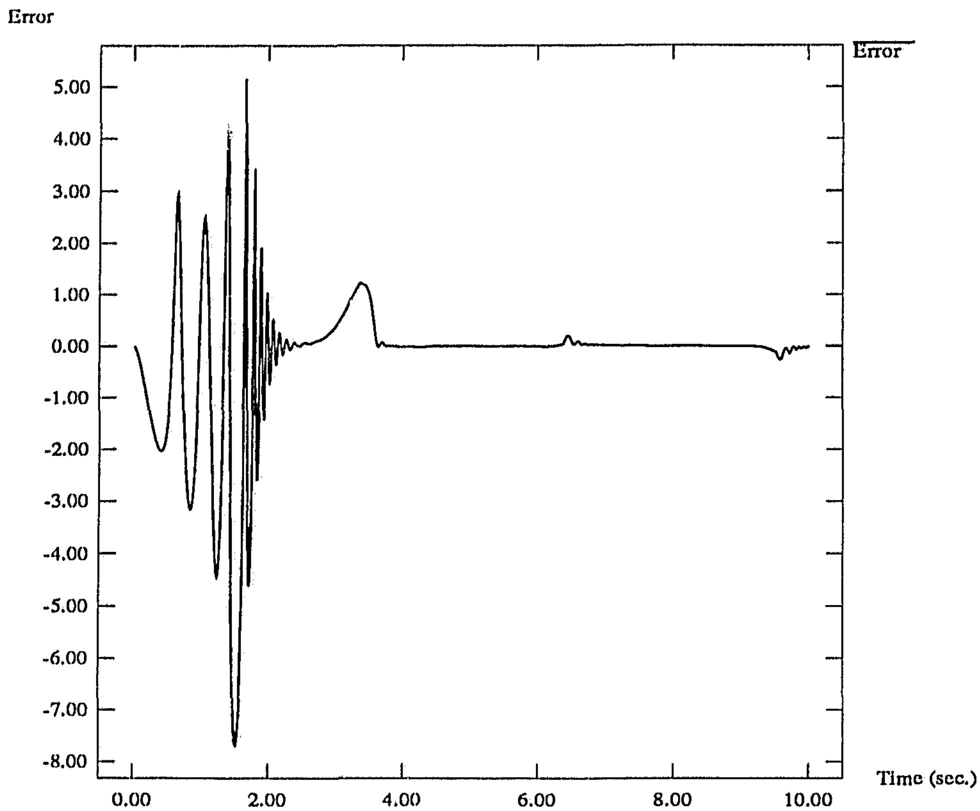

Figure 5.1: Tracking error of an NTV plant with bounded coefficients.

-espectively. All initial conditions and adaptation parameters are chosen in exactly the same way as the first experiment. Again, the closed-loop system exhibits good tracking behavior and the tracking error is plotted in Fig. 5.2. It is observed that the tracking error is kept within 2 percent related to the output of the reference model.

5.2 Multi-variable Plants

The MRAC controller can now be applied to a multi-variable plant given by

$$\dot{y}_p = -A_p(y_p, t)y_p + B_p(y_p, t)[u + g_p(y_p, t)] \tag{5.13}$$

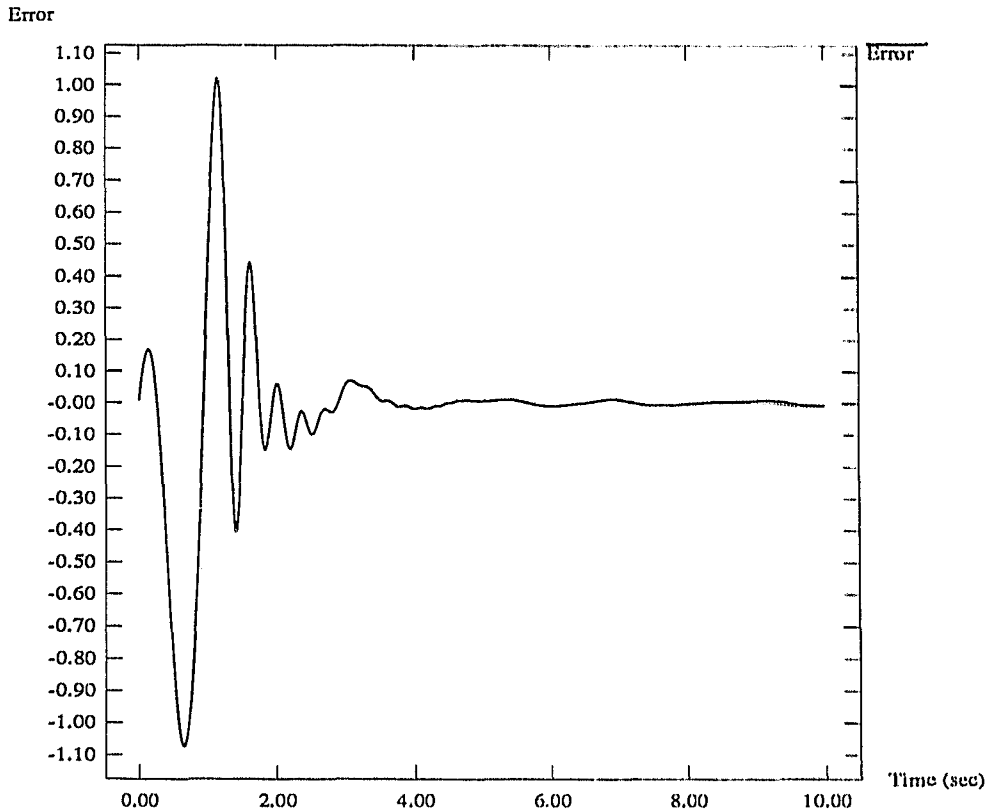

Figure 5.2: Tracking error of an NTV plant with unbounded coefficients.

such that the plant follows a reference model

$$\dot{y}_m = -A_m y_m + B_m r \tag{5.14}$$

where $A_p, A_m \in R^{n\times n}$; $B_p, B_m \in R^{n\times l}$; $y_p, y_m \in R^n$; and $u, r, g_p \in R^l$; $l \leq n$. As in the case of linear time invariant systems, the plant-model pair must satisfy the matching condition [16]

$$(I - B_p B_p^\dagger)(A_p - A_m) = 0 \quad \text{and} \quad (I - B_p B_p^\dagger)B_m = 0 \quad \forall\, y_p, t \tag{5.15}$$

where $B_p^\dagger = (B_p^T B_p)^{-1} B_p^T$.

5.2.1 Adaptive controller design

The control law is given by

$$u = \Theta_b u + \Theta_a y_p + \theta_g + r = \Theta\omega + r \tag{5.16}$$

where $\Theta = [\Theta_b, \Theta_a, \theta_g]$ and $\omega^T = [u, y_p^T, 1]$. Similar to the case of the previous section, (5.16) is actually implemented as

$$u = (I - \Theta_b)^{-1}(\Theta_a y_p + \theta_g + r).$$

it can be verified that there exist a set of "ideal model reference controller" coefficients. They can be derived as

$$\Theta_b^* = I - (B_p^\dagger B_m)^{-1}, \quad \Theta_a^* = (B_p^\dagger B_m)^{-1} B_p^\dagger (A_p - A_m), \tag{5.17}$$

$$\theta_g^* = -(B_p^\dagger B_m)^{-1} g_p \text{ and } \Theta^* = [\Theta_b^*, \Theta_a^*, \theta_g^*]. \tag{5.18}$$

When the exact plant coefficient matrices A_p, B_p and g_p are available, one can use (5.17) and (5.18) to compute Θ^* and then verify that the nonlinear time varying effect caused by A_p, B_p and g_p is well compensated. The resulting system becomes (5.14).

When the plant coefficient matrices are not available, an adaptive matrix Θ is used to substitute Θ^*. Denote the compensation error as $\Phi = \Theta - \Theta^* = [\Phi_b, \Phi_a, \phi_g]$ where $\Phi_a = \Theta_a - \Theta_a^*$, $\Phi_b = \Theta_b - \Theta_b^*$ and $\phi_g = \theta_g - \theta_g^*$. Then (5.16) can be written as

$$u = (I - \Theta_b^*)^{-1}[\Theta_a^* y_p + \theta_g^* + r + \Phi\omega].$$

Substituting (5.17) and (5.18) into the above equation, one obtains

$$u = B_p^\dagger (A_p - A_m) y_p - g_p + B_p^\dagger B_m (r + \Phi\omega) \tag{5.19}$$

which is another form of the control law. By applying (5.19) to the plant of (5.13), one arrives at

$$\dot{y}_p = -A_p y_p + B_p B_p^\dagger (A_p - A_m) y_p + B_p B_p^\dagger B_m (r + \Phi\omega).$$

The matching condition given by (5.15) enables one to rewrite the above expression as

$$\dot{y}_p = -A_m y_p + B_m(\Phi\omega + r). \tag{5.20}$$

Let $e = y_p - y_m$ denote the tracking error. Then

$$\dot{e} = -A_m e + B_m \Phi\omega \tag{5.21}$$

can be obtained by subtracting (5.14) from (5.20).

Sine A_m and B_m are prescribed by the designer, one can assign the eigenvalues of A_m to the right half of the complex plane. Then (5.21) represents an exponentially stable system if either Φ is an all-zero matrix or $\|\Phi\omega\|$ remains bounded. However, this hope is hardly true in real applications where A_p, B_p, g_p and hence Θ^* are not available to the controller. An adaptation law is proposed to adjust Θ, which is given by

$$\dot{\Theta} + \sigma\Theta = -\frac{1}{\alpha} B_m^T P e \omega^T \tag{5.22}$$

where $P > 0$ satisfies $0.5(A_m^T P + P A_m) = Q > 0$. The above equation can be expressed in terms of the compensation error Φ by subtracting $\Gamma = \dot{\Theta}^* + \sigma\Theta^*$ from both sides. As a result, one obtains

$$\dot{\Phi} + \frac{1}{\alpha} B_m^T P e \omega^T = -\sigma\Phi - \Gamma. \tag{5.23}$$

The following stability analysis will show that the adaptive controller synthesized by (5.16) and (5.22) leads to an uniformly ultimately bounded closed-loop system. Similar to the analysis of sub-section 2.2, the stability analysis will focus on the two error equations (5.21) and (5.23), which describe the coupling effect of the tracking error e and compensation error Φ.

5.2.2 Stability analysis

Consider a positive definite function $v = \frac{1}{2}[e^T P e + \alpha Tr\{\Phi\Phi^T\}]$ where $Tr\{X\}$ denotes the trace of a matrix X while $\|X\|_F = \sqrt{Tr\{XX^T\}}$ its Frobenious norm. A Lyapunov-like analysis will lead to

$$\dot{v} = -e^T Q e - \alpha\sigma\|\Phi + \frac{1}{2\sigma}\Gamma\|_F^2 + \frac{\alpha}{4\sigma}\|\Gamma\|_F^2. \tag{5.24}$$

Equation (5.24) can be viewed as a general case of (5.11). In fact, the adaptive controller synthesized by (5.16) and (5.22) is a generalized version of (5.4) and (5.8). Accordingly, the coupling effect described by (5.21) and (5.23) are generalized forms of (5.7) and (5.9) respectively. The detailed derivation of (5.24) is omitted here because it is identical to that of (5.11). The minor difference is some additional matrix calculus such as $Q = A_m^T P + PA_m$.

When $||\Gamma||_F$ is uniformly bounded, then (5.24) implies the existence of a finite v^* such that $\dot{v} < 0$ whenever $v \geq v^*$. The closed-loop system is uniformly ultimately bounded according to [64]. If the entries of Θ^* matrix are not bounded by a constant but by some normalizing signal $p(y_p)$ such that $|\theta^*_{ij}| \leq k\sqrt{1 + ||y_p||^2}$ for some $k > 0$, then $||\Gamma||_F^2 \leq \gamma_0 + \gamma_1||e|| + \gamma_2||e||^2$ for some positive constants γ_0, γ_1 and γ_2. Following the same argument as subsection 2.2, one can further obtain an inequality similar to (5.12). The closed-loop system can still be shown to be uniformly bounded and the tracking error can be controlled by adjusting α, σ and the smallest eigenvalue of Q.

5.2.3 Discussions

The plant described by (5.13) has a useful specific form. Consider an n-th order single-input-single-output plant described by

$$y_p^n - \sum_{i=0}^{n-1} a_{pi} y_p^i = k_p u.$$

If all the states of the plant are available, then the matrices A_p and B_p may be written as

$$A_p = -\begin{bmatrix} a_{pn-1} & \cdots & a_{p1} & a_{p0} \\ 1 & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 1 & 0 \end{bmatrix} \qquad B_p^T = [k_p, 0, \ldots, 0]$$

where k_p represents the gain of the plant. In this case $l = 1$. If one specifies a model pair

$$A_m = -\begin{bmatrix} a_{mn-1} & \dots & a_{m1} & a_{m0} \\ 1 & \dots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & 1 & 0 \end{bmatrix} \qquad B_m^T = [k_m, 0, \dots, 0],$$

then it is easy to confirm that (5.15) is satisfied. The ideal coefficents are given by

$$\Theta_b^* = 1 - \frac{k_p}{k_m}, \quad \Theta_a^* = \frac{1}{k_m}[a_{mn-1} - a_{pn-1}, \dots, a_{m1} - a_{p1}, a_{m0} - a_{p0}], \quad \theta_g^* = 0.$$

According to the analysis in subsection 3.2, the tracking error can be controlled within a computable tolerance as long as the magnitudes of plant coefficients $\{a_{pi}\}_{i=0}^{n-1}$ are bounded by $k_i\sqrt{y_p^2 + 1}$ for some positive k_i.

5.3 An Application Example

A typical application of the MRAC scheme is the tracking control of a robotic manipulator. The manipulator dynamics are described by a second-order differential equation

$$M(q)\ddot{q} + V(\dot{q}, q)\dot{q} + G(q) = \tau \tag{5.25}$$

where q denotes the joint position vector and $\tau \in R^n$ the joint torque vector (control input); $M(q)$, $V(\dot{q}, q) \in R^{n\times n}$ and $G(q) \in R^n$.

Denote as

$$y_p^T = [\dot{q}^T, q^T], \quad A_p = M^{-1}(q)\begin{bmatrix} V(\dot{q}, q) & 0 \\ -I & 0 \end{bmatrix}, \quad B_p = \begin{bmatrix} M^{-1}(q) \\ 0 \end{bmatrix}.$$

Then the system can be represented by a state space equation $\dot{y}_p = -A_p y_p + B_p(\tau - G(q))$. A reference model is prescribed as $\dot{y}_m = -A_m y_m + B_m r$ where $r \in R^n$ is the reference input and

$$A_m = \begin{bmatrix} K_v & K_p \\ -I & 0 \end{bmatrix}, \quad B_m = \begin{bmatrix} I \\ 0 \end{bmatrix}.$$

It is not difficult to verify that this model satisfies the matching condition of (5.15).

The control objective is to minimize the tracking error $e = y_p - y_m$. A routine analysis will derive the dynamic equation of the tracking error as

$$\dot{e} = -A_m e + B_p(\tau - G(q)) + \begin{bmatrix} K_v - M^{-1}(q)V(\dot{q}, q) & K_p \\ 0 & 0 \end{bmatrix} y_p - B_m r$$

which is equivalent to

$$\begin{aligned} \dot{e} &= -A_m e + B_p(\tau - \theta_g^* - \Theta_a^* y_p - \Theta_b^* r) \\ &= -A_m e + B_p(\tau - \theta_g^* - \tilde{\Theta}_a^* \tilde{y}_p - \Theta_b^* r) \end{aligned} \quad (5.26)$$

where $\theta_g^* = G(q)$, $\Theta_b^* = M(q)$, $\Theta_a^* = [\Theta_{a1}^*, \Theta_{a2}^*]$, $\Theta_{a1}^* = M(q)K_v - V(\dot{q}, q)$, $\Theta_{a2}^* = M(q)K_p$ and

$$\tilde{\Theta}_a^* = [\frac{1}{p(\|\dot{q}\|)}\Theta_{a1}^*, \Theta_{a2}^*] \qquad \tilde{y}_p = \begin{bmatrix} \dot{q}p(\|\dot{q}\|) \\ q \end{bmatrix}.$$

Fact F1: For robotic manipulators, the entries of $M(q)$, $V(\dot{q}, q)/\sqrt{\|\dot{q}\|^2 + 1}$ and $G(q)$ are bounded functions of q [63, 64]; the inertial matrix $M(q)$ is a uniformly positive definite. Consequently, the entries of $\dot{M}(q)$, $\frac{d}{dt}[V(\dot{q}, q)/\sqrt{\|\dot{q}\|^2 + 1}]$ and $\dot{G}(q)$ are bounded by $k_r\sqrt{\|\dot{q}\|^2 + 1} = k_r p(\|\dot{q}\|)$ for some positive constant k_r.

This suggests that the stability analysis of general-case MRAC systems also applies to the MRAC of robotic manipulators. The control law is proposed as

$$\tau = \theta_g + \tilde{\Theta}_a \tilde{y}_p + \Theta_b r = \Theta\omega \quad (5.27)$$

where $\Theta = [\theta_g, \tilde{\Theta}_a, \Theta_b]$, $\omega^T = [1, \tilde{y}_p^T, r^T]$; $\tilde{\Theta}_a$, Θ_b and θ_g are adaptive signals updated according to some adaptive law.

When Θ_a^*, Θ_b^* and θ_g^* are used to synthesize (5.27), then one will achieve a perfect model $\dot{e} = -A_m e$ by substituting (5.27) into (5.26). Unfortunately, Θ_a^*, Θ_b^* and θ_g^* are not available. The adaptive matrices Θ_a, Θ_b and θ_g may not compensate perfectly. It is important to study the effect of compensation errors.

According to **F1**, Observation 3.7 and the construction of Θ^*, it is clear that the entries of Θ^* are all bounded functions of q. Therefore

$$\|\Theta^*\|_F \leq \gamma_1 \quad \text{and} \quad \|\dot{\Theta}^*\|_F \leq \gamma_2 p(\|\dot{q}\|); \quad \text{where} \;\; \gamma_1 > 0, \quad \gamma_2 > 0. \qquad (5.28)$$

These two bounds will be useful in the stability analysis.

Substituting (5.27) into (5.26) leads to

$$\begin{aligned} \dot{e} &= -A_m e + B_p(\phi_g + \tilde{\Phi}_a \tilde{y}_p + \Phi_b r) \\ &= -A_m e + B_p \Phi \omega \end{aligned} \qquad (5.29)$$

where $\phi_g = \theta_g - \theta_g^*$, $\Phi_b = \Theta_b - \Theta_b^*$, $\tilde{\Phi}_a = \dot{\Theta}_a - \dot{\Theta}_a^*$, $\Phi = \Theta - \Theta^*$ and $\Theta^* = [\theta_g^*, \dot{\Theta}_a^*, \Theta_b^*]$. The reference model is chosen so that there exist positive definite matrices P and Q such that

$$PA_m + A_m^T P = Q.$$

The stability of the system is again related to a positive definite function

$$v = e^T P e + \alpha Tr\{\Phi^T C(q) \Phi\}$$

where $C(q) = M^{-1}(q)$. The time derivative of v evaluated along (5.29) is written as

$$\begin{aligned} \dot{v} &= -e^T Q e + 2e^T P B_p \Phi \omega + 2\alpha Tr\{\dot{\Phi}^T C(q) \Phi\} + \alpha Tr\{\Phi^T \dot{C}(q) \Phi\} \\ &= -e^T Q e + 2Tr\{[\omega e^T P B_p + \alpha \dot{\Phi}^T C(q)] \Phi\} + \alpha Tr\{\Phi^T \dot{C}(q) \Phi\} \end{aligned} \qquad (5.30)$$

where an identity $x^T y = Tr\{y x^T\}$ has been substituted. Let

$$P = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix}, \quad \text{then} \quad PB_p = \begin{bmatrix} P_{11} \\ P_{21} \end{bmatrix} M^{-1}(q) = P^* M^{-1}(q) = P^* C(q).$$

Substituting the last equality into (5.30) leads to

$$\dot{v} = -e^T Q e + 2Tr\{[\omega e^T P^* + \alpha \dot{\Phi}^T] C(q) \Phi\} + \alpha Tr\{\Phi^T \dot{C}(q) \Phi\}. \qquad (5.31)$$

As in the cases of the previous two sections, (5.31) is potentially unstable because of the compensation error Φ and its unknown feedback effect $\Phi\omega$. An adaptive law is proposed to stablize the system. It is synthesized by

$$\dot{\Theta} = -\sigma \Theta - \frac{1}{\alpha} P^{*T} e \omega^T$$

which can be expressed in terms of the compensation error as

$$\dot{\Phi} = -\sigma\Phi - \frac{1}{\alpha}P^{*T}e\omega^T - R$$

where $R = \dot{\Theta}^* + \sigma\Theta^*$. Substituting the above expression into (5.31) results in

$$\dot{v} = -e^TQe - \alpha Tr\{\Phi^T[2\sigma C(q) - \dot{C}(q)]\Phi\} - \alpha Tr\{R^TC(q)\Phi\}. \quad (5.32)$$

An important step in the stability proof is to make $2\sigma C(q) - \dot{C}(q)$ positive definite at all times by a proper choice of σ. It is pointed out in [64] that the inertial matrix satisfies the following relations

$$\lambda_{M1} \leq M(q) \leq \lambda_{M2} \quad \text{for} \quad 0 < \lambda_{M1} < \lambda_{M2}.$$

Thus $\lambda_{M2}^{-1} \leq C(q) = M^{-1}(q) \leq \lambda_{M1}^{-1}$. The entries of $C(q)$ must be bounded functions of q. Consequently, the entries of $\dot{C}(q)$ involve linear products of $\dot{q}$ with some bounded functions of q. This observation implies that $\|\dot{C}(q)\| \leq \lambda_C\|\dot{q}\|$ for some $\lambda_C > 0$.

Let λ_{min} be the lower bound of the eigenvalues of matrix $2\sigma C(q) - \dot{C}(q)$. It is clear that

$$\lambda_{min} \geq 2\sigma\lambda_{M2}^{-1} - \lambda_C\|\dot{q}\|.$$

There must exist $\sigma_0 > 0$ and $\sigma_1 > 0$ such that if we substitute $\sigma = \sigma_0 + \sigma_1\|\dot{q}\|$ into the above expression, then $\lambda_{min} > 0$ at all times. Now (5.32) can be simplified as

$$\begin{aligned} \dot{v} &\leq -\lambda_Q\|e\|^2 - \alpha\lambda_{min}\|\Phi\|_F^2 + \frac{\alpha}{\lambda_{M1}}\|R\|_F\|\Phi\|_F \\ &= -\lambda_Q\|e\|^2 - \alpha\lambda_{min}(\|\Phi\|_F - \lambda_{M1}^{-1}\lambda_{min}^{-1}\|R\|_F)^2 + \frac{\alpha}{4}\lambda_{M1}^{-2}\lambda_{min}^{-2}\|R\|_F^2. \quad (5.33) \end{aligned}$$

Since the reference model must be a stable system, y_m is always bounded. The stability of y_p is equivalent to the stability of $e = y_p - y_m$. According to (5.28), there exist $k_0 > 0$, $k_1 > 0$ and $k_2 > 0$ such that

$$\|R\|_F^2 \leq k_0 + k_1\|e\| + k_2\|e\|^2.$$

It follows that

$$\dot{v} = -\lambda_Q\|e\|^2 - \alpha\lambda_{min}(\|\Phi\|_F - \lambda_{M1}^{-1}\lambda_{min}^{-1}\|R\|_F)^2 + \tilde{k}_2\|e\|^2 + \tilde{k}_1\|e\| + \tilde{k}_0 \quad (5.34)$$

where

$$\tilde{k}_i = \frac{\alpha}{4}\lambda_{M1}^{-1}\lambda_{min}k_i \quad 0 \le i \le 2.$$

The next step to ensure stability for the closed-loop system is to specify $\lambda_Q > \tilde{k}_2$. Since $Q = PA_M + A_M^T P$, this may be achieved by a sufficiently large velocity feedback gain K_v when both K_v and K_p are diagonal matrices. Now (5.34) becomes

$$\begin{aligned} \dot{v} &= -(\lambda_Q - \tilde{k}_2)(\|e\| - \frac{\tilde{k}_1}{2(\lambda_Q - \tilde{k}_2)})^2 - \alpha\lambda_{min}(\|\Phi\|_F - \lambda_{M1}^{-1}\lambda_{min}^{-1}\|R\|_F)^2 \\ &\quad + \tilde{k}_0 + \frac{\tilde{k}_1^2}{4(\lambda_Q - \tilde{k}_2)^2}. \end{aligned} \tag{5.35}$$

The two positive terms in the above expression are constants. The varying terms are either zero or negative. Equation (5.35) suggests the existence of $\lambda_e > 0$ and $\lambda_\phi > 0$ such that whenever $\|e\| \ge \lambda_e$ or $\|\Phi\|_F \ge \lambda_\phi$, then either

$$\dot{v} \le -(\lambda_Q - \tilde{k}_2)(\lambda_e - \frac{\tilde{k}_1}{2(\lambda_Q - \tilde{k}_2)})^2 + \tilde{k}_0 + \frac{\tilde{k}_1^2}{4(\lambda_Q - \tilde{k}_2)^2} = 0$$

or

$$\dot{v} \le -\alpha\lambda_{min}(\lambda_\phi - \lambda_{M1}^{-1}\lambda_{min}^{-1}\|R\|_F)^2 + \tilde{k}_0 + \frac{\tilde{k}_1^2}{4(\lambda_Q - \tilde{k}_2)^2} = 0.$$

Consider a finite constant $v^* = \lambda_P\lambda_e^2 + \alpha\lambda_{M2}\lambda_\phi$ where λ_P and λ_{M2} denote the largest eigenvalues of matrices P and $C(q)$ respectively, then it is clear that $\dot{v} \le 0$ whenever $v \ge v^*$. Thus the tracking error e and the system coefficient error matrix Φ are uniformly ultimately bounded.

Throughout the whole stability analysis, nothing is assumed about the velocity of the manipulator. This means the stability results are not based on a rather unrealistic assumption that the robot motion is very slow such that $\dot{M}(q) \approx 0$, $\dot{V}(\dot{q}, q) \approx 0$ and $\dot{G}(q) \approx 0$. To the best of the author's knowledge, this is the only version of a MRAC robot controller that is accompanied with such a rigorous stability analysis. Most of the MRAC applications do rely on the assumption of slow robot movement to back up their stability analysis [27]-[30].

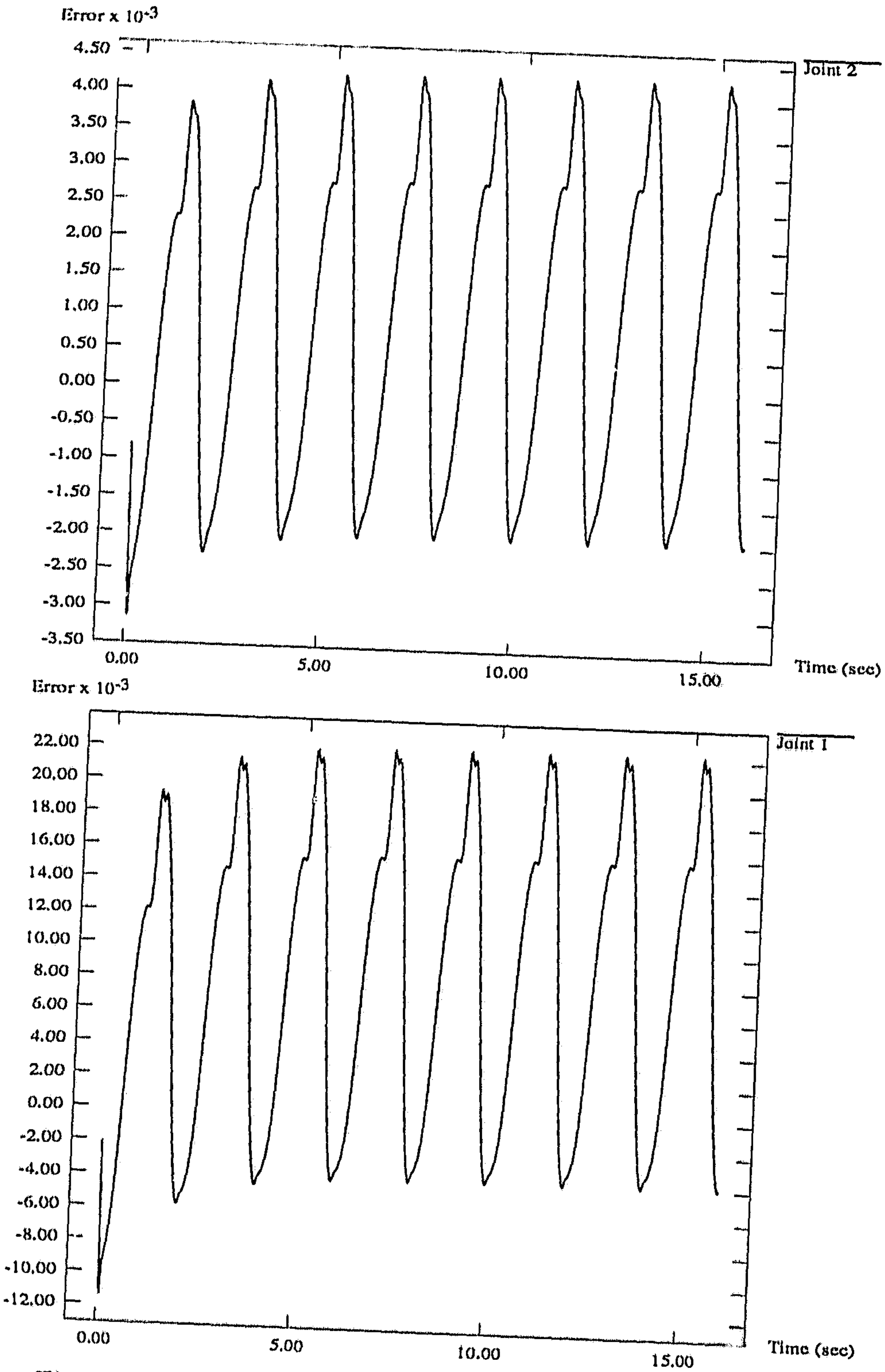

Figure 5.3: Tracking errors of a manipulator (without training).

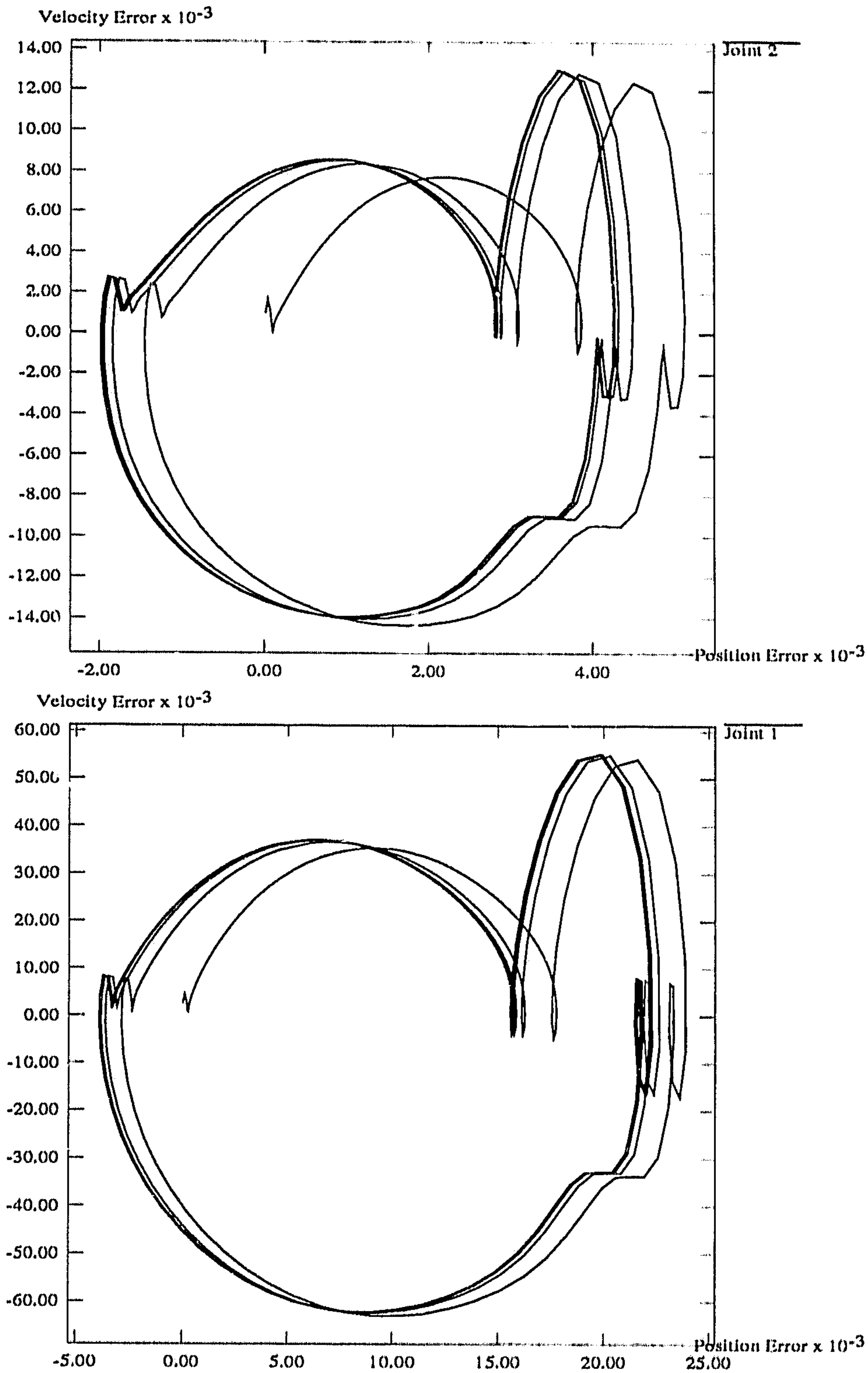

Figure 5.4: Tracking errors of a manipulator (with training).

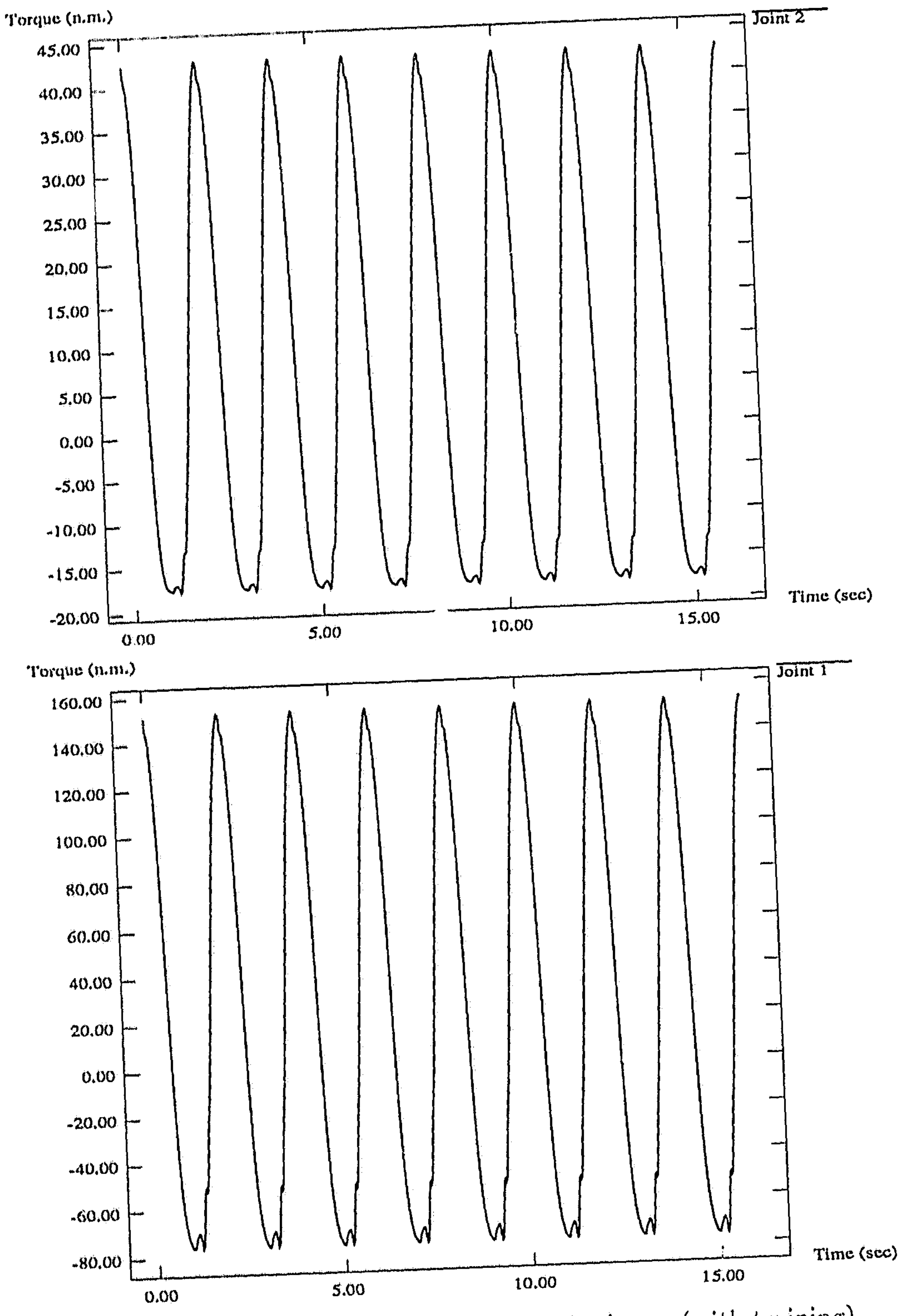

Figure 5.5: Torques applied to the manipulator. (with training).

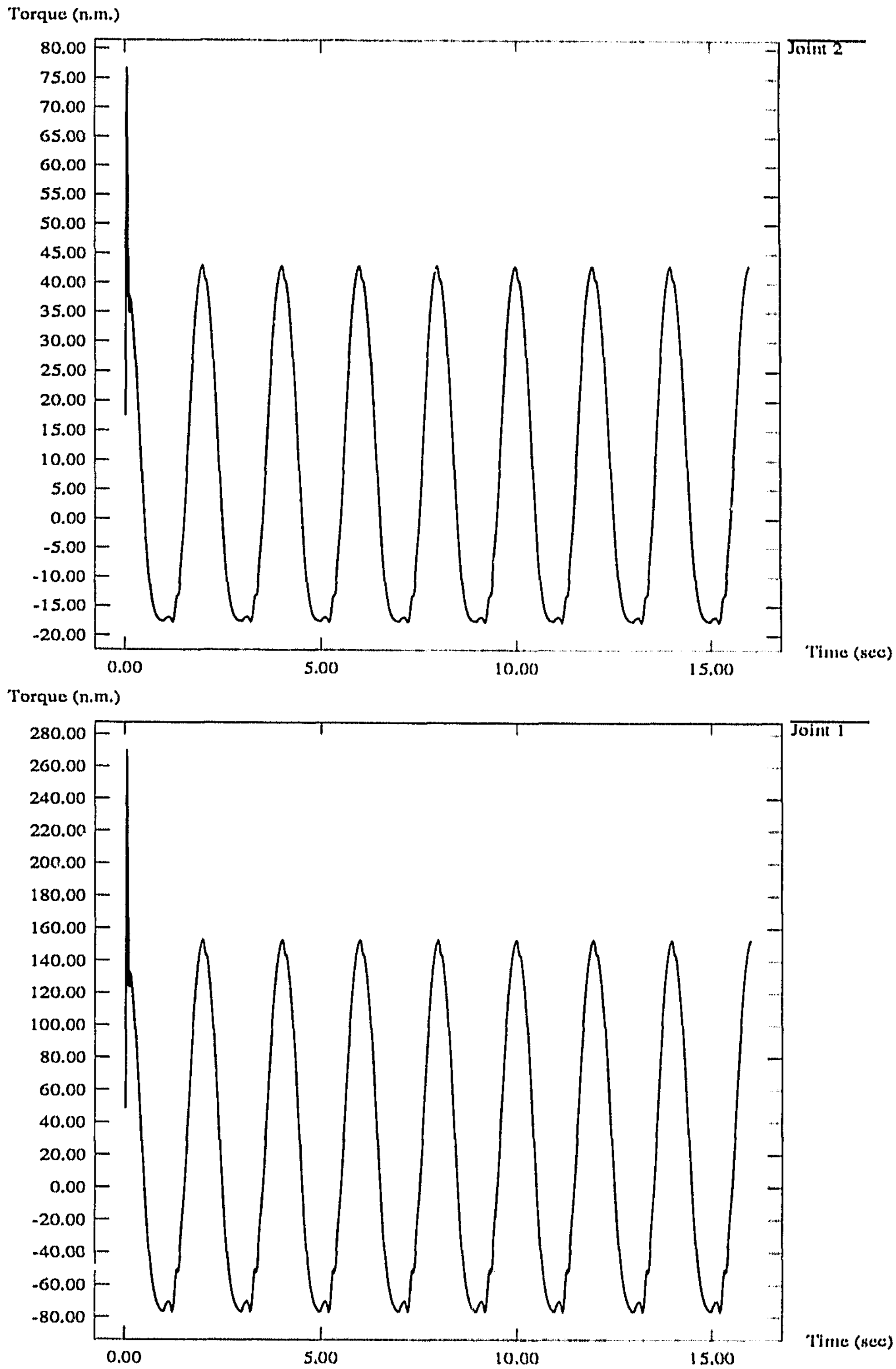

Figure 5.6: Torques applied to the manipulator (without training).

Computer simulation experiments are conducted to test the adaptive controllers. A two-link planar robot described by

$$M(q)\begin{bmatrix}\ddot{q}_1\\ \ddot{q}_2\end{bmatrix}+V(\dot{q},q)\begin{bmatrix}\dot{q}_1\\ \dot{q}_2\end{bmatrix}+G(q)=\begin{bmatrix}\tau_1\\ \tau_2\end{bmatrix} \tag{5.36}$$

is used as the control object, where

$$M(q)=\begin{bmatrix}(2l_1\cos(q_2)+l_2)l_2m_2+l_1^2(m_1+m_2) & l_2^2m_2+l_1l_2\cos(q_2)m_2\\ l_2^2m_2+l_1l_2\cos(q_2)m_2 & l_2^2m_2\end{bmatrix},$$

$$V(\dot{q},q)=\begin{bmatrix}-2l_1l_2m_2\sin(q_2)\dot{\theta}_2 & -l_1l_2m_2\sin(q_2)\dot{\theta}_2\\ l_1l_2m_2\sin(q_2)\dot{q}_1 & 0\end{bmatrix}$$

and

$$G(q)=\begin{bmatrix}g(m_2l_2\cos(q_1+q_2)+(m_1+m_2)l_1\cos(q_1))\\ m_2l_2g\cos(q_1+q_2)\end{bmatrix}$$

where $l_1=.7$, $l_2=.5$ (meter), $m_1=10$ and $m_2=5$ (kg). The desired trajectories are given by

$$q_1=q_2=1-\cos(\pi t).$$

The reference model is characterized by

$$\ddot{q}_m+K_v\dot{q}_m+K_pq_m=r$$

where $K_v=40I$ and $K_p=60I$. The adaptive controller parameters are chosen to be $\sigma=0.001\sqrt{0.1+\dot{q}^T\dot{q}}$ and $\alpha=0.001$. The tracking errors are plotted in Fig. 5.3. If the controller saves the adaptive parameter matrix Θ and retrieves it in the next task (we call it a "training" process), then the tracking error can be reduced as Fig. 5.4 illustrates (the phase plots). The control torques are also plotted in Figs. 5.5 and 5.6, with and without training respectively. It is observed that by using the previous Θ matrix value as the initial guess, the control torque can be significantly reduced.

5.4 LTV Plants of Relative Degree 1

We now consider n-th order linear-time-varying (LTV) plants described by

$$[s^n + \sum_{i=0}^{n-1} a_i(t)s^i]y_p = A(s,t)y_p = B(s,t)u = \sum_{j=0}^{n-1} b_j(t)s^j u \tag{5.37}$$

where "s" denotes a differential operator; $\{a_i\}_{i=0}^{n-1}$ and $\{b_j\}_{j=0}^{n-1}$ are $2n$ plant coefficients; $A(s,t)$ and $B(s,t)$ are LTV differential operators defined in [24].

Assumption 5.2 *A general requirement of these plants is that the inverse plant $B^{-1}(s,t)A(s,t)y_p$ be bounded-in-bounded-out (BIBO). The necessary condition for the inverse plant to be (BIBO) is that all roots of $B(s,t)$ are located within the left half of a complex plane. This means that $\{b_i(t)\}_{j=0}^{n-1}$ will not change signs. It is commonly assumed that the sign of $B(s,t)$ and the order "n" are known. For LTV plants, we also assume that $\{a_i(t)\}_{i=0}^{n-2}$ and $\{b_j(t)\}_{j=0}^{n-2}$ are $(n-1)$ times differentiable; $a_{n-1}(t)$ and $b_{n-1}(t)$ are $(2n-2)$ times differentiable. These derivatives are all bounded.*

In the following analysis, Leibniz's rule proves to be a good tool:

$$\frac{d^n}{dt^n}[g(t)h(t)] = \sum_{i=1}^{n} \frac{n!}{i!(n-i)!}\frac{d^i g}{dt^i}\frac{d^{n-i}h}{dt^{n-i}} \tag{5.38}$$

One can image $h(t)$ as the destinate function on which a differential operation applies, $g(t)$ as a time-varying coefficient associated to the right of operator s^n. Then (5.38) describes an operator $s^n g(t)$, which is equivalent to several lower-order differential operators in addition to $g(t)s^n$. This is the major difference between LTV and LTI operators.

5.4.1 System analysis

We use a modified "standard" MRC control law to control the plants

$$u = [\psi_1(t) + \Theta_1(s,t)F^{-1}(s)]u + [\psi_2(t) + \Theta_2(s,t)F^{-1}(s)]y_p + r' \tag{5.39}$$

where "r'" will be given a little bit later in (5.48); meanwhile, we accept it as a modified input signal which is different from the reference input "r"; Θ_1 and Θ_2 are $(n-2)$-th order LTV differential operators; $F^{-1}(s)$ is an arbitrary $(n-1)$-th order LTI integral operator with all poles in the left half of the complex plane. Here we choose $F(s)$ such that it has different roots. Again, the control input u appears on both sides of (5.39). In practice, (5.39) is implemented as

$$u = [1 - \psi_1(t) - \Theta_1(s,t)F^{-1}(s)]^{-1}\{[\psi_2(t) + \Theta_2(s,t)F^{-1}(s)]y_p + r'\}.$$

Lemma 5.1 *There exists a set of "ideal" coefficients ψ_1^*, ψ_2^* and operators Θ_1^*, Θ_2^* such that plant (5.37) controlled by (5.39) can be expressed as*

$$\dot{y}_p = -a_m y_p + (\phi_1 u + \Phi_1 F^{-1} u + \Phi_2 F^{-1} y_p + \phi_2 y_p) + r' - \theta_g^* \tag{5.40}$$

where $\phi_1 = \psi_1 - \psi_1^*$, $\phi_2 = \psi_2 - \psi_2^*$, $\Phi_1 = \Theta_1 - \Theta_1^*$, $\Phi_2 = \Theta_2 - \Theta_2^*$ and

$$|\theta_g^*| \le \sum_{i=1}^{n-1} \frac{c_i}{(s+\lambda_i)}(|u| + |y_p|) = m_b \tag{5.41}$$

where λ_i are the different roots of $F(s)$ and $c_i > 0$ for $1 \le i \le (n-1)$.

Proof: (5.39) is equivalent to

$$[F(1-\psi_1^*) - F\Theta_1^* F^{-1}]u - (F\Theta_2^* F^{-1} + F\psi_2^*)y_p = F(\omega^T\phi + r') \tag{5.42}$$

$$\phi = \theta - \theta^* \qquad \theta^T = [\psi_1, \Theta_1^T, \Theta_2^T, \psi_2] \qquad \theta^{*T} = [\psi_1^*, \Theta_1^{*T}, \Theta_2^{*T}, \psi_2^*]$$

$$\omega^T = [u, \omega_1^T, \omega_2^T, y_p] \qquad \dot{\omega}_1 = A_f\omega_1 + b_f u \qquad \dot{\omega}_2 = A_f\omega_2 + b_f y_p$$

where (A_f, b_f) describe F^{-1} in state space. By a proper choice of the matrix pair (A_f, b_f), the vectors ω_1 and ω_2 will represent full states of the two low-pass filers $F^{-1}u$ and $F^{-1}y_p$ respectively. In other words, $\omega_1 = [\omega_1, \dot{\omega}_1, \ldots, \omega_1^{(n-1)}]^T$. This implies that

$$\Theta_1^{*T}\omega_1 = \sum_{i=1}^{n} \theta_{1i}^* \frac{d^{i-1}}{dt^{i-1}}\omega_i = (\sum_{i=1}^{n} \theta_{1i}^* s^{i-1})\omega_1.$$

In the following discussion, we will use expressions like

$$\Theta_1^* F^{-1} u = \Theta_1^{*T}\omega_1 \quad \text{and} \quad \Theta_2^* F^{-1} y_p = \Theta_2^{*T}\omega_2.$$

On the left side of the above expressions, Θ_1^* and Θ_2^* represent two TV operators, whereas on the right side, they represent two TV vectors. In the standard MRC scheme, the coefficients of operators Θ_1 and Θ_2 can be directly converted to the components of vectors Θ_1^* and Θ_2^* provided that (A_f, b_f) are properly specified. Thus in the following analysis, we will refer to Θ_1^* and Θ_2^* as both operators and vectors.

Let Θ represent either Θ_1^* or Θ_2^*, and $\{\theta_i(t)\}_{i=0}^{n-2}$ its coefficients. On applying (5.38), we obtain

$$\begin{aligned} F\Theta &= \sum_{m=0}^{2n-3} \sum_{i+j=m} f_i s^i(\theta_j s^j) = \sum_{m=0}^{2n-3} \sum_{i+j=m} f_i \sum_{k=0}^{i} C_i^k \theta_j^{(k)} s^{m-k} \\ &= \sum_{m=0}^{2n-3} \sum_{i+j=m} f_i \theta_j s^m + \sum_{m=1}^{2n-3} \sum_{i+j=m} f_i \sum_{k=1}^{i} C_i^k \theta_j^{(k)} s^{m-k} \\ &= \Theta F + F \otimes \Theta \end{aligned} \quad (5.43)$$

where $C_i^k = \frac{i!}{(i-k)!k!}$, $\theta_j^{(k)}$ denotes the k-th order derivative of θ_j. We use $F \otimes \Theta$ to denote the non-commutable part of $F\Theta$. With this we can re-arrange (5.42) such that

$$\begin{aligned} & \{[F(1-\psi_1^*) - \Theta_1^*]B^{-1}A - \Theta_2^* - F\psi_2^*\}y_p \\ = \; & F(\omega^T \phi + r') + F \otimes \Theta_1^* F^{-1} u + F \otimes \Theta_2^* F^{-1} y_p. \end{aligned} \quad (5.44)$$

Like the standard MRC approach, we first determine ψ_1^* and Θ_1^* such that

$$\begin{aligned} & F(1-\psi_1^*) - \Theta_1^* \\ = \; & f_{n-1}(1-\psi_1^*)s^{n-1} + \sum_{i=0}^{n-2} [f_i(1-\psi_1^*) - \theta_{1i}^* + \tilde{f}_i] s^i \\ = \; & \sum_{i=0}^{n-1} b_i s^i = B(s,t) \end{aligned} \quad (5.45)$$

where $\{f_i\}_{i=0}^{n-1}$ and $\{\theta_{1i}^*\}_{i=0}^{n-2}$ are the coefficients of F and Θ_1^* respectively; $\{\tilde{f}_i\}_{i=0}^{n-2}$ are the additional coefficients derived by substituting (5.38) to $F(1-\psi_1^*)$. Once $\psi_1^* = 1 - b_{n-1}$ is determined, $\{\tilde{f}_i\}_{i=0}^{n-2}$ are all determined. They are linear combinations of the derivatives of $b_{n-1}(t)$ up to the $(n-1)$-th order. The coefficients of

Θ_1^* are then determined by

$$\theta_{1i}^* = f_i(1-\psi_1^*) + \tilde{f}_i - b_i \qquad 0 \le i \le n-2.$$

Next, we determine ψ_2^* and then Θ_2^* such that

$$A(s,t) - \Theta_2^* - F\psi_2^* = F(s+a_m). \tag{5.46}$$

This is similar to the above process because the coefficients of s^n in both $A(s,t)$ and $F(s)(s+a_m)$ are 1. Only n coefficients $\{a_i\}_{i=0}^{n-1}$ have to be matched. Now (5.44) reduces to (5.40) which is equivalent to

$$\dot{y}_p = -a_m y_p + \omega^T \phi + r' - \theta_g^* \tag{5.47}$$

where $-\theta_g^* = F^{-1}(F \otimes \Theta_1^*)F^{-1}u + F^{-1}(F \otimes \Theta_2^*)F^{-1}y_p$.

According to (5.43), $F \otimes \Theta = F\Theta - \Theta F$. This means $F^{-1}(F \otimes \Theta)F^{-1} = \Theta F^{-1} - F^{-1}\Theta$. It follows from Assumption 5.2 that the coefficients of Θ_1^* and Θ_2^* are $(n-1)$ times differentiable and the derivatives are bounded. This suggests

$$\Theta F^{-1} - F^{-1}\Theta = \sum_{i=1}^{n-1} \{\bar{\theta}_i (s+\lambda_i)^{-1} - (s+\lambda_i)^{-1}\tilde{\theta}_i\}$$

where $\{\bar{\theta}_i\}_{i=1}^{n-1}$ and $\{\tilde{\theta}_i\}_{i=1}^{n-1}$ are bounded linear functions of $\{\theta_i\}_{i=0}^{n-2}$ and their derivatives. The above expression enables us to write (5.41). **Q.E.D.**

The most conservative way to stabilize (5.47) is to specify

$$u = r - sgn(e)(k_\theta \|\omega\| + m_b)$$

where $e = y_p - y_m$, $y_m = (s+a_m)^{-1} r$ and $k_\theta \ge \sup_t \|\theta^*\|$. Then the tracking error will satisfy

$$\dot{e} = -a_m e - sgn(e)(k_\theta \|\omega\| + m_b) - \omega^T \theta^* - \theta_g^*.$$

Consider a Lyapunov function candidate $v = 0.5e^2$. One can easily obtain $\dot{v} \le -a_m e^2$ which means the closed-loop system is asymptotically stable.

It should be emphasized that in general, a stable model reference control (MRC) of time-varying plants requires exact knowledge about the system parameters $A(s,t)$ and $B(s,t)$ [24]. But in the special case of relative degree 1 plants, the proposed MRC scheme ensures stable tracking provided that the upper bound of $\|\theta^*\|$ is available.

5.4.2 The effect of θ_g^*

We are now in a position to explain why this modification is used instead of the Tsakalis-Ioannou scheme given in [24]. In general, the T-I scheme is better than the standard MRC because it offers a perfect match when the plant coefficients are available. But there are some technical difficulties in parameter adaptation because of the replacement of the controller parameter θ. If we apply the T-I scheme to plants with relative degree 1, then the tracking error would look like

$$\dot{e} = -a_m e + X(s,t)\omega^T \phi$$

where $X(s,t)$ is a known LTV operator. The controller parameters now hide inside an integral operation $X(s,t)$. Tsakalis and Ioannou [24] proposed an adaptive law to adjust ϕ when the unknown parameter vector θ^* are slowly-varying, or $\theta^* = H(t)p$ where $H(t)$ is a known, fast varying matrix whereas p is an unkown, slowly varying vector.

When the relative degree is higher than 1, Taskalis and Ioannou's algorithm sees to be the only way out. However, when the relative degree is 1, we can propose a new adaptive law w! ch directly updates θ without the slowly-varying assumption.

For plants with relative degree 1, the ideal controller parameter θ^* is a linear vector function of $\{a_i(t)\}_{i=0}^{n-1}$, $\{b_j(t)\}_{j=0}^{n-1}$ and the derivatives of $a_{n-1}(t)$ and $b_{n-1}(t)$. If we have some "structural" knowledge [24] about the plant coefficients, then the structure of θ^* must be almost the same as that of $A(s,t)$ and $B(s,t)$ as (5.45) and (5.46) suggest. In case $\theta^* = H(t)\zeta^*$ where $H(t)$ is a known time-varying matrix and ζ^* an unknown constant vector, we can specify $r' = r - sgn(e)(k_\zeta\|\zeta\| + m_b)$, or equivalently

$$u = r + \omega^T H(t)\zeta - sgn(e)(k_\zeta\|\zeta\| + m_b)$$

where ζ is the estimate of ζ^*. Consequently, the tracking error will satisfy

$$\dot{e} = -a_m e + \omega^T H(t)\Delta\zeta - \theta_g^* - sgn(e)(m_b + k_\zeta\|\zeta\|)$$

where $\Delta\zeta = \zeta - \zeta^*$. Consider now a Lyapunov function candidate $v = \frac{1}{2}(e^2 + \|\Delta\zeta\|^2)$. An $|e|$-modified adaptation law [15] $\dot{\zeta} = -|e|\zeta - eH(t)^T\omega^T$ will lead to

$$\dot{v} \leq -|e|(a_m|e| + \|\zeta\|^2)$$

provided that $k_\zeta \geq \|\zeta^*\|$. As a result, the system is asymptotically stable without persistent excitation.

When no *à priori* knowledge about θ_g^* is available, one can control the system via a normalized signal $a^*(t) = \theta_g^*/m_b$. It is not difficult to see that a^* and $\dot{a}^*$ are bounded. If we specify

$$r' = r + \theta_g = r + a(t)m_b, \tag{5.48}$$

then the control law is given by

$$u = r + \omega^T\theta + a(t)m_b. \tag{5.49}$$

And the tracking error satisfies

$$\dot{e} = -a_m e + \phi^T\omega + \phi_a m_b = -a_m e + \bar{\phi}^T\bar{\omega} \tag{5.50}$$

where $\phi_a = a(t) - a^*(t)$, $\bar{\phi}^T = [\phi^T, \phi_a]$ and $\bar{\omega}^T = [\omega^T, m_b]$. The adaptive law is described by

$$\dot{\bar{\theta}} = -\sigma\bar{\theta} - \frac{e}{\alpha}\bar{\omega} \qquad \text{or} \qquad \dot{\bar{\phi}} = -\sigma\bar{\phi} - \frac{e}{\alpha}\bar{\omega} - \bar{\rho} \tag{5.51}$$

where $\bar{\rho} = \sigma\bar{\theta}^* + \dot{\bar{\theta}}^*$ and $\bar{\theta}^{*T} = [\theta^{*T}, a^*]$.

Lemma 5.2 *The closed-loop adaptive control system (5.50) and (5.51) is uniformly ultimately bounded.* (u.u.b.).

Proof: Considering a positive definite function $v = \frac{1}{2}(e^2 + \alpha\bar{\phi}^T\bar{\phi})$, and evaluating $\dot{v}$ along a closed-loop system (5.50) and (5.51), we obtain

$$\dot{v} \leq -a_m e^2 - \alpha\sigma\|\bar{\phi} + \frac{\bar{\rho}}{2\sigma}\|^2 + \alpha\bar{\gamma} \tag{5.52}$$

where $\bar{\gamma} = \sup_t\{\frac{1}{4\sigma}\|\bar{\rho}\|^2\}$. The above inequality implies the existence of v^* such that $\dot{v} < 0$ whenever $v > v^*$. Thus v and all signals are uniformly ultimately

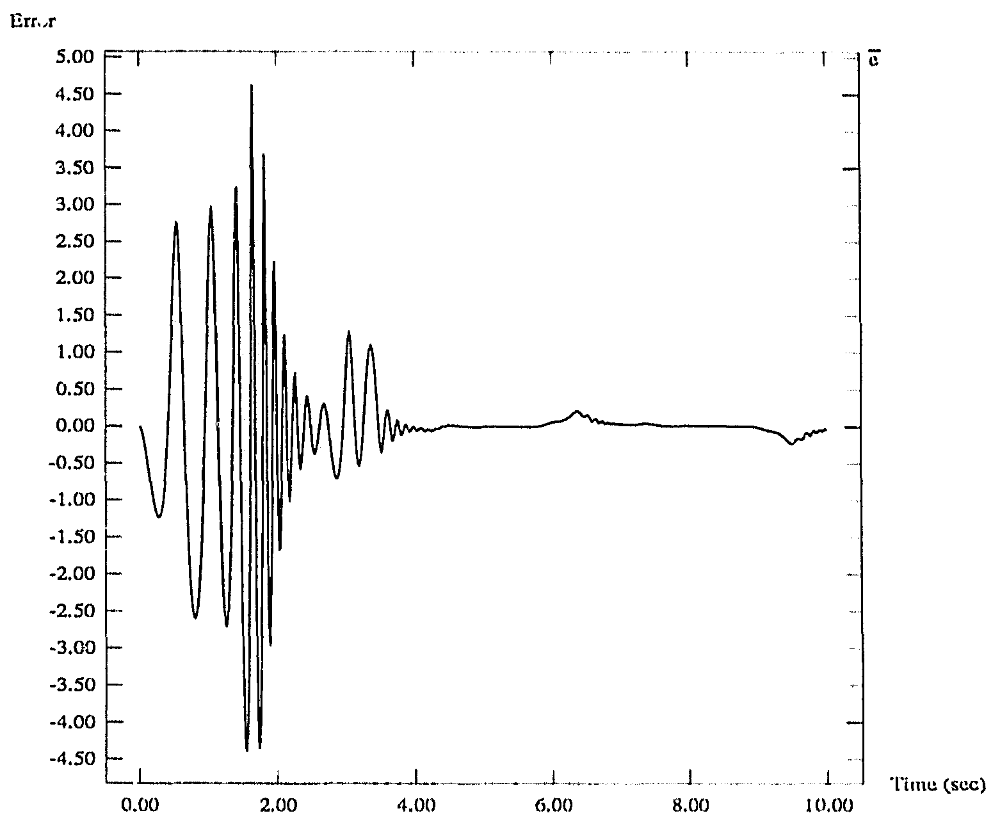

Figure 5.7: Tracking error of a LTV plant with θ_g.

bounded. The system will reach a region where $\dot{v} \geq 0$ in finite time which is an elliptic ball suggested by (5.52). **Q.E.D.**

In Lemma 5.2, we do not require any *à priori* knowledge about the "ideal" controller parameter vector θ^*. It can be fast-varying as long as $\|\dot{\theta}^*\|$ is bounded. The σ-modified adaptation law will catch up to the changing parameter and force all signals to converge into an elliptic ball. As suggested by (5.52), the tracking error can be controlled by adjusting a_m and α.

A simulation experiment is conducted to test the adaptive system. The plant dynamics and the reference model are given by

$$(s^2 - 6s + 2\sin(4t))y_p = [s(2 + \sin(4t)) + 1]u$$

$$(s + 10)y_m = 100\sin(t)$$

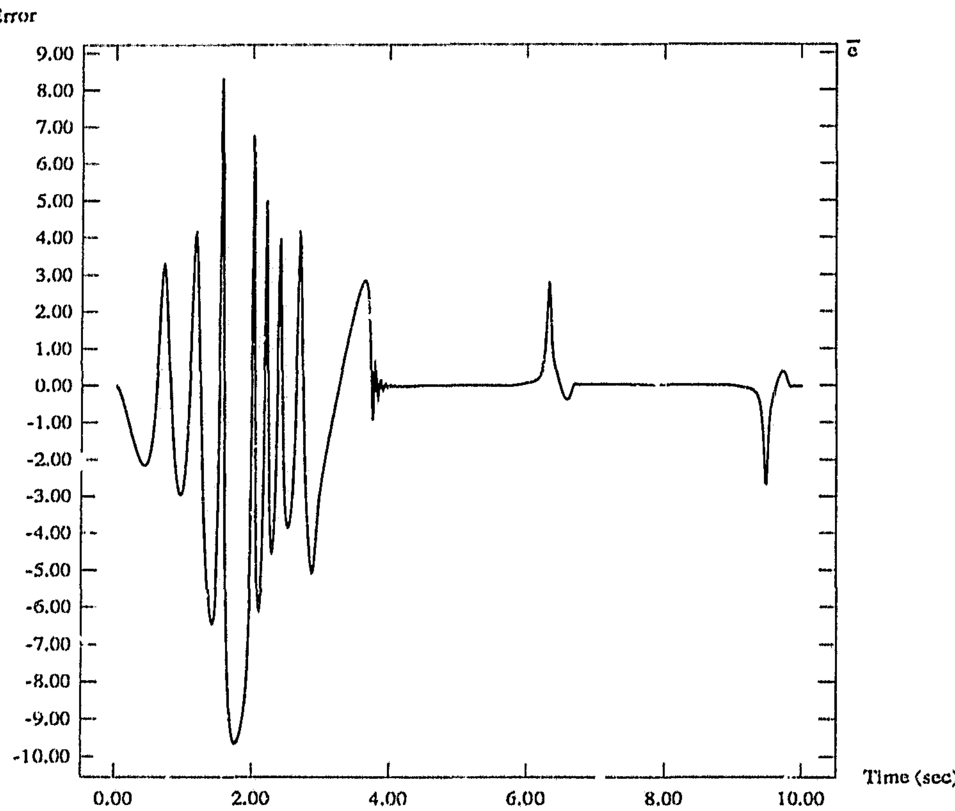

Figure 5.8: Tracking error of a LTV plant without θ_g.

respectively. (5.39) and (5.51) are implemented to control the system. In this case, θ_g is an additional adaptive signal intended to cancel the effect of θ_g^*. (The normalizing signal m_b is replaced by 1 in this case.) The adaptive vector is $\bar{\theta}^T = [\psi_1, \theta_1, \theta_2, \psi_2, \theta_g]$ while the feedback vector $\bar{\omega}^T = [u, (s+1)^{-1}u, (s+1)^{-1}y_p, y_p, 1]$. We choose $\alpha_{\psi_1}^{-1} = .01$ for ψ_1, $\alpha_g^{-1} = 500$ for θ_g and $\alpha^{-1} = 100$ for the rest of the signals. A leakage factor $\sigma = 1$ is uniformly assigned to all adaptive signals. In Assumption 5.2 we assume that the sign of $B(s,t)$ is known and positive. Thus we can set the initial value of $\psi_1 = -100$ such that $(1-\psi_1)^{-1}$ is a small value at the beginning of the adaptation. The initial values of all other adaptive signals are set to be zero.

The tracking performance is plotted in Fig.5.7 where $|e| \leq .25$, about 2.5

percent of the model output. Fig.5.8 plots the result without the compensation signal θ_g. All other conditions remain the same. But the tracking error becomes unacceptable.

5.5 Summary

The design procedures of adaptive controllers for LTV and NTV plants of relative degree one are presented in this chapter. Fisr order NTV/LTV plants are considered in Section 5.1. Section 5.2 extends the results to Multi-dimensional plants with full state feedbacks and Section 5.3 presents an example of applying the adaptive controller to robotic manipulators. LTV plants with relative degree one are studied in Section 5.4. These plants can be represented by a general first order state equation. Their corresponding reference models are all first order systems. The well known σ-modified adaptive law can be applied to such plants to adjust the adaptive coefficients while maintaining stable tracking.

Chapter 6

Adaptive Control of Robots with Uncertain Models

The adaptive controller developed in the previous chapter is intended for general LTV and NTV plants. While it can be applied to control robotic manipulators without any modifications as the simulation results demonstrate, its performance can be improved by incorporating some more information about the robot to be controlled. For a general n-link articulated mechanical structures like manipulators, there exist several important dynamic properties. These properties will improve the performance of an adaptive robotic controller if they are properly utilized. In this chapter, an improved adaptive controller is developed solely for articulated mechanical structures including robotic manipulators.

It should be emphasized that the dynamic properties to be explored in this chapter do not contain any specific knowledge about $M(q)$, $V(\dot{q}, q)$ and $G(q)$. Instead, they describe the relationship between $\dot{M}(q)$ and $V(\dot{q}, q)$ as well as some function bounds on $\dot{M}(q)$ $\dot{V}(\dot{q}, q)$ and $\dot{G}(q)$. Therefore the adaptive controller developed here belongs to the performance based group. Besides the parameter uncertainty, the new adaptive controller is designed to tolerate the effects of external disturbances τ_d in the robotic dynamic equation (1.1). Again, some function bound is imposed on the magnitude of τ_d which is proportional to the magnitudes of position and velocity errors.

Also addressed in this chapter is the possible effect of measurement noise introduced by the feedback of $\dot{q}$ and q. The new adaptive controller is proved to be stable under the effects of all three above-mentioned uncertainties and disturbances. The detailed derivation and analysis are given in order.

6.1 Some Dynamic Properties

Consider a system consist of N rigid bodies connected to each other and to the base. Let the system have n degrees-of-freedom and be described by a vector of generalized (Lagrangian) coordinate $q \in R^n$. Let v_i, $\omega_i \in R^3$ be the mass-center velocity and angular velocity of the i-th body respectively. Then, one can write

$$v_i = \phi_i(q)\dot{q} \qquad \omega_i = \varphi_i(q)\dot{q}$$

where ϕ_i, φ_i: $R^n \Rightarrow R^3$ are smooth and bounded functions of q.

With each body we will associate a Jacobian $J_i(q) \in R^{6\times n}$

$$J_i(q)\dot{q} = \begin{bmatrix} v_i \\ \omega_i \end{bmatrix} = \begin{bmatrix} \phi_i(q) \\ \varphi_i(q) \end{bmatrix} \dot{q}.$$

These Jacobians possess some useful general properties, derived below.

In a mechanical multi-body system, the Lagrangian coordinates can always be selected as relative displacements, linear or angular, with respect to some axes. Let $Z_j(q)$ be the unit vector of the axis related to coordinate q_j. Then, the vector of virtual displacement is given by

$$\Delta_j = Z_j \delta q_j.$$

Let $r_i \in R^3$ be the position vector of the i-th body mass center in the inertial coordinate frame. Then, the virtual change of r_i caused by the virtual displacement δq_j, denoting as $r_{ij}(q)$, is

$$\delta r_{ij}(q) = \Upsilon_{ij}(q)\Delta_j = \Upsilon_{ij}(q) Z_j \delta q_j; \qquad \Upsilon_{ij} \in R^{3\times 3}. \tag{6.1}$$

Lemma 6.1 *The transition matrix Υ_{ij} can be decomposed into a sum of a skew-symmetric and a diagonal matrix*

$$\Upsilon_{ij}(q) = S_{ij}(q) + c_{ij}(q)I, \qquad c_{ij}(q): \; R^n \Rightarrow R.$$

If $\delta r_{ij} \perp Z_j$, then Υ_{ij} becomes skew-symmetric, and if $\delta r_{ij} \| Z_j$, then Υ_{ij} becomes the diagonal 3×3 matrix.

Proof: Equation (6.1) can be written as

$$\delta r_{ij} = (Z_j \times p_{ij} + c_{ij} Z_j)\delta q_j = \Upsilon_{ij} Z_j \delta q_j \tag{6.2}$$

where $p_{ij} \in R^3$ is an unknown vector and $c_{ij}(q)$: $R^n \Rightarrow R$ an unknown function. The Eq.(6.2) clearly indicates that Υ_{ij} can be written as

$$\Upsilon_{ij} = S_{ij}(q) + c_{ij}I; \qquad S_{ij}(q) = \begin{bmatrix} 0 & (p_{ij})_x & -(p_{ij})_y \\ -(p_{ij})_x & 0 & (p_{ij})_z \\ (p_{ij})_y & -(p_{ij})_z & 0 \end{bmatrix}$$

where $S_{ij}(q)$ is a skew-symmetric matrix. The unknown function $c_{ij}(q)$ can be determined by

$$c_{ij}(q) = Z_j^T \delta r_{ij}.$$

It follows that

$$Z_j \times p_{ij} = \delta r_{ij} - c_{ij}(q) Z_j = s. \tag{6.3}$$

The last equation allows multiple solution of p_{ij}. Imposing an additional condition $p_{ij} \perp Z_j$, we can solve (6.3) for $p_{ij}(q)$.

Denoting, for simplicity, $p_{ij} = [x, y, z]^T$ and assume $z \neq 0$, we get

$$z = Z_{jy}s_x - Z_{jx}s_y; \qquad Z_{jz}y = Z_{jy}z - s_x; \qquad Z_{jz}x = Z_{jx}z + s_y.$$

Then

$$\Upsilon_{ij} = \begin{bmatrix} c_{ij} & x & -y \\ -x & c_{ij} & z \\ y & -z & c_{ij} \end{bmatrix}.$$

From (6.1) it follows that

$$v_{ij} = \Upsilon_{ij} Z_j \dot{q}_j \tag{6.4}$$

where v_{ij} may be considered as a virtual velocity of the i-th body mass-center caused by the virtual generalized velocity $\dot{q}_j$. Eq. (6.4) implies that

$$v_i = \sum_{j=1}^{n} \Upsilon_{ij} Z_j \dot{q}_j. \tag{6.5}$$

Q.E.D.

Since an infinitesimally small angular displacement can be represented by a vector, a similar formula can be derived for the angular velocities

$$\omega_{ij} = \Omega_{ij} Z_j \dot{q}_j \qquad \omega_i = \sum_{j=1}^{n} \Omega_{ij} Z_j \dot{q}_j. \tag{6.6}$$

Lemma 6.2 *Matrices Υ_i and Ω_{ij} have identical structure.*

Proof: Transition matrices Ω_{ij} for infinitesimally small angular displacements have the same structure as Υ_{ij}, except the vector δr_{ij} should be replaced by a vector representing this angular displacement. As was mentioned in the main text, matrices Υ_{ij} and Ω_{ij} become particularly simple in case of a open chain mechanism. **Q.E.D.**

Using Eqs. (6.5) and (6.6) the Jacobian can be written as

$$J_i(q) = \begin{bmatrix} \Upsilon_{i1} Z_1 & \Upsilon_{i2} Z_2 & \dots & \Upsilon_{in} Z_n \\ \Omega_{i1} Z_1 & \Omega_{i2} Z_2 & \dots & \Omega_{in} Z_n \end{bmatrix}.$$

In the particular case of an articulated open-loop mechanism (robotic manipulator) Υ_{ij} and Ω_{ij} have very simple structures. All Ω_{ij} become identity matrices (since $r_{ij} \perp Z_j$), and $\Upsilon_{ij} Z_j$ can be represented as $Z_j \times h_{ij}$, where h_{ij} is the vector from the j-th joint to the i-th mass-center. That is,

$$J_i(q) = \begin{bmatrix} Z_1 \times h_{i1} & \dots & Z_i \times h_{ii} & 0 & \dots & 0 \\ Z_1 & \dots & Z_i & 0 & \dots & 0 \end{bmatrix}.$$

Since $\phi_i(q)$ and $\varphi_i(q)$ are smooth functions of q, we can assume

$$|\frac{\partial^2 j_{ijk}(q)}{\partial q_n \partial q_m}| < \infty \tag{6.7}$$

where $j_{ijk}(q)$: $R^n \Rightarrow R$ is the jk-th entry of $J_i(q)$.

Using the Euler–Lagrange equation, one not only derives the dynamics of a class of nonlinear mechanical systems, but also arrives at the following useful properties about the system dynamics. These properties will be used in the design and stability analysis of adaptive controllers.

Property 6.1 *Let $\|X\|_F$ denote the Frobinious norm of matrix X. Then both $\|M(q)\|_F$ and $\|G(q)\|_F$ are uniformly bounded. Let $m_{jk}(q)$ be the jk-th entry of $M(q)$, then*

$$|\frac{\partial^2 m_{jk}}{\partial q_i \partial q_l}| < \infty \quad \forall\ q_i, q_l.$$

Proof: One can write

$$M(q) = \sum_{i=1}^{n} J_i^T(q) Q_i J_i(q) \quad \text{and} \quad G(q) = g \sum_{i=1}^{n} m_i J_i^T(q) h$$

where $h^T = [0,0,1,0,0,0]$; g is the gravity constant. The boundedness of $\|M(q)\|_F$ and $\|G(q)\|_F$ is self-evident from the above expression. The boundedness of $\frac{\partial^2 m_{ij}}{\partial q_n \partial q_m}$ follows equation (6.7). **Q.E.D.**

Property 6.2 *For a properly defined $V(\dot{q}, q)$ (this matrix does not have a unique form), the matrix $\dot{M}(q) - 2V(\dot{q}, q)$ is Skew-symmetric. In order words, the following identity holds*

$$x^T[\dot{M}(q) - 2V(\dot{q}, q)]x = 0 \ \ \forall \ \ x \neq 0, \ q, \dot{q}. \tag{6.8}$$

Proof: The k, jth element of $\dot{M}(q)$ is given by the chain rule as

$$\dot{m}_{kj} = \sum_{i=1}^{n} \frac{\partial m_{kj}}{\partial q_i} \dot{q}_i.$$

And the k,jth element of $\dot{M}(q) - 2V(\dot{q}, q)$ should be

$$\begin{aligned} \dot{m}_{kj} - 2v_{kj} &= \sum_{i=1}^{n}\{\frac{\partial m_{kj}}{\partial q_i} - [\frac{\partial m_{kj}}{\partial q_i} + \frac{\partial m_{ki}}{\partial q_j} - \frac{\partial m_{ij}}{\partial q_k}]\}\dot{q}_i \\ &= \sum_{i=1}^{n}[\frac{\partial m_{ij}}{\partial q_k} - \frac{\partial m_{ki}}{\partial q_j}]\dot{q}_i \end{aligned}$$

Since $M(q) = M^T(q)$, it follows that $m_{ij} = m_{ji}$. By interchanging the indices k and j in the above equation, one will find that $\dot{m}_{kj} - 2v_{kj} = 2v_{jk} - \dot{m}_{jk}$. Therefore the matrix $\dot{M} - 2V(\dot{q}, q)$ is skew symmetric and (6.8) is true. **Q.E.D.**

Property 6.3

$$V(\dot{q}, q)x = V_m(q)[\dot{q}x] \quad \forall\ x \in R^n$$

where $V_m(q) \in R^{n\times n^2}$ *is a uniformly bounded nonlinear matrix function and*

$$[\dot{q}x] = [\dot{q}_1 x_1, \ldots, \dot{q}_1 x_n,\ \ldots, \ldots,\ \dot{q}_n x_1, \ldots, \dot{q}_n x_n]^T \in R^{n^2}.$$

Proof: The ith component of vector $V(\dot{q}, q)x$ can be expressed as

$$v_i^T(\dot{q}, q)x = \sum_{j=1}^{n}\sum_{k=1}^{n} v_{ijk}(q)\dot{q}_j x_k$$

where $v_{ijk}(q)$ is given by

$$v_{ijk}(q) = \frac{1}{2}[\frac{\partial m_{kj}}{\partial q_i} + \frac{\partial m_{ki}}{\partial q_j} - \frac{\partial m_{ij}}{\partial q_k}].$$

According to property 1, $v_{ijk}(q)$ is uniformly bounded for all possible q. Let $l = jn + k$, then $v_{ijk}(q)$ is the i-lth entry of the matrix $V_M(q)$ while $\dot{q}_j x_k$ the lth component of vector $[\dot{q}x]$. It also follows that $\|V_M(q)\|_F$ is uniformly bounded. **Q.E.D.**

Property 6.4

$$V(\dot{q} + x, q) = V(x, q) + V(\dot{q}, q) \quad \forall\ x \in R^n, \tag{6.9}$$

and

$$\|V(x, q)\| < \infty \quad \text{if} \quad \|x\| < \infty. \tag{6.10}$$

Proof: The i-kth entry of $V(\dot{q}+x,q)$ is given by

$$v_{ik}(\dot{q}+x,q)=\sum_{j=1}^{n} v_{ijk}(q)(\dot{q}_j+x_j)$$

which implies (6.9) and (6.10). **Q.E.D.**

Property 6.5 $\|\dot{M}(q)\|_F \le c_M\|\dot{q}\|$, $\|\dot{V}_M(q)\| \le c_V\|\dot{q}\|$ *and* $\|\dot{G}(q)\| \le c_G\|\dot{q}\|$ *for some constants* $c_M>0$, $c_V>0$ *and* $c_G>0$.

Proof: Consider the i,kth entry of $M(q)$. Its time derivative can be expressed as $\dot{m}_{ij}(q)=\dot{q}^T\nabla_q m_{ij}(q)$. According to property 1, $\|\nabla_q m_{ik}(q)\|$ is bounded. Therefore

$$\|\dot{M}(q)\|_F=\sqrt{\sum_{i=1}^{n}\sum_{k=1}^{n}\dot{m}_{ik}^2}\le\|\dot{q}\|\sqrt{\sum_{i=1}^{n}\sum_{k=1}^{n}\|\nabla_q m_{ik}\|^2}\le c_M\|\dot{q}\|.$$

The same argument applies to the cases of $\dot{V}_M(q)$ and $\dot{G}(q)$. **Q.E.D**

6.2 The Adaptive Controller Design

Introduce variables $s=\dot{e}+\Lambda e$ and $\psi=\dot{q}_d-\Lambda e$, where $e=q-q_d$; $\Lambda=\Lambda^T>0$. The control law is given by

$$\tau=-Ks+A_M\dot{\psi}+A_V[\dot{q}\psi]+A_G \tag{6.11}$$

where $[\dot{q}\psi]^T=[\dot{q}_1\psi_1\ \dots\ \dot{q}_1\psi_n\ \dot{q}_2\psi_1\ \dots\ \dot{q}_2\psi_n\ \ \dots\ \ \dot{q}_n\psi_1\ \dots\ \dot{q}_n\psi_n]\in R^{n^2}$; $K=K^T>0$; the adaptive matrices $A_M\in R^{n\times n}$, $A_V\in R^{n\times n^2}$ and $A_G\in R^{n\times 1}$ are intended to compensate the nonlinear dynamic terms $M(q)$, $V_m(q)$ and $G(q)$ respectively. According to Properties 1 and 3, $M(q)$, $V_m(q)$ and $G(q)$ are all uniformly bounded. The vector $A_V[\dot{q}\psi]$ introduced in accordance with Property 3 ($V(\dot{q},q)\psi=V_m(q)[\dot{q}\psi]$) plays an important role in the latter stability analysis.

Let $A=[A_M,A_V,A_G]$ and $\zeta^T=[\dot{\psi}^T,[\dot{q}\psi]^T,1]$. Then the control and adaptive laws can be written as

$$\tau=-Ks+A\zeta, \tag{6.12}$$

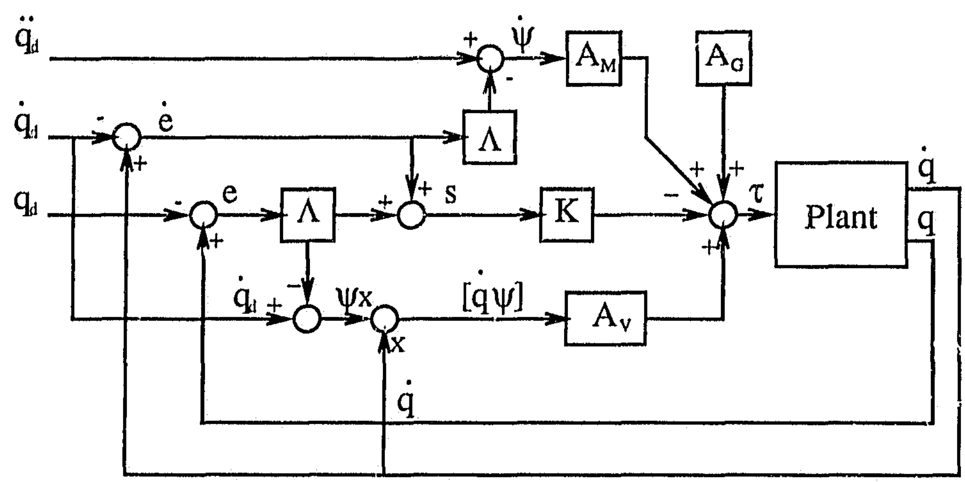

Figure 6.1: Block diagram of the control system.

$$\dot{A} + \sigma A = -\frac{1}{\alpha} s \zeta^T \tag{6.13}$$

where $\sigma = \sigma_0 + \sigma_1 \|\dot{q}\|$; σ_0, σ_1 and α are positive constants. The adaptive controller only requires position and velocity feedbacks. A block diagram of the closed-loop adaptive control system is given in figure 6.1.

Remarks: The control law (6.11) is somewhat similar to that of Slotine and Li [39]. There are two important differences. First we have introduced a new vector of control variables $[\dot{q}\psi]$. With this vector, the adaptive matrices A_M, A_V, and A_G are no longer required to compensate for potentially unbounded terms. Secondly, the adaptive matrices in (6.11) depend on the tracking error and are determined by the adaptive law, while in [39] the corresponding matrix coefficients represent estimates of matrices $M(q)$, $V(\dot{q}, q)$, and $G(q)$ of the plant dynamic model. The adaptive controller by Slotine and Li belongs to the "model-based" [36] methods. The modeled dynamic, in the sense defined in the introduction, should be completely known, and should be updated on-line during the plant motion. It is a very computational intensive process for a plant with many degree-of-freedoms. In contrary, the law (6.11) implements so called "performance-based" methods [36]. The adaptive matrices in (6.11) don't explicitly depend on the system dynamics, and the adaptation law, (6.13), is very simple.

Two other works on performance-based adaptive control, Lin [35], and Seraji [36], use quite different control laws. The corresponding adaptive matrices from their works should compensate for potentially unbounded terms that depend on robot joint velocities and even accelerations ($C(x^p, x^v)$ in [35] and $d_i(q, \dot{q}, \ddot{q})$ in [36]). As the result, the stability analysis in these works relies on a "slowly-varying" assumption regarding the uncertain terms of the dynamic model. Introducing vector $[\dot{q}\psi]$ we relax this assumption. While the dynamic model has potentially unbounded terms, the adaptive matrices in (6.11) are required to compensate for bounded parts of these terms. We have achieved that, in particular, by increasing the number of controller variables by the n^2-vector $[\dot{q}\psi]$. However, the adaptive law (6.13) for updating corresponding entries of A_V is very simple and suitable for parallel processing, and we have proved that the increasing in number of controller variables does not produce any negative effect on stability and robustness of the controlled system.

The adaptive law (6.13) has similar structure as the σ-modification law by Reed and Ioannou [9]. They differ in modification factor σ. In the proposed adaptation law (6.13), the factor $\sigma = \sigma_0 + \sigma_1||\dot{q}||$ varies according to the magnitude of the joint velocity. This enables the adaptive quantities A_M, A_V and A_G to trace $M(q)$, $V_m(q)$ and $G(q)$ with a proper speed. As a result, the stability analysis no longer relies on the "slowly varying" assumption.

6.3 Stability Analysis

The stability analysis is divided into two phases. First, the unmodeled dynamics τ_d are considered without any measurement noise in the feedback circuits. The proposed adaptive controller is proven to be able to control the robot such that the tracking errors are kept within a computable tolerance. Then, a similar but more involved analysis is conducted for the cases where the feedback channels are contaminated by some bounded noises. In both cases, the closed-loop is stable as long as the feedback gain K is sufficiently high.

6.3.1 Robustness with respect to unmodeled dynamics

Combining (1.1) and (6.12) leads to

$$M(q)\ddot{q} + V(\dot{q}, q)\dot{q} + G(q) + \tau_d = -Ks + A_M\dot{\psi} + A_V[\dot{q}\psi] + A_G.$$

Subtracting $M(q)\dot{\psi} + V(\dot{q}, q)\psi + G(q) = M(q)\dot{\psi} + V_M(q)[\dot{q}\psi] + G(q)$ from both sides of the above expression, and using the fact that $s = \dot{q} - \psi$, one can write

$$\begin{aligned} M(q)\dot{s} &= -[K + V(\dot{q}, q)]s + \Phi_M\dot{\psi} + \Phi_V[\dot{q}\psi] + \Phi_G - \tau_d \\ &= -[K + V(\dot{q}, q)]s + \Phi\zeta - \tau_d \end{aligned} \tag{6.14}$$

where $\Phi = [\Phi_M, \Phi_V, \Phi_M]$, $\Phi_M = A_M - M(q)$, $\Phi_V = A_V - V_m(q)$ and $\Phi_G = A_G - G(q)$. The above equation represents the trajectory of a robot (1.1) controlled by (6.12). The unmodeled dynamics and the external disturbances are assumed to be bounded by

$$\|\tau_d\| \leq d_0 + d_1\|\dot{e}\| + d_2\|e\| \tag{6.15}$$

where $d_0 > 0$, $d_1 > 0$ and $d_2 > 0$ are some constants.

Theorem 6.1 *For the closed-loop system (6.13) and (6.14), the tracking errors $\dot{e}$, e and the compensation error Φ are uniformly ultimately bounded if the feedback gain K is sufficiently large such that λ_k, the smallest eigenvalue of $K = K^T > 0$, satisfies*

$$\lambda_k > d_1 + 0.5(d_2 + d_1\|\Lambda\|) \tag{6.16}$$

$$\lambda_k\lambda_\Lambda^2 > d_2\|\Lambda\| + 0.5(d_2 + d_1\|\Lambda\|) \tag{6.17}$$

where $\|\Lambda\|$ denotes the induced norm of $\Lambda = \Lambda^T > 0$ and λ_Λ the smallest eigenvalue of Λ.

Proof: Consider a positive definite function

$$L = \frac{1}{2}\{s^T M(q)s + \alpha Tr[\Phi\Phi^T] + e^T(K\Lambda + \Lambda K)e\}.$$

Its time derivative $\dot{L}$ is evaluated along (6.14) as

$$\dot{L} = -\dot{e}^T K\dot{e} - e^T\Lambda K\Lambda e - s^T\tau_d + Tr\{(s\zeta^T + \alpha\dot{\Phi})\Phi^T\} \tag{6.18}$$

where the identity (6.8) has been substituted to cancel out $s^T\dot{M}(q)s$; another identity $x^Ty = Tr\{xy^T\}$ has been applied to get $s^T\Phi\zeta = Tr\{s\zeta^T\Phi^T\}$ and hence the last term of the above equation.

With adaptive law (6.13), one can verify the following relation

$$s\zeta^T + \alpha\dot{\Phi} = -\alpha(\Gamma + \sigma\Phi). \tag{6.19}$$

where $\Gamma = [\Gamma_M, \Gamma_V, \Gamma_G]$ and

$$\Gamma_M = \sigma M + \dot{M}, \quad \Gamma_V = \sigma V_m + \dot{V}_m \quad \Gamma_G = \sigma G + \dot{G}. \tag{6.20}$$

Substituting (6.19) and identity $Tr[XX^T] = \|X\|_F^2$ into (6.18) leads to

$$\dot{L} = -\dot{e}^T K\dot{e} - e^T\Lambda K\Lambda e - s^T\tau_d - \alpha\sigma\|\Phi + \frac{\Gamma}{2\sigma}\|_F^2 + \frac{\alpha}{4\sigma}\|\Gamma\|_F^2 \tag{6.21}$$

According to (6.20), one can write

$$\|\frac{\Gamma}{\sigma}\|_F^2 \leq (\|[M(q), V_m(q), G(q)]\|_F + \|\frac{[\dot{M}, \dot{V}_m, \dot{G}]}{\sigma_0 + \sigma_1\|\dot{q}\|}\|_F)^2 \leq (\rho + \beta)^2 \tag{6.22}$$

where by Properties 6.1, 6.3 and 6.5 (proved in Appendix A),

$$\rho = \sup_q \|[M(q), V_m(q), G(q)]\|_F \quad \text{and} \quad \beta = \sup_{\dot{q}, q}\{\|\frac{[\dot{M}, \dot{V}_m, \dot{G}]}{\sigma_0 + \sigma_1\|\dot{q}\|}\|_F\}.$$

The fact that $\|\dot{q}\| \leq \|\dot{q}_d\|_{max} + \|\dot{e}\|$ enables one to write

$$\frac{\alpha}{4\sigma}\|\Gamma\|_F^2 \leq \frac{1}{4}\alpha\sigma(\rho + \beta)^2 \leq \gamma_0 + \gamma_1\|\dot{e}\| \tag{6.23}$$

where

$$\gamma_0 = \frac{\alpha}{4}(\rho + \beta)^2(\sigma_0 + \sigma_1\|\dot{q}_d\|_{max}) \quad \text{and} \quad \gamma_1 = \frac{\alpha}{4}(\rho + \beta)^2\sigma_1. \tag{6.24}$$

Substituting (6.23) into (6.21) results in

$$\dot{L} \leq -\lambda_k\|\dot{e}\|^2 - \lambda_k e^T\Lambda^2 e - s^T\tau_d - \alpha\sigma\|\Phi + \frac{\Gamma}{2\sigma}\|_F^2 + \gamma_0 + \gamma_1\|\dot{e}\| \qquad (6.25)$$

where λ_k is the smallest eigenvalue of K.

Pause: Intuitively, one can sense the design strategy here: If K is sufficiently large, the two negative quadratic terms $-\lambda_k\|\dot{e}\|^2$ and $-\lambda_k e^T\Lambda^2 e$ in (6.25) will dominate all other possible terms of $\|\dot{e}\|$, $\|e\|$ and $\|\dot{e}\|\|e\|$ whose coefficients are bounded by some positive constants. Thus $\dot{L} < 0$ when any one of $\|\dot{e}\|$, $\|e\|$ or $\|\Phi\|$ exceeds a certain level. The detailed analysis are given as follows.

According to (6.15), one can write

$$\begin{aligned} -s^T\tau_d &\leq (\|\dot{e}\| + \|\Lambda\|\|e\|)(d_0 + d_1\|\dot{e}\| + d_2\|e\|) \\ &= d_0\|\dot{e}\| + d_1\|\dot{e}\|^2 + c_0\|e\| + c_1\|e\|^2 + c_2\|e\|\|\dot{e}\| \end{aligned}$$

where $\|\Lambda\|$ denotes the induced norm of matrix Λ; $c_0 = d_0\|\Lambda\|$; $c_1 = d_2\|\Lambda\|$ and $c_2 = d_2 + d_1\|\Lambda\|$. It then follows that

$$\dot{L} \leq -a_1\|\dot{e}\|^2 - a_2\|e\|^2 + c_2\|\dot{e}\|\|e\| + a_3\|\dot{e}\| + c_0\|e\| - \alpha\sigma\|\Phi + \frac{\Gamma}{2\sigma}\|_F^2 + \gamma_0 \quad (6.26)$$

where $a_1 = \lambda_k - d_1$; $a_2 = \lambda_k\lambda_\Lambda^2 - c_1$; $a_3 = d_0 + \gamma_1$ and λ_Λ denotes the smallest eigenvalue of matrix Λ. A further manipulation of (6.26) leads to

$$\begin{aligned} \dot{L} &\leq -\frac{c_2}{2}(\|\dot{e}\| - \|e\|)^2 - p_1(\|\dot{e}\| - \frac{a_3}{2p_1})^2 - p_2(\|e\| - \frac{c_0}{2p_2})^2 - \alpha\sigma\|\Phi + \frac{\Gamma}{2\sigma}\|^2 + \gamma \\ &\leq -p_1(\|\dot{e}\| - \frac{a_3}{2p_1})^2 - p_2(\|e\| - \frac{c_0}{2p_2})^2 - \alpha\sigma(\|\Phi\|_F - \frac{\rho+\beta}{2})^2 + \gamma \qquad (6.27) \end{aligned}$$

where (6.22) has been substituted to eliminate the nonlinear function Γ; according to (6.16) and (6.17), λ_k is sufficiently large such that

$$\begin{aligned} p_1 &= a_1 - \frac{c_2}{2} = \lambda_k - d_1 - 0.5(d_2 + d_1\|\Lambda\|) > 0 \\ p_2 &= a_2 - \frac{c_2}{2} = \lambda_k\lambda_\Lambda^2 - d_2\|\Lambda\| - 0.5(d_2 + d_1\|\Lambda\|) > 0 \\ \gamma &= \gamma_0 + \frac{1}{4}(\frac{(d_0+\gamma_1)^2}{p_1} + \frac{d_0^2\|\Lambda\|^2}{p_2}). \end{aligned}$$

Inequality (6.27) defines an ellipsoid in a 3-D Euclidian space $\|\dot{e}\|$-$\|e\|$-$\|\Phi\|$ whose inside contains the set $\{\dot{e}, e, \Phi \mid \dot{L} \geq 0\}$. In order words, $\dot{L} < 0$ whenever the norms of tracking or compensation errors exceed this ellipsoid. Thus there must exists a constant L^* such that $\dot{L} < 0$ whenever $L > L^*$, hinting that L is uniformly ultimately bounded. **Q.E.D.**

Theorem 6.2 *The tracking errors and compensation error of the closed-loop system (6.13) and (6.14) are ultimately bounded by ellipsoid*

$$\lim_{t\to\infty}\{p_1(\|\dot{e}\| - \frac{d_0+\gamma_1}{2p_1})^2 + p_2(\|e\| - \frac{d_0\|\Lambda\|}{2p_2})^2 + \alpha\sigma(\|\Phi\|_F - \frac{\rho+\beta}{2})^2\} \leq \gamma. \quad (6.28)$$

Proof: Introduce

$$\begin{aligned} f_1(t) &= \dot{e}^T K \dot{e} + e^T \Lambda K \Lambda e + \alpha\sigma\|\Phi + \frac{1}{2\sigma}\Gamma\|_F^2, \\ f_2(t) &= \frac{\alpha}{4\sigma}\|\Gamma\|_F^2 - s^T\tau_d. \end{aligned}$$

Then (6.21) can be written as $\dot{L} = f_2(t) - f_1(t)$, or

$$L(t) = L(0) + F_2(t) - F_1(t)$$

where $F_1(t) = \int_0^t f_1(t)dt$, $F_2(t) = \int_0^t f_2(t)dt$ and $L(0)$ the initial value of L.

Since $f_1(t) \geq 0$, $F_1(t)$ must either converge to some constant or diverge to ∞. In the first case, $F_1(t) \to F^*$, then $f_1(t) \to 0$. (6.28) will be satisfied; In the second case $F_1(t) \to \infty$, we have

$$\lim_{t\to\infty}\frac{L(t)}{F_1(t)} = \lim_{t\to\infty}\frac{F_2(t)}{F_1(t)} - 1 = \lim_{t\to\infty}\frac{f_2(t)}{f_1(t)} - 1 = 0.$$

It implies that

$$\lim_{t\to\infty}\dot{L} = \lim_{t\to\infty}[f_2(t) - f_1(t)] = 0.$$

A substitution of the above result into (6.27) leads to (6.28). **Q.E.D.**

According to Theorems 6.1 and 6.2, the tracking error can be reduced arbitrarily by increasing K alone. The impulse response of the closed-loop system will not be affect by the high feedback gain, because the intermediate vector $s = \dot{e} + \Lambda e$ determines the transient response of the system. By increasing K, the whole feedback of Ks will be increased. When $\|s\|$ is bounded, the tracking error can be shown to satisfy a first order linear differential equation

$$s = \dot{e} + \Lambda e = \nu$$

where ν represents a bounded nonlinear disturbance. No matter what value of K is assigned, the transient behavior of the system is actually determined by Λ, which is assigned as a positive definite matrix. By altering the eigenvalues of Λ, one can adjust the transient response of the closed-loop system. Simulation results on the effect of K will be given later on in this chapter.

6.3.2 Robustness with respect to measurement noises

A measurement noise is unavoidable in the feedback circuits. In spite of relatively small magnitude, the noise effect may be distorting, since noise introduces errors directly in the adaptation law. Let n_v and n_p denote the noise vectors introduced by measuring $\dot{q}$ and q. Since the joint velocity could be measured from a different channel rather than taking the numerical differentiation of q, n_v is not necessarily the derivative of n_p.

Because of the noise, the control and adaptation laws have to use the noisy state feedbacks to calculate the control torque and update the controller coefficients. It follows that

$$\begin{aligned} \tau &= -Ks_n + A_M\dot{\psi}_n + A_V[\dot{q}_n\psi_n] + A_G \\ &= -Ks_n + A\zeta_n \qquad (6.29) \\ \dot{A} &= -\sigma_n A - \frac{1}{\alpha}s_n\zeta_n^T \qquad (6.30) \end{aligned}$$

where a subscript "n" is used to emphasize that the corresponding state variables contain some noisy effect. For example, $\dot{q}_n = \dot{q} + n_v$, $s_n = s + n_v + \Lambda n_p$, $\psi_n = \dot{q}_d - \Lambda(e + n_p)$ and $\dot{\psi}_n = \ddot{q}_d - \Lambda(\dot{e} + \cdot)$, *etc.* Consequently, ζ_n is a noisy version of ζ computed from noisy feedback states $\dot{q}_n$ and q_n.

Lemma 6.3 *The trajectory of robot (1.1) controlled by (6.29) is described by*

$$M(q)\dot{s}_n = -[K + V(\dot{q}, q)]s_n + \Phi\zeta_n - \tilde{\tau}_d \tag{6.31}$$

where

$$\tilde{\tau}_d = \tau_d - M(q)\dot{n}_v - V(n_v, q)\dot{q}_d - V(\dot{q}, q)n_v + V(n_v, q)\Lambda e_n$$

which is bounded by

$$\|\tilde{\tau}_d\| \leq \tilde{d}_0 + \tilde{d}_1\|\dot{e}_n\| + \tilde{d}_2\|e_n\| \tag{6.32}$$

for some constants $\tilde{d}_0 > 0$, $\tilde{d}_1 > 0$ *and* $\tilde{d}_2 > 0$ *if* $\|\dot{n}_v\|$, $\|n_v\|$ *and* $\|n_p\|$ *are bounded.*

Proof: Combining (1.1) and (6.29) leads to

$$M(q)\ddot{q} + V(\dot{q}, q)\dot{q} + G(q) + \tau_d = -Ks_n + A_M\dot{\psi}_n + A_V[\dot{q}_n\psi_n] + A_G.$$

Substituting $M(q)\ddot{q} = M(q)\ddot{q}_n - M(q)\dot{n}_v$ into the above equation results in

$$M(q)\ddot{q}_n + V(\dot{q}, q)\dot{q} + G(q) + \tau_1 = -Ks_n + A_M\dot{\psi}_n + A_V[\dot{q}_n\psi_n] + A_G \tag{6.33}$$

where $\tau_1 = \tau_d - M(q)\dot{n}_v$. A subtraction of

$$M(q)\dot{\psi}_n + V(\dot{q}_n, q)\psi_n + G(q) = M(q)\dot{\psi}_n + V_M(q)[\dot{q}_n\psi_n] + G(q)$$

from both sides of (6.33) leads to

$$\begin{aligned} M(q)\dot{s}_n + V(\dot{q}, q)\dot{q} - V(\dot{q}_n, q)\psi_n + \tau_1 &= -Ks_n + \Phi_M\dot{\psi}_n + \Phi_V[\dot{q}_n\psi_n] + \Phi_G \\ &= -Ks_n + \Phi\zeta_n \end{aligned} \tag{6.34}$$

where the relation $\dot{s}_n = \ddot{q}_n - \dot{\psi}_n$ has been substituted.

According to Property 6.4, one can write

$$V(\dot{q},q)\dot{q} = V(\dot{q},q)\dot{q}_n - V(\dot{q},q)n_v, \tag{6.35}$$

$$V(\dot{q}_n,q)\psi_n = V(\dot{q},q)\psi_n + V(n_v,q)\psi_n. \tag{6.36}$$

This enables one to rewrite (6.34) as

$$M(q)\dot{s}_n + V(\dot{q},q)s_n + \tilde{\tau}_d = -Ks_n + \Phi\zeta_n \tag{6.37}$$

where

$$\begin{aligned}\tilde{\tau}_d &= \tau_1 - V(\dot{q},q)n_v - V(n_v,q)\psi_n \\ &= \tau_d - M(q)\dot{n}_v - V(\dot{q},q)n_v - V(n_v,q)\psi_n \\ &= \tau_d - M(q)\dot{n}_v - V(\dot{q},q)n_v - V(n_v,q)\dot{q}_d + V(n_v,q)\Lambda e_n.\end{aligned}$$

In the last equation, $M(q)\dot{n}_v$ and $V(n_v,q)\dot{q}_d$ are uniformly bounded because $\dot{n}_v$, n_v and $\dot{q}_d$ are bounded by assumption while Properties 6.1 and 6.4 apply to these cases. The remaining terms can be conveniently bounded in terms of $\|\dot{e}_n\|$ and $\|e_n\|$ because $\|\dot{q}\| \leq \|\dot{e}_n\| + \|\dot{q}_d - n_v\|$ and $\|\dot{e}\| \leq \|\dot{e}_n\| + \|n_v\|$. Thus inequality (6.32) holds. Equation (6.37) is another form of (6.31). **Q.E.D.**

It is important to notice the resemblance between (6.30) and (6.13); (6.31) and (6.14); (6.32) and (6.15). The major difference is the subscript "n" which indicates that the corresponding vector contains some bounded additive noise. Some effects of the measurement noise can be added to the equivalent disturbance $\tilde{\tau}_d$. Lemma 6.3 proves that these effects can at most grow as fast as $\|\dot{e}_n\|$ and $\|e_n\|$. The conditions in this case become very similar to the noise-free case. Instead of trying to prove that $\dot{e}$, e and Φ are bounded, one may prove that $\dot{e}_n$, e_n and Φ are bounded. The boundedness of $\dot{e}$ and e follows from the fact that n_v and n_p are pre-assumed to be bounded. Now, the following result becomes straightforward.

Theorem 6.3 *For the closed-loop system (6.30) and (6.31), $\dot{e}_n$, e_n and Φ are uniformly ultimately bounded if the measurement errors $\dot{n}_v$, n_v and n_p are bounded* *such that (6.32) holds; the feedback gain K is sufficiently large such that λ_k, the smallest eigenvalue of $K = K^T > 0$, satisfies*

$$\lambda_k > \tilde{d}_1 + 0.5(\tilde{d}_2 + \tilde{d}_1\|\Lambda\|) \quad (6.38)$$

$$\lambda_k\lambda_\Lambda^2 > \tilde{d}_2\|\Lambda\| + 0.5(\tilde{d}_2 + \tilde{d}_1\|\Lambda\|). \quad (6.39)$$

Proof: Consider a positive definite function

$$L_n = \frac{1}{2}\{s_n^T M(q)s_n + \alpha Tr[\Phi\Phi^T] + e_n^T(K\Lambda + \Lambda K)e_n\}.$$

Its time derivative $\dot{L}_n$ is evaluated along (6.14) as

$$\dot{L}_n = -\dot{e}_n^T K\dot{e}_n - e_n^T\Lambda K\Lambda e_n - s_n^T\tilde{\tau}_d + Tr\{(s_n\zeta_n^T + \alpha\dot{\Phi})\Phi^T\} \quad (6.40)$$

where the identity (6.8) has been substituted to cancel out $s_n^T\dot{M}(q)s_n$; another identity $x^Ty = Tr\{xy^T\}$ has been applied to get $s_n^T\Phi\zeta_n = Tr\{s_n\zeta_n^T\Phi^T\}$ and hence the last term of the above equation.

With adaptive law (6.30), one can verify the following relation

$$s_n\zeta_n^T + \alpha\dot{\Phi} = -\alpha(\Gamma + \sigma_n\Phi) \quad (6.41)$$

where $\Gamma = [\Gamma_M, \Gamma_V, \Gamma_G]$ and

$$\Gamma_M = \sigma_n M + \dot{M}, \quad \Gamma_V = \sigma_n V_m + \dot{V}_m \quad \Gamma_G = \sigma_n G + \dot{G}. \quad (6.42)$$

Substituting (6.41) and identity $Tr[XX^T] = \|X\|_F^2$ into (6.40) leads to

$$\dot{L}_n = -\dot{e}_n^T K\dot{e}_n - e_n^T\Lambda K\Lambda e_n - s_n^T\tilde{\tau}_d - \alpha\sigma_n\|\Phi + \frac{\Gamma}{2\sigma_n}\|_F^2 + \frac{\alpha}{4\sigma_n}\|\Gamma\|_F^2 \quad (6.43)$$

According to (6.42), one can write

$$\|\frac{\Gamma}{\sigma_n}\|_F^2 \leq (\|[M(q), V_m(q), G(q)]\|_F + \|\frac{[\dot{M}, \dot{V}_m, \dot{G}]}{\sigma_0 + \sigma_1\|\dot{q}_n\|}\|_F)^2 \leq (\rho + \tilde{\beta})^2 \quad (6.44)$$

where by Properties 6.1, 6.3 and 6.5 (proved in Appendix A),

$$\rho = \sup_q \|[M(q), V_m(q), G(q)]\|_F \quad \text{and} \quad \tilde{\beta} = \sup_{\dot{q},q}\{\|\frac{[\dot{M}, \dot{V}_m, \dot{G}]}{\sigma_0 + \sigma_1\|\dot{q}_n\|}\|_F\}.$$

The fact that $\|\dot{q}_n\| \leq \|\dot{q}_d - n_v\|_{max} + \|\dot{e}_n\|$ enables one to write

$$\frac{\alpha}{4\sigma_n}\|\Gamma\|_F^2 \leq \frac{1}{4}\alpha\sigma_n(\rho + \tilde{\beta})^2 \leq \tilde{\gamma}_0 + \tilde{\gamma}_1\|\dot{e}_n\| \tag{6.45}$$

where

$$\tilde{\gamma}_0 = \frac{\alpha}{4}(\rho + \tilde{\beta})^2(\sigma_0 + \sigma_1\|\dot{q}_d - n_v\|_{max}) \text{ and } \tilde{\gamma}_1 = \frac{\alpha}{4}(\rho + \tilde{\beta})^2\sigma_1. \tag{6.46}$$

Substituting (6.45) into (6.43) results in

$$\dot{L}_n \leq -\lambda_k\|\dot{e}_n\|^2 - \lambda_k e_n^T\Lambda^2 e_n - s_n^T\tilde{\tau}_d - \alpha\sigma_n\|\Phi + \frac{\Gamma}{2\sigma_n}\|_F^2 + \tilde{\gamma}_0 + \tilde{\gamma}_1\|\dot{e}_n\|$$

where λ_k is the smallest eigenvalue of K.

According to (6.32), one can write

$$\begin{aligned}-s_n^T\tilde{\tau}_d &\leq (\|\dot{e}_n\| + \|\Lambda\|\|e_n\|)(\tilde{d}_0 + \tilde{d}_1\|\dot{e}_n\| + \tilde{d}_2\|e_n\|) \\ &= \tilde{d}_0\|\dot{e}_n\| + \tilde{d}_1\|\dot{e}_n\|^2 + \tilde{c}_0\|e_n\| + \tilde{c}_1\|e_n\|^2 + \tilde{c}_2\|e_n\|\|\dot{e}_n\|\end{aligned}$$

where $\tilde{c}_0 = \tilde{d}_0\|\Lambda\|$; $\tilde{c}_1 = \tilde{d}_2\|\Lambda\|$ and $\tilde{c}_2 = \tilde{d}_2 + \tilde{d}_1\|\Lambda\|$. It then follows that

$$\begin{aligned}\dot{L}_n &\leq -\tilde{a}_1\|\dot{e}_n\|^2 - \tilde{a}_2\|e_n\|^2 + \tilde{c}_2\|\dot{e}_n\|\|e_n\| + \tilde{a}_3\|\dot{e}_n\| + \tilde{c}_0\|e_n\| \\ &\quad -\alpha\sigma_n\|\Phi + \frac{\Gamma}{2\sigma_n}\|_F^2 + \tilde{\gamma}_0\end{aligned} \tag{6.47}$$

where $\tilde{a}_1 = \lambda_k - \tilde{d}_1$; $\tilde{a}_2 = \lambda_k\lambda_\Lambda^2 - \tilde{c}_1$; $\tilde{a}_3 = \tilde{d}_0 + \tilde{\gamma}_1$. A further manipulation of (6.47) leads to

$$\begin{aligned}\dot{L}_n &\leq -\frac{\tilde{c}_2}{2}(\|\dot{e}_n\| - \|e_n\|)^2 - \tilde{p}_1(\|\dot{e}_n\| - \frac{\tilde{a}_3}{2\tilde{p}_1})^2 - \tilde{p}_2(\|e_n\| - \frac{\tilde{c}_0}{2\tilde{p}_2})^2 \\ &\quad -\alpha\sigma_n\|\Phi + \frac{\Gamma}{2\sigma_n}\|^2 + \tilde{\gamma} \\ &\leq -\tilde{p}_1(\|\dot{e}_n\| - \frac{\tilde{a}_3}{2\tilde{p}_1})^2 - \tilde{p}_2(\|e_n\| - \frac{\tilde{c}_0}{2\tilde{p}_2})^2 \\ &\quad -\alpha\sigma_n(\|\Phi\|_F - \frac{\rho + \tilde{\beta}}{2})^2 + \tilde{\gamma}\end{aligned} \tag{6.48}$$

where (6.44) has been substituted to eliminate the nonlinear function Γ; according to (6.16) and (6.17), λ_k is sufficiently large such that

$$\tilde{p}_1 = \tilde{a}_1 - \frac{\tilde{c}_2}{2} = \lambda_k - \tilde{d}_1 - 0.5(\tilde{d}_2 + \tilde{d}_1\|\Lambda\|) > 0$$

$$\tilde{p}_2 = \tilde{a}_2 - \frac{\tilde{c}_2}{2} = \lambda_k\lambda_\Lambda^2 - \tilde{d}_2\|\Lambda\| - 0.5(\tilde{d}_2 + \tilde{d}_1\|\Lambda\|) > 0$$
$$\tilde{\gamma} = \tilde{\gamma}_0 + \frac{1}{4}(\frac{(\tilde{d}_0 + \tilde{\gamma}_1)^2}{\tilde{p}_1} + \frac{\tilde{d}_0^2\|\Lambda\|^2}{\tilde{p}_2}).$$

Inequality (6.48) defines an elliptic ball in a 3-D Euclidian space $\|\dot{e}_n\|$-$\|e_n\|$-$\|\Phi\|$ whose inside contains the set $\{\dot{e}_n, e_n, \Phi \mid \dot{L}_n \geq 0\}$. In order words, $\dot{L}_n < 0$ whenever the norms of tracking or compensation errors exceed this elliptic ball. Thus there must exists a constant L_n^* such that $\dot{L}_n < 0$ whenever $L_n > L_n^*$, hinting that L_n is uniformly ultimately bounded. **Q.E.D.**

Using the same argument in Theorem 6.2, a similar bound can be obtained for the tracking errors $\dot{e}_n$ e_n and compensation error Φ. The detailed discussions are not given here in order to save some space.

It is interesting to note that λ_k alone can be simply adjusted to ensure system stability. This is favorable because Ks_n can be implemented with simple hardware amplifiers. Increasing λ_k means merely increasing the feedback gains.

6.4 Discussion

In choosing $\sigma(t)$, we would like to minimize $\sigma(t)\|\frac{\Gamma}{\sigma(t)}\|^2$. Since Γ is an unknown function, an alternative approach would be minimizing $\sigma(t)(\rho + \frac{\nabla_q\|\Gamma\|_F}{\sigma(t)})^2$. Here $\nabla_x f$ denotes the gradient of f with respect to x, we assume $\rho \approx \nabla_q\|\Gamma\|_F$. Then a reasonable choice of $\sigma(t)$ is $\sigma_1 = 1$ and σ_0 a small positive number.

In (6.28), α serves as the weight on the contribution of Φ to $\dot{L}$. A choice of large α implies faster convergence of $A \to [M(q), V_m(q), G(q)]$ while slower convergence of $\|e\| \to 0$ and $\|\dot{e}\| \to 0$. A small choice of α implies an opposite rate of convergence speeds. Thus one might prefer a small α.

The choice of α also depends on the sampling rate of the system. Generally, a smaller α requires a higher sampling rate. Because the integral operations in

(6.13) and (6.30) are implemented numerically, the numerical errors are inversely proportional to α as (6.13) and (6.30) indicate. When the initial values of A_M and A_V are too far away from $M(q)$ and $V_m(q)$, the initial tracking errors are naturally large. If a high gain $\frac{1}{\alpha}$ is used to evaluate (6.13) and (6.30), the numerical errors will be significant, and that could cause the system to become unstable.

In practice, the system sampling rate is limited, usually less than 500 samples per second. In our simulation, we find that this significantly limits our choice of α. An effective way to extend the range of α is to specify

$$\alpha = \alpha_0 + \alpha_1 \exp\{-\frac{t}{T}\}.$$

If α_1 is assigned by a large positive number, then the system will start with some small adaptive gain which gradually increases as $A \rightarrow [M(q), V_m(q), G(q)]$. By a proper choice of the time constant T, we achieve a small adaptive gain $\frac{1}{\alpha_1}$ during the transient period of adaptation and a high gain $\frac{1}{\alpha_0}$ in steady state tracking. In this case, we do not have to increase the system sampling rate. But we have to tolerate a longer transient adaptation time. With this modification, $\dot{L}$ will have one more negative definite term $\dot{\alpha}Tr[\Phi\Phi^T]$.

It can be shown that α can be replaced by a positive definite matrix without affecting the above three theorems. In our actual implementations, we replace α with a positive diagonal matrix. Thus different adaptive gains are assigned to different columns of adaptive matrix A. It is observed that the last column A_G needs a larger adaptive gain, because it represents an integral control term that helps to reduce steady state errors.

6.5 Simulation Results

Computer simulation experiments are conducted to test the adaptive controllers. The control object is the two-link planar robot given in Appendix A. Several sets of simulation results are in order.

6.5.1 Step response of the system

The objective of this experiment is to observe the step response of the system under different values of α_m and α_v. The robot is supposed to move from $q_1 = q_2 = 0$ to the position $q_1 = q_2 = 45^o$. The control and adaptive laws are given by (6.12) and (6.13). We choose $K = 30I$, $\Lambda = 20I$, $\alpha_g = .0025$, $\sigma(t) = ||\dot{q}||$. All initial values of the close-loop system are set to zero.

Figure (6.2) plots the family of step responses of the two joints under the condition $g = \alpha_m^{-1} = \alpha_v^{-1} = \{10, 2, 1, 0.5, 0.1\}$. (The upper part represents joint two and the lower part represents joint one. The rest of figures follow the same convention.) One may note that different α values are used to update A_M, A_V and A_G respectively.

6.5.2 Effect of the K matrix

In order to observe the effect of feedback gain matrix K, the above simulation is repeated with the same initial values. This time, four different K matrices are tested ($K = 10I$, $K = 30I$, $K = 60I$ and $K = 100I$). The step response is observed to be improved by increasing K. When K is too small ($K = 10I$), the closed-llop tend to be unstable as Fig. 6.3 demonstrates.

6.5.3 Tracking at different speeds

Adaptive controller (6.12), (6.13) is used to track the desired trajectory for the two joint positions

$$q_{1d} = q_{2d} = 1 + \sin(2\pi f t)$$

where f is chosen to be $f_1 = 0.5$ and $f_2 = 2$ respectively, representing "slow" and "fast" movement of the robot. We choose $K = 30I$, $\Lambda = 20I$, $\alpha_g^{-1} = 400$, $\sigma(t) = ||\dot{q}||$. The system sampling rate is 500Hz; All initial values of the system are set to zero. The two varying gains are determined by

$$\alpha_m = \alpha_v = 0.1 + 100e^{-\frac{t}{T}} \quad \text{where } T = 0.1.$$

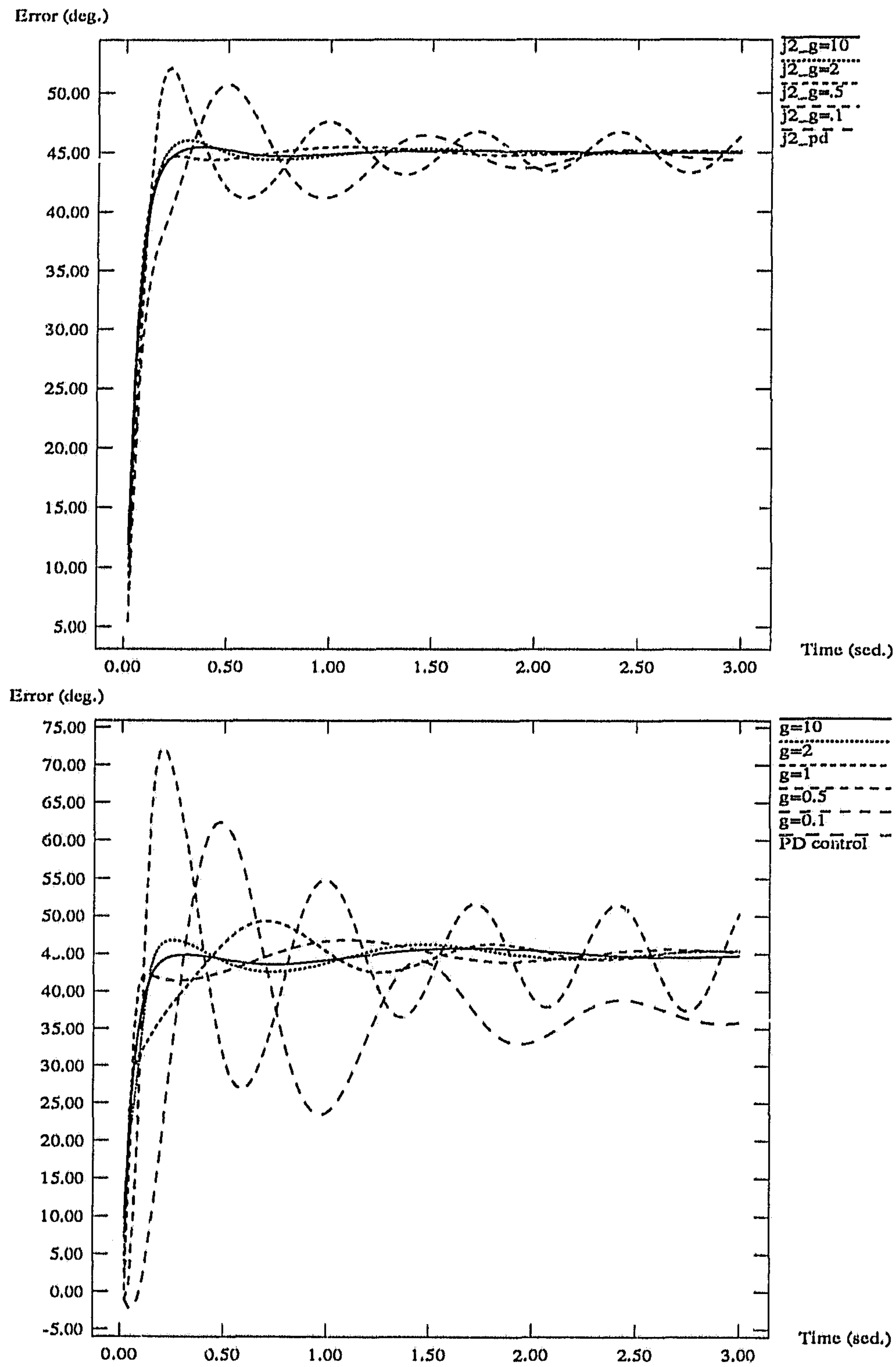

Figure 6.2: Step responses of the two joints.

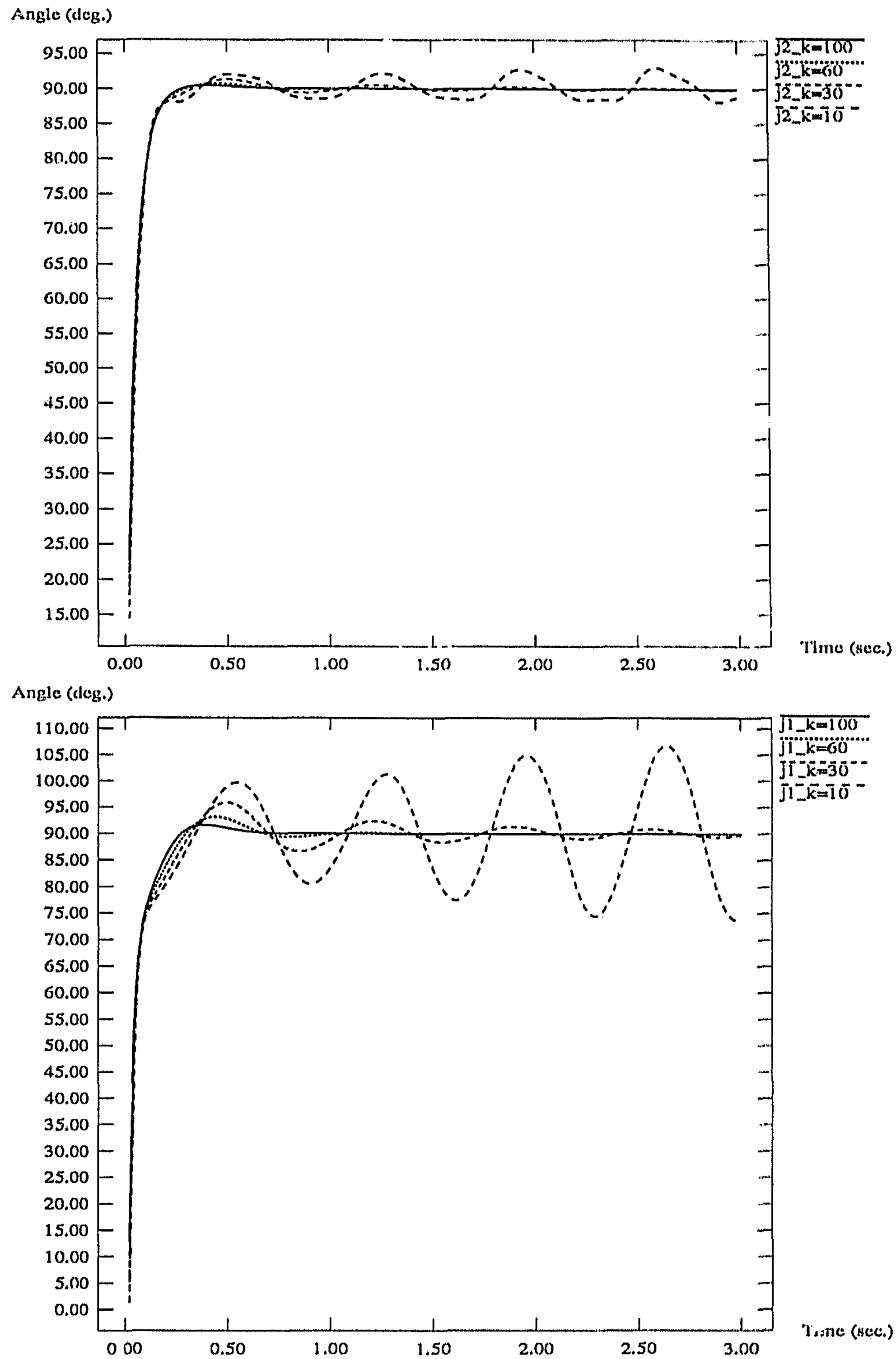

Figure 6.3: Effect of the feedback gain matrix K.

Simulation results are plotted in figures (6.4) and (6.5), with (6.4) plotting the slow tracking performance while (6.5) the fast tracking cases. The dimension of the joint position is the radian in this experiment.

6.5.4 Tracking with a varying load

The model and experimental conditions are exactly the same as in the previous experiment. In order to observe the performance of the controller under varying load, we let m_2 of the robot model change from 5kg to 10kg at the beginning of the third second. As we can see from figure (6.6), the robot tracks the desired trajectory well. In this experiment, the speed of tracking is set slow in order to observe the slight difference caused by the load change. Such a difference is almost invisible when the closed-loop system tracks a desired trajectory at fast speeds.

6.5.5 Tracking in measurement noise

The desired path is $q_1 = q_2 = 1 - \cos(f\pi t)$ where f is chosen to be $f_1 = 2$ for fast tracking and $f_2 = 0.5$ for slow tracking. In this experiment, we study the effect of measurement noise. The noise sources are assumed to be white with uniform distribution [-.1, .1] (rad.). We assume the noise effects of different sensors are independent of each other.

Equations (6.29) and (6.30) are implemented. Every parameter is the same as in 6.3 except $\sigma(t) = 0.001 + \|\dot{q}_n\|$. The system sampling rate is 500Hz; all initial values of the system are set to zero.

The performance under white noise is given in (6.7) for slow tracking and (6.8) for fast tracking. The white noise of one measurement channel is superimposed in the figures. Thus one can compare the magnitude of the noise with the desired trajectory. (The signal to noise ratio is about 20 dB.) Because of the measuring noise, the robot movement deviates from the desired path. But the system is still stable. Since the white noise contains all possible frequency components, it must be able to excite the potentially unstable movement of the close-loop system. Yet

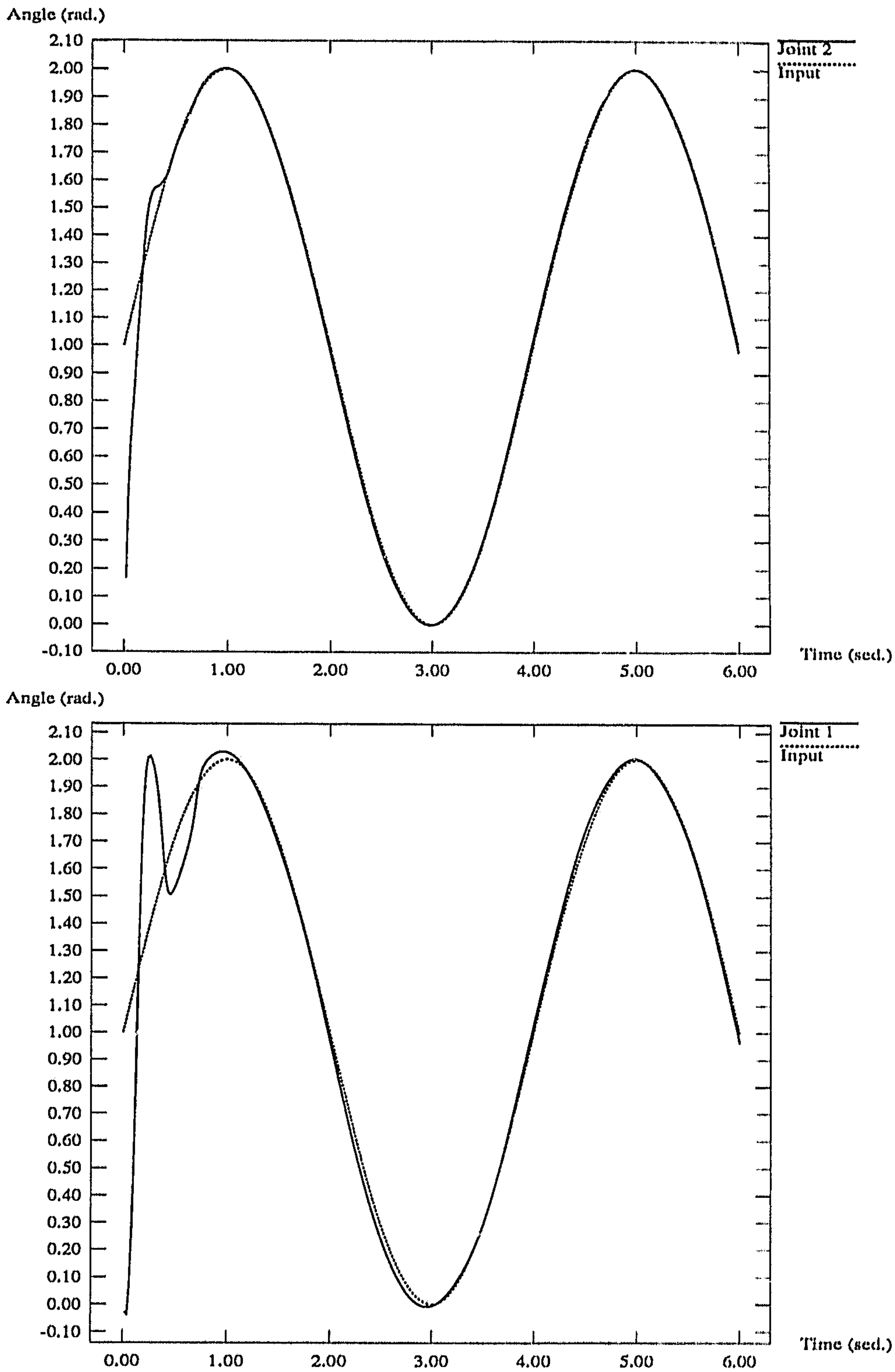

Figure 6.4: Slow tracking performance.

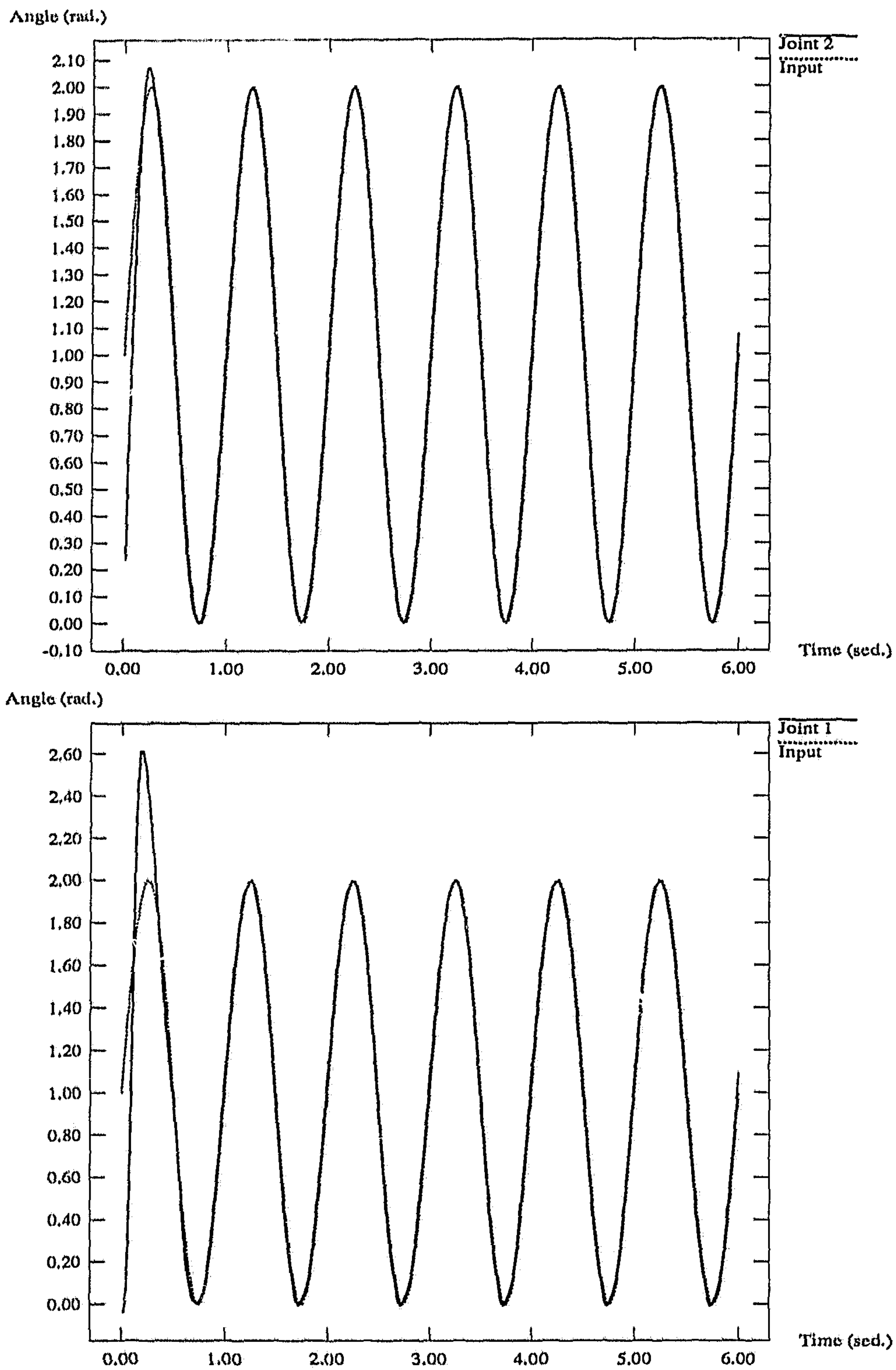

Figure 6.5: Fast tracking performance.

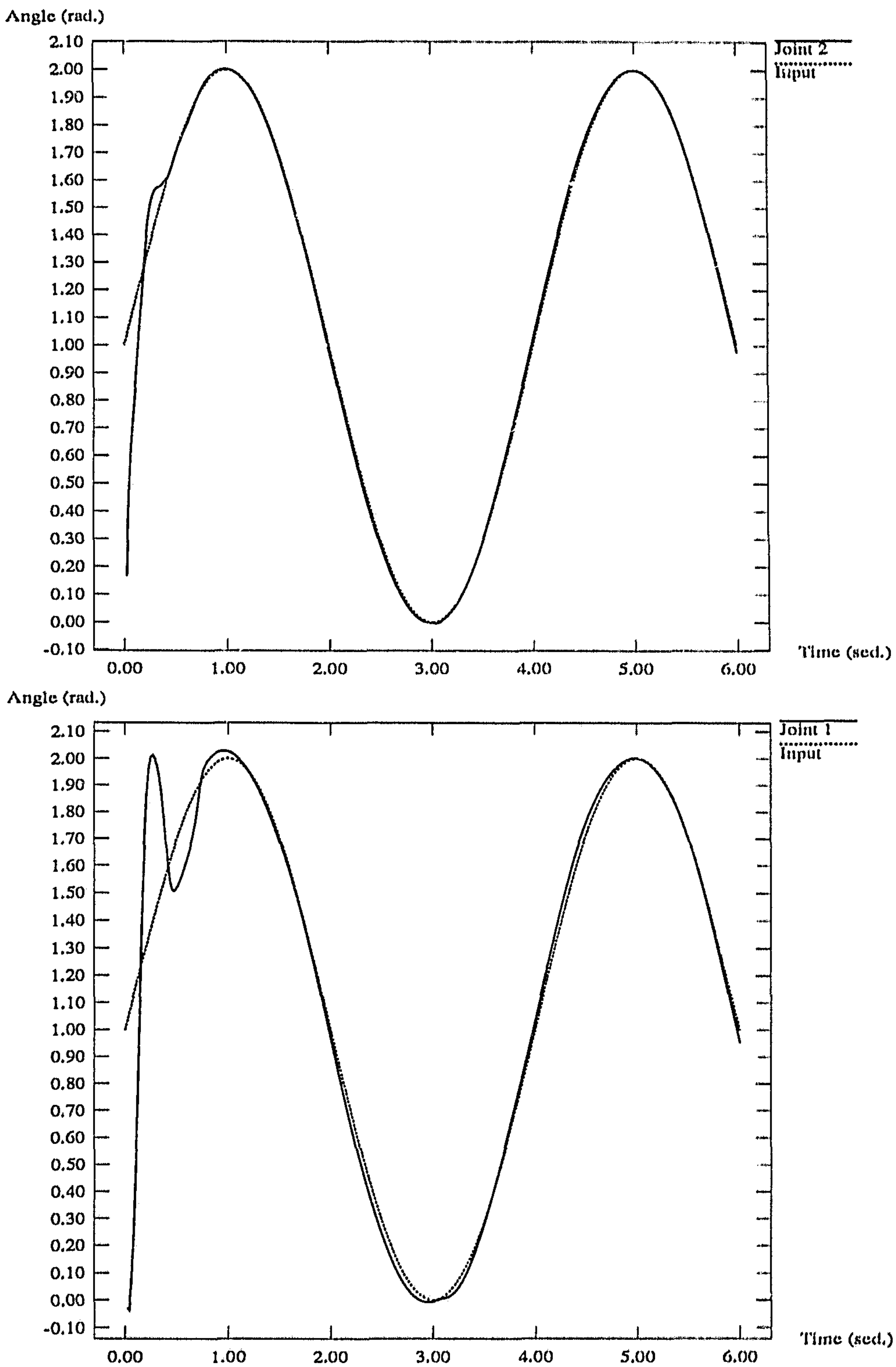

Figure 6.6: Tracking with a varying load.

we find the system is stable at least in the above experimental conditions. This agrees with our analytical result (theorem 6.3).

6.6 Summary

In this study, we present a simple adaptive controller for trajectory control of robotic manipulators. The control scheme requires minimum knowledge about the system dynamics and its implementation is very simple. Only some integral operations are involved. Since the adaptation of each element is identical to that of the others, the algorithms are suitable for parallel implementation.

As pointed out by Slotine [39], there are two main groups of adaptive manipulator controllers: the "linear parameterization" and "approximation" groups. The present study belongs to the second group. Adaptive controllers in this group require little knowledge of system dynamics. Their structures are very simple and easy to implement. A common short-coming of these algorithms is that they "had to rely on various restrictive assumptions or approximations for adaptive control design and analysis, such as linearizing the robot dynamics, approximating the joint motions as decoupled, or assuming 'slow' variations of the inertia matrix" [39]. Our algorithm does not have such problems due to the relaxed assumptions in theorems 6.1–6.3.

In theorem 6.3, we prove that the designed controllers are robust in the presence of both unmodeled dynamics and measuring noise. A similar study on Slotine and Li's algorithm was conducted by Reed and Ioannou [9]. It is observed that the original algorithm by [39] is potentially unstable in the presence of some bounded disturbances. A modified version results in stable performance but the tracking errors are no longer guaranteed to converge to zero. According to theorem 6.3 and the experimental results, the designed algorithm is able to work in the same conditions with reasonably good performance. But the implementation is much simpler.

In order to test the designed algorithms, we conducted a series of simulation experiments. These controllers are observed to work well under different speeds.

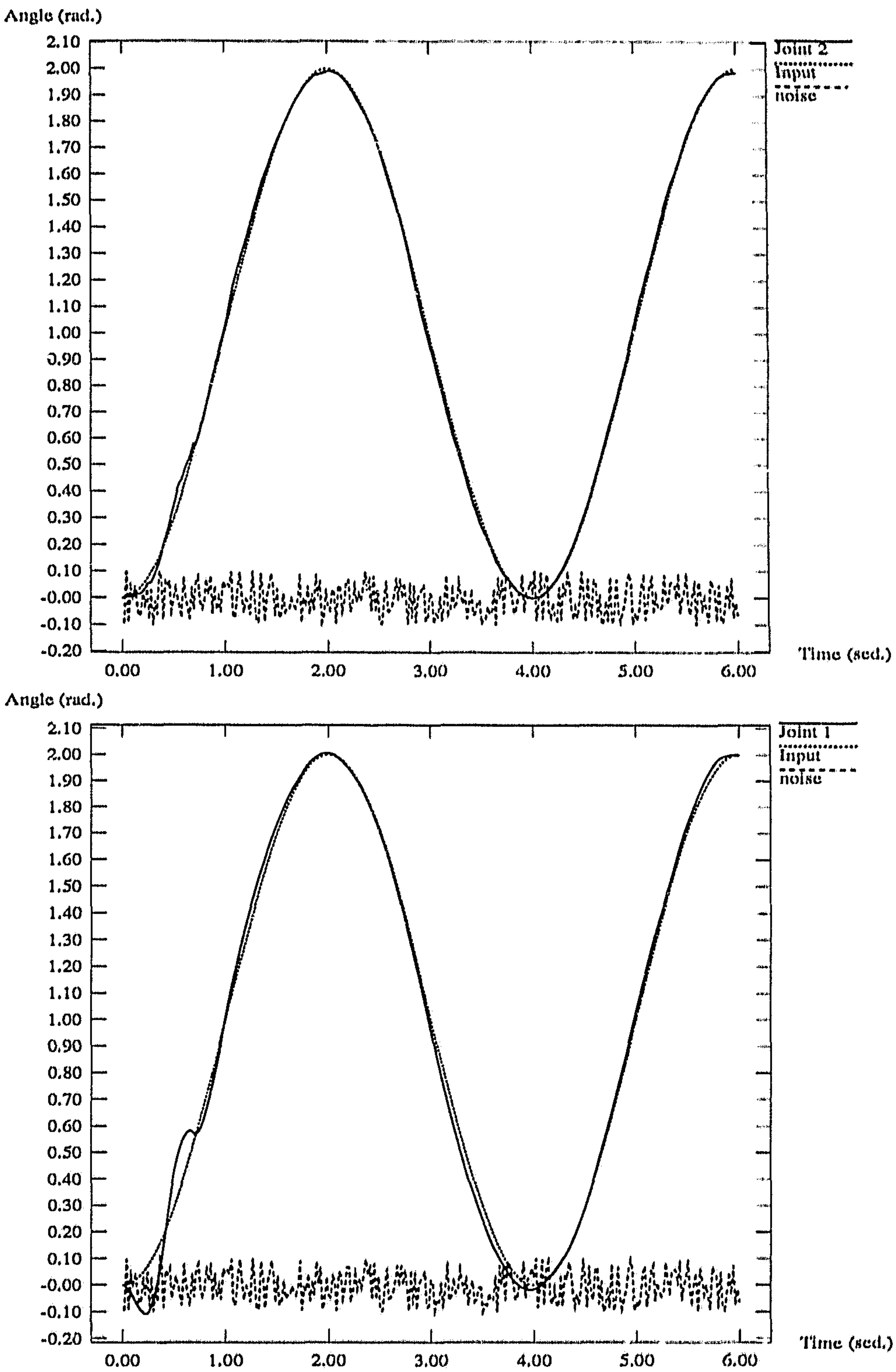

Figure 6.7: Slow tracking with measurement noise.

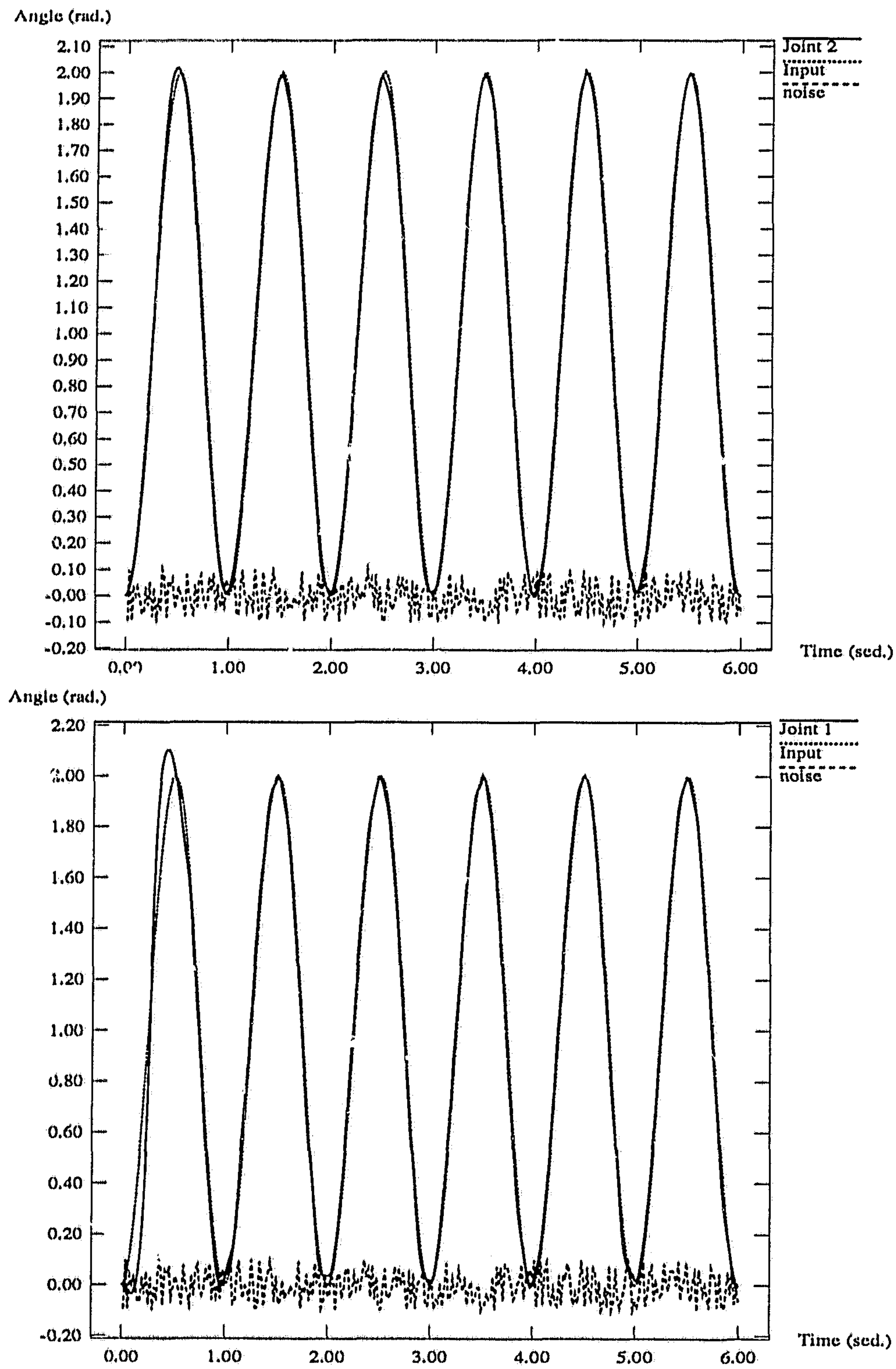

Figure 6.8: Fast tracking with measurement noise.

The close-loop system is robust under either external force or internal measurement noise. It is observed, from both analytical and experimental studies, that a large K always leads to a better performance. In our experiments, we generally use $K = 30I$. The results are reasonably good. Yet they can be further improved by simply increasing K.

Appendix A

<u>A Particular Example</u> The simulation experiment is conducted on a two-link planar robot. Its equation of motion is described by

$$\begin{aligned}
\tau_1 &= [(2l_1\cos(q_2)+l_2)l_2m_2 + l_1^2(m_1+m_2)]\ddot{q}_1 + [l_2^2m_2 + l_1l_2\cos(q_2)m_2]\ddot{q}_2 \\
&\quad -2l_1l_2m_2\sin(q_2)\dot{q}_2\dot{q}_1 - l_1l_2m_2\sin(q_2)\dot{q}_2^2 \\
&\quad +g(m_2l_2\cos(q_1+q_2)+(m_1+m_2)l_1\cos(q_1)) \\
\tau_2 &= [l_2^2m_2 + l_1l_2\cos(q_2)m_2]\ddot{q}_1 + l_2^2m_2\ddot{q}_2 + l_1l_2m_2\sin(q_2)\dot{q}_1^2 + m_2l_2g\cos(q_1+q_2)
\end{aligned}$$

where $l_1 = .7$, $l_2 = .5$ (meter), $m_1 = 10$ and $m_2 = 5$ (kg). The matrix form of the system is given by

$$M(q)\begin{bmatrix}\ddot{q}_1\\ \ddot{q}_2\end{bmatrix} + V(\dot{q},q)\begin{bmatrix}\dot{q}_1\\ \dot{q}_2\end{bmatrix} + G(q) = \begin{bmatrix}\tau_1\\ \tau_2\end{bmatrix}$$

where

$$M(q) = \begin{bmatrix}(2l_1\cos(q_2)+l_2)l_2m_2 + l_1^2(m_1+m_2) & l_2^2m_2 + l_1l_2\cos(q_2)m_2 \\ l_2^2m_2 + l_1l_2\cos(q_2)m_2 & l_2^2m_2\end{bmatrix},$$

and

$$G(q) = \begin{bmatrix}g(m_2l_2\cos(q_1+q_2)+(m_1+m_2)l_1\cos(q_1)) \\ m_2l_2g\cos(q_1+q_2)\end{bmatrix}$$

are uniquely determined; but $V(\dot{q},q)$ is not unique. Two possible forms are presented here

$$V_*(\dot{q},q) = \begin{bmatrix}-2l_1l_2m_2\sin(q_2)\dot{q}_2 & -l_1l_2m_2\sin(q_2)\dot{q}_2 \\ l_1l_2m_2\sin(q_2)\dot{q}_1 & 0\end{bmatrix}$$

and

$$V(q,q) = \begin{bmatrix} -l_1 l_2 m_2 \sin(q_2)\dot{q}_2 & -l_1 l_2 m_2 \sin(q_2)(\dot{q}_1 + \dot{q}_2) \\ l_1 l_2 m_2 \sin(q_2)\dot{q}_1 & 0 \end{bmatrix}.$$

One can easily verify that $V_*(\dot{q}, q)\dot{q} = V(\dot{q}, q)\dot{q}$. But only $V(\dot{q}, q)$ satisfies (6.8). The other properties can be verified in this system.

Chapter 7

A Robust Self-Tuning Computed-Torque Controller

In the previous two chapters, adaptive controllers based on some knowledge about the manipulator dynamics are developed. In order to ensure small tracking errors, these controllers require high feedback and adaptive gains. In this chapter, a self-tuning controller is developed which does not require high-gain feedbacks while achieving good performance. This is done by exploring more detailed knowledge about the system dynamics. A general algorithm for computing the $n \times l$ matrix regressor $W(\dot{q}, q)$ without using the acceleration feedback is derived. It can be applied to any open-chain mechanisms. The unknown inertia parameters can be estimated by means of the resulting regressor. A distinct feature of this chapter is the proof of robustness for a class of computed-torque controllers. The computed-torque controllers are proven to be able to tolerate a certain degree of parameter uncertainty and unmodeled disturbances while maintaining stable tracking. This relaxes the requirement of the adaptation law design. The closed-loop system is proved to be stable without the persistent excitation.

7.1 Robustness of the Controller

The dynamics of a n-link robotic manipulator are described by a second order nonlinear differential equation

$$M(q)\ddot{q} + V(\dot{q}, q)\dot{q} + G(q) = \tau + \tau_d \tag{7.1}$$

where τ_d represents the external disturbance; $M(q)$, $V(\dot{q}, q) \in R^{n\times n}$ and $G(q) \in R^n$ are nonlinear functions of $\dot{q}$, q as well as a set of inertia parameters, such as mass, center of mass and inertia of all links. In practice these parameters are available for the first $(n-1)$ links. The inertia parameters of the last link are, in general, inaccurate because the manipulator may frequently pick up or release unknown objects. The dynamic model coefficients available to the controller are $\hat{M}(q)$, $\hat{V}(\dot{q}, q)$ and $\hat{G}(q)$ computed using the inaccurate inertia parameters. The objective of this section is to study the mis-matched effects on the stability of computed-torque controllers.

The control law is given by

$$\tau = \hat{M}(q)\psi + \hat{V}(\dot{q}, q)\dot{q} + \hat{G}(q) \tag{7.2}$$

where $\psi = \ddot{q}_d - K_v\dot{e} - K_p e$; $e = q - q_d$ is the tracking error; q_d is the desired trajectory; $K_v > 0$ and $K_p > 0$ are two positive definite matrices. The notation $\hat{}$ indicates that the corresponding matrix or vector is evaluated by substituting a set of "model" parameters ζ_m instead of the real parameter ζ_r. (The definition of ζ and its relation with $M(q)$, $V(\dot{q}, q)$ and $G(q)$ will be given in sections 3 and 4.)

The closed-loop system is determined by substituting (7.2) into (7.1), that is

$$\ddot{e} + K_v\dot{e} + K_p e = M^{-1}(q)[\Delta M\psi + \Delta V\dot{q} + \Delta G - \tau_d] \tag{7.3}$$

where $\Delta M = \hat{M}(q) - M(q)$, $\Delta V = \hat{V}(\dot{q}, q) - V(\dot{q}, q)$ and $\Delta G = \hat{G}(q) - G(q)$ are the mis-matched effects caused by the inaccurate model parameter ζ_m.

According to [37], [63] and [64], $M(q)$ and $G(q)$ are uniformly bounded. This means

$$||M^{-1}(q)\Delta M\psi|| \leq c_M||\psi|| \leq c_{m0} + c_{m1}||\dot{e}|| + c_{m2}||e||, \tag{7.4}$$

$$\|M^{-1}(q)\Delta G\| \leq c_G. \tag{7.5}$$

The i-th component of vector $V(\dot{q},q)\dot{q}$ can be expressed as a quadratic form $\dot{q}^T V_i(q)\dot{q}$ [37] where $V_i(q) \in R^{n\times n}$, $1 \leq i \leq n$ are uniformly bounded matrices. The i-th component of vector $\Delta V\dot{q}$ should then be expressed as $\dot{q}^T(\hat{V}_i - V_i)\dot{q}$ which suggests

$$\|M^{-1}(q)\Delta V\dot{q}\| \leq c_V\|\dot{q}\|^2 \leq c_{v0} + c_{v1}\|\dot{e}\| + c_{v2}\|\dot{e}\|^2. \tag{7.6}$$

In (7.4)–(7.6), c_{m0}, c_{m1}, c_{m2}, c_G, c_{v0}, c_{v1} and c_{v2} are positive constant bounds depending on the parameter error $\Delta\zeta = \zeta_m - \zeta_r$. In Sections 3 and 4, it will be shown that these constant bounds become zero when $\zeta_r = \zeta_m$.

Equation (7.3) can be expressed in state space as

$$\dot{\epsilon} = A_m\epsilon + \tau_\sigma \tag{7.7}$$

where $\epsilon^T = [\dot{e}^T, e^T]$;

$$A_m = \begin{bmatrix} -K_v & -K_p \\ I & 0 \end{bmatrix} \quad \text{and} \quad \tau_\sigma = \begin{bmatrix} M^{-1}(q)(\Delta M\psi + \Delta V\dot{q} + \Delta G + \tau_d) \\ 0 \end{bmatrix}.$$

It is not difficult to find positive definite matrix pair

$$P = \begin{bmatrix} I & aI \\ aI & \frac{1}{2}(aK_v + K_p^T) \end{bmatrix} \quad \text{and} \quad Q = \begin{bmatrix} K_v - aI & 0 \\ 0 & aK_p \end{bmatrix} \quad 0 < a < 1, \quad aI < K_v$$

such that

$$\frac{1}{2}(PA_m + A_m^T P) = -Q. \tag{7.8}$$

The largest and smallest eigenvalues of P and Q are denoted by λ_{Pmax}, λ_{Qmax} and λ_{Pmin}, λ_{Qmin} respectively. Assuming that $\|\tau_d\|$ is uniformly bounded and using (7.4)–(7.6), one can write

$$\begin{aligned} \|P\tau_\sigma\| &= \left\| \begin{bmatrix} I \\ aI \end{bmatrix} M^{-1}(q)(\Delta M\psi + \Delta V\dot{q} + \Delta G + \tau_d) \right\| \\ &\leq c_0 + c_1\|\epsilon\| + c_2\|\epsilon\|^2 \end{aligned} \tag{7.9}$$

for some constants $0 < c_0 \leq c_{mo} + c_G + c_{vo}$, $0 < c_1 \leq \max\{c_{m1}, c_{m2}\} + c_{v1}$ and $0 < c_2 = c_{v2}$.

Theorem 7.1 *The closed-loop system (7.7) is uniformly bounded if*

$$\alpha > 0, \quad \alpha^2 > 4\beta\gamma \quad \textit{and} \quad \|\epsilon(0)\| \leq \frac{1}{\beta\sqrt{2\lambda_{Pmin}}}(\alpha + \sqrt{\alpha^2 - 4\beta\gamma}) \tag{7.10}$$

where $\epsilon(0)$ *denotes the initial tracking error;*

$$\alpha = \inf_{q,\dot{q}}\{2\frac{\epsilon^T Q\epsilon - c_1\|\epsilon\|^2}{\epsilon^T P\epsilon}\} = 2\frac{\lambda_{Qmin} - c_1}{\lambda_{Pmin}} > 0; \tag{7.11}$$

$$\beta = \sup_{q,\dot{q}}\{\frac{c_2\|\epsilon\|^3\sqrt{8}}{(\epsilon^T P\epsilon)^{\frac{3}{2}}}\} = \frac{c_2\sqrt{8}}{\lambda_{Pmin}^{\frac{3}{2}}} \quad \textit{and} \quad \gamma = \sup_{q,\dot{q}}\{\frac{\sqrt{2}\|\epsilon^T\|c_0}{\sqrt{\epsilon^T P\epsilon}}\} = \frac{\sqrt{2}c_0}{\sqrt{\lambda_{Pmin}}}. \tag{7.12}$$

Proof: Consider a positive definite function $L = \frac{1}{2}\epsilon^T P\epsilon$. Its time derivative evaluated along (7.7) is given by

$$\dot{L} = -\epsilon^T Q\epsilon + \epsilon^T P\tau_\sigma \leq c_0\|\epsilon\| - (\lambda_{Qmin} - c_1)\|\epsilon\|^2 + c_2\|\epsilon\|^3 \tag{7.13}$$

where (7.8) and (7.9) have been substituted. A substitution of (7.11) and (7.12) into (7.13) results in

$$\dot{L} \leq -\alpha L + \beta L^{\frac{3}{2}} + \gamma L^{\frac{1}{2}}.$$

According to the principle of comparison, L is uniformly bounded by a scalar function L_* which satisfies

$$\dot{L}_* = -\alpha L_* + \beta L_*^{\frac{3}{2}} + \gamma L_*^{\frac{1}{2}} \tag{7.14}$$

with an initial condition $L(0) = L_*(0)$.

When the conditions given by (7.10) are satisfied, a variable substitution $l_*^2 = L_*$ will change (7.14) to

$$\begin{aligned} \dot{l}_* &= \frac{1}{2}(\gamma - \alpha l_* + \beta l_*^2) \\ &= \frac{\beta}{2}(l_* - l_1)(l_* - l_2) \end{aligned} \tag{7.15}$$

where

$$l_1 = \frac{1}{2\beta}(\alpha - \sqrt{\alpha^2 - 4\gamma\beta}) \quad \text{and} \quad l_2 = \frac{1}{2\beta}(\alpha + \sqrt{\alpha^2 - 4\gamma\beta}).$$

If $l_*(0)$, the initial value of l_*, is less than or equal to l_2, the closed-form solution of (7.15) can be obtained as

$$\frac{l_* - l_2}{l_* - l_1} = \frac{l_*(0) - l_2}{l_*(0) - l_1} e^{0.5t\sqrt{\alpha^2 - 4\gamma\beta}}$$

which can be written as

$$l_*(t) = \frac{l_2[l_*(0) - l_1] + l_1[l_2 - l_*(0)]e^{0.5t\sqrt{\alpha^2-4\gamma\beta}}}{[l_*(0) - l_1] + [l_2 - l_*(0)]e^{0.5t\sqrt{\alpha^2-4\gamma\beta}}}.$$

The analytical solution of $l_*(t)$ suggests that the tracking error ϵ is uniformly bounded as long as $l_*(0) \leq l_2$ which is equivalent to the last inequality of (7.10). **Q.E.D.**

According to (7.10), (7.11) and (7.12), An effective way to ensure a stable system and reduce tracking error is to identify the correct parameter vector ζ_m such that c_1 and c_2 become as small as possible. This will be discussed in the following sections.

7.2 Parameter Uncertainties

A robotic manipulator is an open-chain articulated mechanical system. Its links can be labeled from the base to the end-tip in a consecutive order. Each link is connected to its neighbors or the base by revolute or prismatic joints, and has a "link coordinate frame" attached to it. Customarily [63, 64] the origin of the i-th link frame is located at the i-th joint, connecting the i-th and the $(i-1)$-th link; the z_i axis is always directed along the axis of the i-th joint. In other words, the i-th link either rotates or translates along the z_i axis. The coordinate frame attached to the robot base is assumed inertial. Unless stated otherwise, all vectors in this study are expressed in the base frame. The entire system consists of n links and possesses n degree-of-freedom.

Let $q^T = [q_1, \ldots, q_n]$ be the vector of the generalized coordinate. The component q_i defines the relative displacement (rotation or translation) of the two neighboring links with respect to the z_i axis. Each link is associated with the Jacobian matrix which is defined as

$$\begin{bmatrix} v_{c_i} \\ \omega_i \end{bmatrix} = J_{c_i}(q)\dot{q} \qquad J_{c_i}(q) \in R^{6\times n}$$

where v_{c_i} is the linear velocity of the mass center and ω_i the angular velocity of the i-th link. For a robot with all revolute joints, the Jacobian matrix with respect to the mass center of the k-th link is given by

$$J_{c_k}(q) = \begin{bmatrix} z_1 \times (P_{1k} + c_k) & z_2 \times (P_{2k} + c_k) & \ldots & z_k \times c_k & 0 & \ldots & 0 \\ z_1 & z_2 & \ldots & z_k & 0 & \ldots & 0 \end{bmatrix} \tag{7.16}$$

where z_i is an unit vector representing the z_i axis; P_{ik} is a vector between the origins of the i-th and the k-th link frames; c_k is a vector representing the k-th link mass center. The $(k+1)$-th and higher ordered columns are zero because of the open-chain mechanical structure; the generalized velocities of higher joints do not affect the linear and angular velocities of lower links. In case the i-th joint of a robot is translational, the i-th column of the above Jacobian matrix will be changed from

$$\begin{bmatrix} z_i \times (P_{ik} + c_k) \\ z_i \end{bmatrix} \quad \text{to} \quad \begin{bmatrix} z_i \\ 0 \end{bmatrix}.$$

Thus, the following discussion will focus on a robot with all revolute joints without loss of generality.

The inertia forces and moments of inertia forces of the i-th link reduced to the i-th link mass center are

$$f_i = -\frac{d}{dt}(m_i v_{c_i}) \quad \text{and} \quad n_i = -\frac{d}{dt}(T_i\, \omega_i) \tag{7.17}$$

where m_i is the mass and T_i the inertia tensor of the i-th link. Then applying the D'Alamberl principle, the dynamics of a robotic manipulator are expressed as

$$\begin{aligned} \tau + \tau_d &= \sum_{i=1}^{n} J_{c_i}^T(\frac{d}{dt}\begin{bmatrix} m_i v_{c_i} \\ T_i\omega_i \end{bmatrix} - m_i\begin{bmatrix} g \\ 0 \end{bmatrix}) \\ &= \sum_{i=1}^{n} J_{c_i}^T(q)(Q_i\begin{bmatrix} \dot{v}_{c_i} \\ \dot{\omega}_i \end{bmatrix} + \begin{bmatrix} 0 \\ \omega_i \times T_i\omega_i \end{bmatrix} - \begin{bmatrix} m_i g \\ 0 \end{bmatrix}) \end{aligned} \tag{7.18}$$

where $g \in R^3$ is the gravitation vector;

$$Q_i = \begin{bmatrix} m_i I_{3\times3} & 0 \\ 0 & T_i \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} \dot{v}_{c_i} \\ \dot{\omega}_i \end{bmatrix} = J_{c_i}(q)\ddot{q} + \dot{J}_{c_i}(q)\dot{q} \tag{7.19}$$

where $I_{3\times3}$ is an identity matrix. Substituting (7.19) into (7.18) and then comparing the result with (7.1), one immediately finds that

$$M(q) = \sum_{i=1}^{n} J_{c_i}^T(q)Q_iJ_{c_i}(q), \tag{7.20}$$

$$V(\dot{q}, q)\dot{q} = \sum_{i=1}^{n} J_{c_i}^T(q)(Q_i\dot{J}_{c_i}^T(q)\dot{q} + \begin{bmatrix} 0 \\ \omega_i \times T_i\omega_i \end{bmatrix}), \tag{7.21}$$

$$G(q) = -\sum_{i=1}^{n} m_iJ_{c_i}^T(q)\begin{bmatrix} g \\ 0 \end{bmatrix}. \tag{7.22}$$

Assuming that the parameter uncertainty is due to $\hat{c}_n$, $\hat{m}_n$ and $\hat{T}_n$, the unknown load of the last link, then (7.20) immediately implies

$$\begin{aligned} \Delta M &= J_{\hat{c}_n}^T\hat{Q}_nJ_{\hat{c}_n} - J_{c_n}^TQ_nJ_{c_n} \\ &= \begin{bmatrix} \Delta M_{11} & \Delta M_{12} & \dots & \Delta M_{1n} \\ \Delta M_{21} & \Delta M_{22} & \dots & \Delta M_{2n} \\ \vdots & \vdots & \dots & \vdots \\ \Delta M_{n1} & \Delta M_{n2} & \dots & \Delta M_{nn} \end{bmatrix} \end{aligned} \tag{7.23}$$

where

$$\begin{aligned} \Delta M_{ij} &= \hat{m}_n[z_i \times (P_{in} + \hat{c}_n)]^T[z_j \times (P_{jn} + \hat{c}_n)] - \\ &\quad -m_n[z_i \times (P_{in} + c_n)]^T[z_j \times (P_{jn} + c_n)] + z_i^T(\hat{T}_n - T_n)z_j \\ &= \Delta m_n[(z_i^Tz_j)(P_{in}^TP_{jn}) - (z_i^TP_{jn})(z_j^TP_{in})] + \\ &\quad +[(z_i^Tz_j)(P_{in} + P_{jn}) - (z_j^TP_{in})z_i - (z_i^TP_{in})z_j]^T\Delta\tilde{c}_n + z_i^T\Delta\tilde{T}_nz_j \end{aligned} \tag{7.24}$$

where $\Delta m_n = \hat{m}_n - m_n$, $\Delta\tilde{c}_n = \hat{m}_n\hat{c}_n - m_n c_n$ and

$$\Delta\tilde{T}_n = \hat{T}_n - T_n + \hat{m}_n(\hat{c}_n^T\hat{c}_n I_{3\times3} - \hat{c}_n\hat{c}_n^T) - m_n(c_n^T c_n I_{3\times3} - c_n c_n^T).$$

In deriving (7.24), a useful identity $(\vec{a}\times\vec{b})^T(\vec{c}\times\vec{d}) = (\vec{a}^T\vec{c})(\vec{b}^T\vec{d}) - (\vec{a}^T\vec{d})(\vec{c}^T\vec{b})$ has been substituted. Similarly, (7.22) leads to

$$\Delta G^T = [\Delta G_1, \ldots, \Delta G_n]$$

where

$$\begin{aligned}\Delta G_i &= m_n g^T[z_i\times(P_{in}+c_n)] - \hat{m}_n g^T[z_i\times(P_{in}+\hat{c}_n)] \\ &= -\Delta m_n g^T(z_i\times P_{in}) - \Delta\tilde{c}_n^T(g\times z_i)\end{aligned} \tag{7.25}$$

According to (7.21), the expression of $\Delta V\dot{q}$ involves the time derivative of $J_{c_n}(q)$. To evaluate the analytical form of $\dot{J}_{c_i}(q)$, one can consider the i-th column of $J_{c_k}(q)$. It consists of two sub-vectors $z_i\times(P_{ik}+c_k)$ and z_i. The corresponding time derivatives of these sub-vectors are given by

$$\dot{z}_i\times(P_{ik}+c_k) + z_i\times(\omega_k\times c_k + v_k - v_i) \quad \text{and} \quad \dot{z}_i = \omega_i\times z_i = S(\omega_i)z_i$$

respectively, where v_i is the linear velocity of the i-th joint frame origin; the relative linear velocity between the mass center of the k-th link and the origin of the i-th link frame is given by

$$\omega_k\times c_k + v_k - v_i = \dot{P}_{ik} + \dot{c}_k;$$

while $S(x)$ is a skew-symmetric matrix such that $S(x)y = x\times y\ \forall x, y\in R^3$.

Now one can write

$$\begin{aligned}\dot{J}_{c_k}(q) &= \begin{bmatrix} \dot{z}_1\times(P_{1,k}+c_k) & \dot{z}_2\times(P_{2,k}+c_k) & \ldots & \dot{z}_k\times c_k & 0 & \ldots & 0 \\ \dot{z}_1 & \dot{z}_2 & \ldots & \dot{z}_k & 0 & \ldots & 0\end{bmatrix} \\ &\quad + \begin{bmatrix} z_1\times(\omega_k\times c_k + v_{k1}) & \ldots & z_k\times(\omega_k\times c_k) & 0 & \ldots & 0 \\ 0 & \ldots & 0 & 0 & \ldots & 0\end{bmatrix} \\ &= \bar{J}_{c_k}(q,\dot{q}) + \tilde{J}_{c_k}(q,\dot{q}).\end{aligned} \tag{7.26}$$

where $v_{ki} = v_k - v_i$. $\bar{J}_{c_k}(q,\dot{q})$, the first term in (7.26), is a matrix function of both q and $\dot{q}$ because of the time derivative $\dot{z}_i$. This matrix function can be evaluated using (7.16) by substituting $\dot{z}_i$ in the place of z_i.

As demonstrated in [37], the i-th component of vector $\Delta V\dot{q}$ can be expressed as a quadratic form $\dot{q}^T\Delta V_i\dot{q}$. Its explicit expression can be obtained by substituting (7.26) into (7.21) as

$$\begin{aligned}
\dot{q}^T\Delta V_i\dot{q} \;=\; & \Delta m_n \sum_{j=1}^{n}[(z_i^T\dot{z}_j)(P_{in}^TP_{jn}) - (z_i^TP_{jn})(\dot{z}_j^TP_{in})]\dot{q}_j + \\
& +\Delta\tilde{c}_n^T\sum_{j=1}^{n}[(z_i^T\dot{z}_j)(P_{in}+P_{jn}) - (\dot{z}_j^TP_{in})z_i - (z_i^TP_{in})\dot{z}_j]\dot{q}_j + \\
& +\sum_{j=1}^{n} z_i^T\Delta\tilde{T}_n\dot{z}_j\dot{q}_j + \\
& +\Delta m_n\sum_{j=1}^{n}[(z_i^Tz_j)(P_{in}^Tv_{nj}) - (z_i^Tv_{nj})(z_j^TP_{in})]\dot{q}_j + \\
& +\Delta\tilde{c}_n^T\sum_{j=1}^{n}[(z_i^Tz_j)(v_{nj} - S(\omega_n)P_{in}) - (z_i^Tv_{nj}z_j) + (z_j^TP_{in})S(\omega_n)z_i]\dot{q}_j + \\
& +\sum_{j=1}^{n} z_i^TS(\omega_n)(m_nc_nc_n^T - \hat{m}_n\hat{c}_n\hat{c}_n^T)z_j\dot{q}_j + \\
& +z_i^TS(\omega_n)(\hat{T}_n - T_n)\omega_n. \qquad (7.27)
\end{aligned}$$

The sum of the first three rows in (7.27) is the i-th component of vector

$$[J_{\hat{c}_n}^T(q)\hat{Q}_n\bar{J}_{\hat{c}_n}(q,\dot{q}) - J_{c_n}^TQ_n\bar{J}_{c_n}(q,\dot{q})]\dot{q}. \qquad (7.28)$$

Since $\bar{J}_{c_n}(q,\dot{q})$ and $\bar{J}_{\hat{c}_n}(q\dot{q})$ are very similar to $J_{c_n}(q)$ and $J_{\hat{c}_n}(q)$, the expression of (7.28) is very similar to

$$\Delta M\dot{q} = [J_{\hat{c}_n}^T(q)\hat{Q}_nJ_{\hat{c}_n}(q) - J_{c_n}^TQ_nJ_{c_n}(q)]\dot{q} \qquad (7.29)$$

whose i-th component is given by $\sum_{j=1}^{n}\Delta M_{ij}\dot{q}_j$. The only difference between (7.28) and (7.29) is that z_j in (7.29) is replaced by $\dot{z}_j$ in (7.28).

The sum of rows 4, 5 and 6 of (7.27) is the i-th component of vector

$$[J_{\hat{c}_n}^T(q)\hat{Q}_n\tilde{J}_{\hat{c}_n}(q,\dot{q}) - J_{c_n}^T(q)Q_n\tilde{J}_{c_n}(q,\dot{q})]\dot{q}$$

$$= \{\hat{m}_n[z_i \times (P_{in} + \hat{c}_n)]^T \sum_{j=1}^{n}[z_j \times (v_{nj} + \omega_n \times \hat{c}_n)] -$$
$$- m_n[z_i \times (P_{in} + c_n)]^T \sum_{j=1}^{n}[z_j \times (v_{nj} + \omega_n \times c_n)]\}\dot{q}_j. \tag{7.30}$$

A substitution of identity $(\vec{a} \times \vec{b})^T(\vec{c} \times \vec{d}) = (\vec{a}^T\vec{c})(\vec{b}^T\vec{d}) - (\vec{a}^T\vec{d})(\vec{c}^T\vec{b})$ decomposes the above expression into the sum of those three rows.

The last row of (7.27) is obviously the i-th component of vector

$$J_{\hat{c}_n}^T(q)\begin{bmatrix} 0 \\ \omega_n \times \hat{T}_n\omega_n \end{bmatrix} - J_{c_n}^T(q)\begin{bmatrix} 0 \\ \omega_n \times T_n\omega_n \end{bmatrix}. \tag{7.31}$$

According to (7.21), $\Delta V\dot{q}$ is the sum of (7.28), (7.30) and (7.31) under the assumption that only the last link inertia parameters are inaccurate. Thus the derivation of (7.27) becomes clear at this step.

The left side of (7.27) is a quadratic form while the right side of (7.27) only contains the first-order expression of $\dot{q}_j$. However, a careful reader will notice that every single term in (7.27) involves the product of $\dot{q}_j$ with either $\dot{z}_j = \omega_j \times z_j$ or ω_n, which are functions of vector $\dot{q}$. Therefore the notation $\dot{q}^T\Delta V_i\dot{q}$ as the i component of vector $\Delta V\dot{q}$ is justified.

Equations (7.24), (7.25) and (7.27) all suggest that ΔM, $\Delta V\dot{q}$ and ΔG will be reduced to zero if the inertia parameters are accurate. However, they are too complicated to use in practice. In order to estimate the inertia parameters from a proper observation signal, a more simplified expression is needed. That is the job of the next section.

7.3 Linear Parameterization

Equation (7.18) can also be expressed as

$$\tau + \tau_d = \frac{d}{dt}\{\sum_{i=1}^{n} J_{c_i}^T \begin{bmatrix} m_i v_{c_i} \\ T_i\omega_i \end{bmatrix}\} - \sum_{i=1}^{n}(J_{c_i}^T m_i \begin{bmatrix} g \\ 0 \end{bmatrix} + \dot{J}_{c_i}^T \begin{bmatrix} m_i v_{c_i} \\ T_i\omega_i \end{bmatrix}) \tag{7.32}$$

Introduce a nonlinear function

$$F(\dot{q},q) = \frac{1}{D+\sigma}\{D\sum_{i=1}^{n} J_{c_i}^T \begin{bmatrix} m_i v_{c_i} \\ T_i\omega_i \end{bmatrix} - \sum_{i=1}^{n}(m_i J_{c_i}^T(q) \begin{bmatrix} g \\ 0 \end{bmatrix} + \dot{J}_{c_i}^T \begin{bmatrix} m_i v_{c_i} \\ T_i\omega_i \end{bmatrix})\} \tag{7.33}$$

where $D = \frac{d}{dt}$ is a differential operator; $\sigma > 0$ is a constant. Substituting (7.33) into (7.32) and then comparing the result with (7.1), one will find

$$M(q)\ddot{q} + V(\dot{q},q)\dot{q} + G(q) = \frac{d}{dt}F(\dot{q},q) + \sigma F(\dot{q},q). \tag{7.34}$$

The next step is to linearize $F(\dot{q},q)$ with respect to the unknown parameter ζ_r and then design a proper estimator to identify $\zeta_m = \zeta_r$. The nonlinear function $F(\dot{q},q)$ can be linearized with respect to the system inertia parameters such that

$$F(\dot{q},q) = F_*(\dot{q},q) + \frac{\partial F(\dot{q},q)}{\partial \zeta}\zeta_r \tag{7.35}$$

where $\frac{\partial F(\dot{q},q)}{\partial \zeta} \in R^{n\times l}$ is the linearization matrix; $\zeta_r \in R^l$ is the true robot inertia parameter vector. The exact definition of ζ_r will be given a little bit later. Equation (7.35) is an exact linearization rather than a first-order approximation. Since the inertia parameters of the first $(n-1)$ links are assumed to be available, the first term $F_*(\dot{q},q)$ can be computed using the correct parameters of the first $(n-1)$ links:

$$F_*(\dot{q},q) = \frac{1}{D+\sigma}\{D\sum_{i=1}^{n-1} J_{c_i}^T \begin{bmatrix} m_i v_{c_i} \\ T_i\omega_i \end{bmatrix} - \sum_{i=1}^{n-1}(m_i J_{c_i}^T \begin{bmatrix} g \\ 0 \end{bmatrix} + \dot{J}_{c_i}^T \begin{bmatrix} m_i v_{c_i} \\ T_i\omega_i \end{bmatrix})\}. \tag{7.36}$$

Equation (7.36) differs from (7.33) in that only the dynamics of the first $(n-1)$ links are added together. The dynamics of the last link appear in the second term of (7.35) which can be computed by

$$\begin{aligned} \frac{\partial F}{\partial \zeta}\zeta_r &= \frac{1}{D+\sigma}\{DJ_{c_n}^T(q)\begin{bmatrix} m_n v_{c_n} \\ T_n\omega_n \end{bmatrix} - \dot{J}_{c_n}^T(q)\begin{bmatrix} m_n v_{c_n} \\ T_n\omega_n \end{bmatrix} - m_n J_{c_n}^T \begin{bmatrix} g \\ 0 \end{bmatrix}\} \\ &= \frac{D}{D+\sigma}A(\dot{q},q,\zeta_r) - \frac{1}{D+\sigma}B(\dot{q},q,\zeta_r) \end{aligned} \tag{7.37}$$

where

$$\begin{aligned} A(\dot{q}, q, \zeta_r) &= J_{c_n}^T \begin{bmatrix} m_n v_{c_n} \\ T_n \omega_n \end{bmatrix} \\ B(\dot{q}, q, \zeta_r) &= m_n J_{c_n}^T \begin{bmatrix} g \\ 0 \end{bmatrix} + \dot{J}_{c_n}^T \begin{bmatrix} m_n v_{c_n} \\ T_n \omega_n \end{bmatrix}. \end{aligned}$$

7.3.1 Linearizing $A(\dot{q}, q, \zeta_r)$ with respect to ζ_r

Let a_i denote the i-th component of $A(\dot{q}, q, \zeta_r)$. According to (7.37), one can write

$$a_i = z_i^T\, T_n\, \omega_n + m_n\, (z_i \times P_{in})^T v_{c_n} + m_n\, (z_i \times c_n)^T v_{c_n}. \qquad (7.38)$$

The linear velocity of the n-th link mass center consists of two parts: $v_{c_n} = v_n + \omega_n \times c_n$, where v_n is the linear velocity of the n-th link frame origin; c_n denotes the vector between the link-frame origin and the mass center of the n-th link. This decomposition separates v_n from $\omega_n \times c_n$ (which involves an unknown parameter vector c_n). It then follows that

$$\begin{aligned} m_n\, (z_i \times P_{in})^T v_{c_n} &= m_n (z_i \times P_{in})(v_n + \omega_n \times c_n) \\ &= m_n (z_i \times P_{in})(v_n + S(\omega_n) c_n) \\ &= m_n \kappa_i(\dot{q}, q) + \tilde{\eta}_i^T(\dot{q}, q) c_n m_n \end{aligned} \qquad (7.39)$$

where $\kappa_i(\dot{q}, q) = (z_i \times P_{in})^T v_n$ and $\tilde{\eta}_i^T(\dot{q}, q) = (z_i \times P_{in})^T S(\omega_n)$; $S(\omega_n)$ is a skew-symmetric matrix such that $S(\omega_n)x = \omega_n \times x\ \forall x \neq 0$. Substituting (7.39) into (7.38) yields

$$\begin{aligned} a_i &= z_i^T\, T_n\, \omega_n + m_n\, \kappa_i(\dot{q}, q) + \tilde{\eta}_i^T(\dot{q}, q)\, m_n\, c_n \\ &\quad + (z_i \times c_n)^T\, v_n\, m_n + (z_i \times c_n)^T (\omega_n \times c_n)\, m_n \end{aligned} \qquad (7.40)$$

One can re-write a tri-product term in (7.40) as

$$(z_i \times c_n)^T v_n = z_i^T (c_n \times v_n) = -z_i^T S(v_n) c_n.$$

A substitution of the above expression into (7.40) results in

$$\begin{aligned} a_i &= z_i^T\, T_n\, \omega_n + m_n\, \kappa_i(\dot{q}, q) + \eta_i^T(\dot{q}, q) m_n c_n \\ &\quad + (z_i \times c_n)^T (\omega_n \times c_n) m_n \end{aligned} \qquad (7.41)$$

where $\eta_i(\dot{q}, q) = \bar{\eta}_i(\dot{q}, q) - z_i^T S(v_n)$ is independent of the measurement parameters. The last term of (7.41) can be handled in the following way.

$$
\begin{aligned}
(z_i \times c_n)^T(\omega_n \times c_n) &= z_i^T[c_n \times (\omega_n \times c_n)] \\
&= (z_i^T\omega_n)(c_n^T c_n) - (z_i^T c_n)(c_n^T\omega_n) \\
&= (z_i^T\omega_n)(c_n^T c_n) - z_i^T(c_n c_n^T)\omega_n \qquad (7.42)
\end{aligned}
$$

where an identity $\vec{a} \times (\vec{b} \times \vec{c}) = (\vec{a}^T\vec{c})\vec{b} - (\vec{a}^T\vec{b})\vec{c}$ has been substituted.

Now, one can substitute (7.42) into (7.41) to obtain

$$
a_i = z_i^T\ \tilde{T}_n\ \omega_n + m_n\ \kappa_i(\dot{q}, q) + \eta_i^T(\dot{q}, q) m_n c_n \qquad (7.43)
$$

where $\tilde{T}_n = T_n + m_n(c_n^T c_n I_{3\times3} - c_n c_n^T)$; $I_{3\times3}$ is a 3×3 identity matrix.

It is not difficult to see that $\tilde{T}_n = {}^0_nR\ {}^n\tilde{T}_n\ {}^n_0R$ and $\omega_n = {}^0_nR\ {}^n\omega_n$ where ${}^0_nR$ is a rotation matrix that transfers from the n-th link frame to the base frame; ${}^n\tilde{T}_n$ and ${}^n\omega_n$ still keep the same physical meaning as $\tilde{T}_n$ and ω_n, but now they are expressed in the n-th link frame as their leading superscript "n" indicates. A moment's thought will convince one that ${}^n\tilde{T}_n$ is a constant matrix.

Let

$$
{}^n\omega_n = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \quad \text{and} \quad {}^n\tilde{T}_n = \begin{bmatrix} \tilde{T}_{xx} & \tilde{T}_{xy} & \tilde{T}_{xz} \\ \tilde{T}_{xy} & \tilde{T}_{yy} & \tilde{T}_{yz} \\ \tilde{T}_{xz} & \tilde{T}_{yz} & \tilde{T}_{zz} \end{bmatrix}.
$$

A key step in linearizing (7.43) is to write

$$
{}^n\tilde{T}_n\ {}^n\omega_n = \begin{bmatrix} \tilde{T}_{xx} & \tilde{T}_{xy} & \tilde{T}_{xz} \\ \tilde{T}_{xy} & \tilde{T}_{yy} & \tilde{T}_{yz} \\ \tilde{T}_{xz} & \tilde{T}_{yz} & \tilde{T}_{zz} \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \Lambda({}^n\omega_n)\rho
$$

where $\rho^T = [\tilde{T}_{xx}, \tilde{T}_{yy}, \tilde{T}_{zz}, \tilde{T}_{xy}, \tilde{T}_{xz}, \tilde{T}_{yz}]$ and

$$
\Lambda({}^n\omega_n) = \begin{bmatrix} \omega_x & 0 & 0 & \omega_y & \omega_z & 0 \\ 0 & \omega_y & 0 & \omega_x & 0 & \omega_z \\ 0 & 0 & \omega_z & 0 & \omega_x & \omega_y \end{bmatrix}.
$$

Since $\tilde{T}_n\omega_n = {}^0_nR\,{}^n\tilde{T}_n\,{}^n\omega_n = {}^0_nR\Lambda({}^n\omega_n)\rho$, one can re-write (7.43) as

$$\begin{aligned} a_i &= z_i^T\,{}^0_nR\Lambda({}^n\omega_n)\rho + m_n\kappa_i(\dot{q},q) + \eta_i^T(\dot{q},q)m_nc_n \\ &= z_i^T\,{}^0_nR\Lambda({}^n\omega_n)\rho + m_n\kappa_i(\dot{q},q) + \eta_i^T(\dot{q},q)\,{}^0_nR\,{}^nc_nm_n \end{aligned} \tag{7.44}$$

where ${}^nc_n$ is a transformed version of c_n. It is a constant vector expressed in the n-th link frame. Now, if one chooses $\zeta_r = [\rho^T, m_n, {}^n\tilde{c}_n^T]^T$ where ${}^n\tilde{c}_n = {}^nc_nm_n$, then $A(\dot{q},q,\zeta_r)$ can be linearized as

$$\begin{aligned} A(\dot{q},q,\zeta_r) &= \begin{bmatrix} z_1^T\,{}^0_nR\Lambda({}^n\omega_n) & \kappa_1(\dot{q},q) & \eta_1^T(\dot{q},q)\,{}^0_nR \\ \vdots & \vdots & \vdots \\ z_n^T\,{}^0_nR\Lambda({}^n\omega_n) & \kappa_n(\dot{q},q) & \eta_n^T(\dot{q},q)\,{}^0_nR \end{bmatrix}\zeta_r \\ \kappa_i(\dot{q},q) &= (z_i \times P_{in})^Tv_n = z_i^T(P_{in}\times v_n) \\ &= z_i^TS(P_{in})v_n; \end{aligned} \tag{7.45}$$

$$\begin{aligned} \eta_i^T(\dot{q},q) &= (z_i\times P_{in})^TS(\omega_n) - z_i^TS(v_n) \\ &= z_i^T[S(P_{in})S(\omega_n) - S(v_n)]. \end{aligned} \tag{7.46}$$

Let us introduce a matrix function

$$\Gamma(z) = \begin{bmatrix} z_1^T\,{}^0_nR\,\Lambda & z_1^TS(P_{1n})v_n & z_1^TS(P_{1n})S(\omega_n) - S(v_n)\,{}^0_nR \\ \vdots & \vdots & \vdots \\ z_n^T\,{}^0_nR\,\Lambda & z_n^TS(P_{nn})v_n & z_n^TS(P_{nn})S(\omega_n) - S(v_n)\,{}^0_nR \end{bmatrix} \tag{7.47}$$

where $z = \{z_i\}_{i=1}^n$. Now, one can write

$$A(\dot{q},q,\zeta_r) = \Gamma(z)\zeta_r. \tag{7.48}$$

Although $\Gamma(z)$ is explicitly expressed as a function of $z = \{z_i\}_{i=1}^n$, it is a matrix function of $\dot{q}$, and q as indicated by (7.47). In the next section, z will be replaced with $\dot{z} = \{\dot{z}_i\}_{i=1}^n$. The resulting matrix $\Gamma(\dot{z})$ is used to linearize $B(\dot{q},q,\zeta_r)$ with respect to ζ_r.

7.3.2 Linearizing $B(\dot{q}, q, \zeta_r)$ with respect to ζ_r

Recall from (7.26) that the time derivative of $J_{c_n}(q)$ is given by

$$\begin{aligned} \dot{J}_{c_n}(q) &= \begin{bmatrix} \dot{z}_1 \times (P_{1,n} + c_n) & \dot{z}_2 \times (P_{2,n} + c_n) & \ldots & \dot{z}_n \times c_n \\ \dot{z}_1 & \dot{z}_2 & \ldots & \dot{z}_n \end{bmatrix} \\ &+ \begin{bmatrix} z_1 \times (v_{c_n} - v_1) & z_2 \times (v_{c_n} - v_2) & \ldots & z_n \times (v_{c_n} - v_n) \\ 0 & 0 & \ldots & 0 \end{bmatrix} \\ &= \bar{J}_{c_n}(q, \dot{q}) + \tilde{J}_{c_n}(q, \dot{q}). \end{aligned}$$

It then follows that

$$\begin{aligned} B(\dot{q}, q, \zeta_r) &= m_n J_{c_n}^T \begin{bmatrix} g \\ 0 \end{bmatrix} + \tilde{J}_{c_n} \begin{bmatrix} m_n v_{c_n} \\ T_n \omega_n \end{bmatrix} + \Gamma(\dot{z})\zeta_r \\ &= C(\dot{q}, q, \zeta_r) + \Gamma(\dot{z})\zeta_r \end{aligned} \tag{7.49}$$

where the matrix $\Gamma(\dot{z})$ is the same matrix function defineded in (7.47) except that every z_i is replaced by $\dot{z}_i$;

$$C(\dot{q}, q, \zeta_r) = m_n J_{c_n}^T \begin{bmatrix} g \\ 0 \end{bmatrix} + \tilde{J}_{c_n} \begin{bmatrix} m_n v_{c_n} \\ T_n \omega_n \end{bmatrix}.$$

Since $(z_i \times v_{c_n})^T v_{c_n} = 0$, the i-th element of $C(\dot{q}, q, \zeta_r)$ can be expressed as

$$\begin{aligned} d_i &= m_n z_i^T [(P_{in} + c_n) \times g - v_i \times v_{c_n}] \\ &= m_n z_i^T [P_{in} \times g - S(g) c_n - v_i \times (v_n + \omega_n \times c_n)] \\ &= m_n z_i^T (P_{in} \times g - v_i \times v_n) - z_i^T [S(v_i) S(\omega_n) + S(g)] \, {}^0_n R \, {}^n\tilde{c}_n \\ &= m_n \alpha_i(\dot{q}, q) + \beta_i^T(\dot{q}, q) \, {}^0_n R \, {}^n\tilde{c}_n \end{aligned} \tag{7.50}$$

where

$$\alpha_i(\dot{q}, q) = z_i^T [S(P_{in}) g - S(v_i) v_n]; \tag{7.51}$$

$$\beta_i^T(\dot{q}, q) = -z_i^T [S(v_i) S(\omega_n) + S(g)]. \tag{7.52}$$

Introduce another linearizing matrix function

$$\aleph = - \begin{bmatrix} o^T & \alpha_1(\dot{q}, q) & \beta_1^T(\dot{q}, q) \, {}^0_n R \\ \vdots & \vdots & \vdots \\ o^T & \alpha_n(\dot{q}, q) & \beta_n^T(\dot{q}, q) \, {}^0_n R \end{bmatrix} \tag{7.53}$$

where $o \in R^6$ is an all-zero vector. Then one can write $C(\dot{q}, q, \zeta_r) = -\aleph\zeta_r$. Substituting into (7.49) results in

$$B(\dot{q}, q, \zeta_r) = [\Gamma(\dot{z}) - \aleph]\zeta_r. \tag{7.54}$$

By substituting (7.48) and (7.54), one may rewrite (7.37) as

$$\frac{\partial F}{\partial \zeta}\zeta_r = \frac{1}{D+\sigma}[D\Gamma(z) + \aleph - \Gamma(\dot{z})]\zeta_r. \tag{7.55}$$

And the linearization matrix can be computed by

$$\frac{\partial F}{\partial \zeta} = \frac{D}{D+\sigma}\Gamma(z) + \frac{1}{D+\sigma}[\aleph - \Gamma(\dot{z})] \tag{7.56}$$

7.4 Parameter Identification

According to (7.34), one can pass the control torque vector τ through a first order low-pass filter to obtain

$$u = \frac{1}{D+\sigma}\tau = F(\dot{q}, q) - \tilde{\tau}_d$$

where $\tilde{\tau}_d = \frac{1}{D+\sigma}\tau_d$ is the low-pass filtered version of the external disturbances. Equation (7.35) suggests that the above equation can be separated as

$$\tilde{u} = u - F_*(\dot{q}, q) = \frac{\partial F(\dot{q}, q)}{\partial \zeta}\zeta_r - \tilde{\tau}_d \tag{7.57}$$

where $F_*(\dot{q}, q)$ can be computed by (7.36) using the correct parameters of the first $(n-1)$ links. Obviously, $\tilde{u} = u - F_*(\dot{q}, q)$ and the linearization matrix function $\frac{\partial F}{\partial \zeta}$ are available to the controller. Thus (7.57) fits the typical framework of a least square estimator. The estimator will try to minimize

$$\begin{aligned}\int_{t_0}^{t_1} \|\tilde{u} - \frac{\partial F}{\partial \zeta}\zeta_m\|^2 dt &= \int_{t_0}^{t_1} \|u - F_*(\dot{q}, q) - \frac{\partial F}{\partial \zeta}\zeta_m\|^2 dt \\ &= \int_{t_0}^{t_1} \|F(\dot{q}, q) - F_m(\dot{q}, q) - \tilde{\tau}_d\|^2 dt \\ &= \int_{t_0}^{t_1} \|\frac{1}{D+\sigma}(r + \tau_d)\|^2 dt \end{aligned} \tag{7.58}$$

where (7.57) has been substituted and

$$r = \Delta M\ddot{q} + \Delta V\dot{q} + \Delta G.$$

When $\tau_d = 0$, (7.58) becomes a zero-square problem because there exists a $\zeta_m = \zeta_r$ such that $r = 0$.

7.4.1 Identification under ideal conditions

When the external disturbance is $\tau_d = 0$, the minimization of (7.58) becomes a search for ζ_m such that

$$\tilde{u} - \frac{\partial}{\partial\zeta}F(\dot{q}, q)\zeta_m = \frac{\partial}{\partial\zeta}F(\dot{q}, q)\Delta\zeta = 0$$

where $\Delta\zeta = \zeta_r - \zeta_m$ denotes the estimate error. The above expression can be written as

$$\Delta\tilde{u} = \tilde{u} - \frac{\partial F}{\partial\zeta}\zeta_m = \frac{\partial F}{\partial\zeta}\Delta\zeta. \tag{7.59}$$

Introducing a Lyapunov function candidate $L = 0.5\Delta\zeta^T\Delta\zeta$, the time derivative of L is given by $\dot{L} = -\Delta\zeta^T\dot{\zeta}$ because ζ_r is a constant parameter vector. Now, it is convenient to design an adaptation law for ζ_m such that

$$\dot{\zeta}_m = \frac{\partial F^T}{\partial\zeta}\Delta\tilde{u} \tag{7.60}$$

where $\frac{\partial F^T}{\partial\zeta}$ is the transposition of matrix $\frac{\partial F}{\partial\zeta}$. By substituting (7.60) and then (7.59), one will find that

$$\dot{L} = -\Delta\zeta^T\frac{\partial F^T}{\partial\zeta}\Delta\tilde{u} = -\Delta\zeta^T\frac{\partial F^T}{\partial\zeta}\frac{\partial F}{\partial\zeta}\Delta\zeta \leq 0. \tag{7.61}$$

The above expression implies that $L = 0.5||\Delta\zeta||^2$ is a non-increasing function. In many cases, it will decrease to zero within a finite time.

The algorithm of (7.60) actually solves a perfect matching problem (7.59) instead of a least square one. In real applications, the external disturbance τ_d is inevitable. One must be very careful in applying (7.60). When $\tau_d \neq 0$, (7.58) should be reduced to

$$\Delta\tilde{u} = \tilde{u} - \frac{\partial F}{\partial \zeta}\zeta_m = \frac{\partial F}{\partial \zeta}\Delta\zeta - \tilde{\tau}_d \tag{7.62}$$

instead of (7.59). Accordingly, (7.61) becomes

$$\dot{L} = -\Delta\zeta^T \frac{\partial F^T}{\partial \zeta}\Delta\tilde{u} = -\Delta\zeta^T \frac{\partial F^T}{\partial \zeta}\frac{\partial F}{\partial \zeta}\Delta\zeta - \Delta\zeta^T \frac{\partial F^T}{\partial \zeta}\tilde{\tau}_d \tag{7.63}$$

which is not necessarily semi-negative definite. The convergence of $L = 0.5\|\Delta\zeta\|^2$ is no lon‿'r guaranteed.

7.4.2 Identification in the presence of external disturbances

Introduce vectors

$$\mathcal{Y}^T = [\lambda^k \tilde{u}^T(t_o), \lambda^{k-1}\tilde{u}^T(t_o+\delta), \ldots, \lambda\tilde{u}(t_o+(k-1)\delta), \tilde{u}^T(t_o+k\delta)],$$
$$\mathcal{X}^T = [\lambda^k \frac{\partial}{\partial \zeta}F^T(t_o), \ldots, \lambda\frac{\partial}{\partial \zeta}F^T(t_o+(k-1)\delta), \frac{\partial}{\partial \zeta}F^T(t_o+k\delta)]$$

where $0 < \lambda \leq 1$ is a scalar forgetting factor; $\tilde{u}(t_o+i\delta)$ and $\frac{\partial}{\partial \zeta}F(t_o+i\delta)$ are the samples of $\tilde{u}$ and $\frac{\partial F}{\partial \zeta}$ at the i-th sampling instant respectively. Then the least square problem of (7.58) is equivalent to

$$\min_{\zeta_m}\{\|\mathcal{Y} - \mathcal{X}\zeta_m\|^2\}$$

with $\lambda = 1$. An explicit expansion of the above expression is given by

$$\begin{aligned} \|\mathcal{Y} - \mathcal{X}\zeta_m\|^2 &= \mathcal{Y}^T\mathcal{Y} + \zeta_m^T\mathcal{X}^T\mathcal{X}\zeta_m - 2\mathcal{Y}^T\mathcal{X}\zeta_m \\ &= \mathcal{Y}^T\mathcal{Y} + \zeta_m^T\mathcal{A}\zeta_m - 2\mathcal{Z}^T\mathcal{A}\zeta_m \\ &= \mathcal{Y}^T\mathcal{Y} - \mathcal{Z}^T\mathcal{Z} + (\zeta_m - \mathcal{Z})^T\mathcal{A}(\zeta_m - \mathcal{Z}). \end{aligned} \tag{7.64}$$

where $\mathcal{Z} = \mathcal{A}^{-1}\mathcal{X}^T\mathcal{Y}$ and

$$\mathcal{A} = \mathcal{X}^T\mathcal{X} = \sum_{i=0}^{k}\frac{\partial}{\partial \zeta}F^T(t_o+i\delta)\frac{\partial}{\partial \zeta}F(t_o+i\delta)\lambda^{2(k-i)}. \tag{7.65}$$

In order to arrive at (7.64), $\mathcal{A}$ as defined in (7.65) must be positive definite and hence $\mathcal{A}^{-1}$ exists. It should be emphasized that the existence of $\mathcal{A}^{-1}$ does not necessarily mean the existence of $[\frac{\partial F^T}{\partial \zeta}\frac{\partial F}{\partial \zeta}]^{-1}$ at each individual sampling instant. In fact, even if $\frac{\partial F^T}{\partial \zeta}\frac{\partial F}{\partial \zeta}$ is singular at all sampling instants, it is still possible that $\mathcal{A}^{-1}$ exists. The larger k is, the more likely that $\mathcal{A}^{-1}$ exists.

Since

$$\mathcal{Z} = \mathcal{A}^{-1}\mathcal{X}^T\mathcal{Y} = [\mathcal{X}^T\mathcal{X}]^{-1}\mathcal{X}^T\mathcal{Y} \tag{7.66}$$

is independent of ζ_m, the first two terms in (7.64) can not be controlled by adjusting ζ_m. The third term of (7.64) is positive definite. The only possible way to minimize (7.64) is $\zeta_m = \mathcal{Z}$ which solves the least square problem of (7.58). There are many computational efficient algorithms to compute (7.66) recursively without actually storing the samples of $\tilde{u}$ and $\frac{\partial F}{\partial \zeta}$. A detailed discussion of these algorithms is beyond the scope of this study.

The introduction of forgetting factor $0 < \lambda \leq 1$ will prevent the matrix $\mathcal{A}$ from overflow due to the accumulation of data samples of $\frac{\partial F}{\partial \zeta}$. Equation (7.65) is actually a discrete convolution of matrix $\frac{\partial F^T}{\partial \zeta}\frac{\partial F}{\partial \zeta}$ with a scalar function $f(k) = \lambda^k$, $0 \leq k$. The matrix $\mathcal{A}$ is bounded as long as $\lambda < 1$ and $\dot{q}$, q are bounded (which has been proved in Section 2).

7.4.3 The QR algorithm

According to the discussions of the previous section, $\mathcal{X}$ depends on the robot trajectory $\dot{q}(t)$ and $q(t)$ whose samples at time instants $t_o + i\delta$, $0 \leq i \leq k$ are used to compute the sub-matrix $\frac{\partial}{\partial \zeta}F(t_o + i\delta)$. It is very difficult to check the rank of $\mathcal{X}$ which is at most $10 = \min\{(k+1)n, 10\}$ (provided that $(k+1)n > 10$). In case $\mathcal{X}$ is not of full rank ($\text{rank}(\mathcal{X}) < \min\{(k+1)n, 10\}$), an alternative method is sought to determine a set of admissible ζ_m.

The mathematical formula is still $\min_{\zeta_m}\{\|\mathcal{Y} - \mathcal{X}\zeta_m\|^2\}$. Yet the problem is how to solve ζ_m when $\mathcal{X}$ is rank deficient (not of full rank). In numerical analysis theory, there is a very efficient way to deal with this kind of problem. It is called the QR algorithm [70]. For any arbitrary $l \times n$ matrix $\mathcal{M}$ where $l > n$, there exists an $l \times l$ orthogonal matrix $\mathcal{Q}$ and an $l \times n$ upper triangular matrix $\mathcal{R}$ such that $\mathcal{QR} = \mathcal{M}$. The upper triangular matrix $\mathcal{R}$ has two possible forms, depending on the rank of $\mathcal{M}$,

$$\mathcal{R} = \begin{bmatrix} * & * & * & \dots & * \\ 0 & * & * & \dots & * \\ 0 & 0 & * & \dots & * \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & * \\ 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix} \quad \text{or} \quad \mathcal{R} = \begin{bmatrix} * & * & * & \dots & * & \dots & * \\ 0 & * & * & \dots & * & \dots & * \\ 0 & 0 & * & \dots & * & \dots & * \\ \vdots & \vdots & \vdots & \ddots & \vdots & & \vdots \\ 0 & 0 & 0 & \dots & * & \dots & * \\ 0 & 0 & 0 & \dots & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots & & \vdots \\ 0 & 0 & 0 & \dots & 0 & \dots & 0 \end{bmatrix} \tag{7.67}$$

where "$*$" represents a non-zero entry; the first form indicates that $\mathcal{M}$ is of full rank and the second form means that $\mathcal{M}$ is rank deficient. In both cases, one can always write $\mathcal{R}^T = [\mathcal{R}_1^T, \mathcal{O}]$ where $\mathcal{R}_1$ denotes the l_1 non-zero rows of $\mathcal{R}$ while $\mathcal{O}$ is an $(l - l_1) \times n$ all-zero submatrix.

If the orthogonal matrix $\mathcal{Q}$ with respect to $\mathcal{X}$ is available, then by the fact that $\mathcal{Q}^T\mathcal{Q} = I$, one can write

$$\mathcal{Q}^T\mathcal{Y} - \mathcal{Q}^T\mathcal{X}\zeta_m = \mathcal{V} - \mathcal{R}\zeta_m \tag{7.68}$$

where $\mathcal{Y}^T\mathcal{Q} = \mathcal{V}^T = [\mathcal{V}_1^T, \mathcal{V}_2^T]$; $\mathcal{V}_1 \in R^{l_1}$ corresponds to the l_1 non-zero rows of matrix $\mathcal{R}$. The fact that an orthogonal matrix $\mathcal{Q}$ preserves length enables one to write

$$\|\mathcal{Y} - \mathcal{X}\zeta_m\|^2 = \|\mathcal{V} - \mathcal{R}\zeta_m\|^2 = \|\mathcal{V}_1 - \mathcal{R}_1\zeta_m\|^2 + \|\mathcal{V}_2\|^2.$$

No matter what ζ_m vector is applied to (7.68), the vector $\mathcal{R}\zeta_m$ can at most cancel out $\mathcal{V}_1$ while leaving $\mathcal{V}_2$ unaffected. Therefore the best thing one can achieve by selecting ζ_m is

$$\mathcal{V}_1 = \mathcal{R}_1\zeta_m \tag{7.69}$$

which minimizes $\|\mathcal{Y} - \mathcal{X}\zeta_m\|^2$.

The solution of (7.69) is straightforward if $l_1 = 10$. When $l_1 < 10$, there are infinitely many solutions that satisfy (7.69). One convenient way is to set the last

$(10 - l_1)$ components of ζ_m to zero and solve the first l_1 components of ζ_m from (7.69). The rank of $\mathcal{R}_1$ can be easily checked out by examining the number of its non-zero rows.

7.4.4 The Householder transforms

An important step is to find $\mathcal{Q}$ for an arbitrary $l \times n$ matrix $\mathcal{M}$. The first column of $\mathcal{M}$ is denoted as m_1. Consider a vector $b_1 = m_1 + \|m_1\|e_1$ where $e_1^T = [1, 0, \ldots, 0]$ and an $l \times l$ matrix

$$\mathcal{H}_1 = I - \frac{2}{\|b_1\|^2} b_1 b_1^T.$$

A straightforward calculus will verify that $\mathcal{H}_1^T \mathcal{H}_1 = \mathcal{H}_1 \mathcal{H}_1 = I$ which means that $\mathcal{H}_1$ is an orthogonal matrix. It is interesting to notice that

$$\begin{aligned} \mathcal{H}_1 m_1 &= m_1 - \frac{2b_1^T m_1}{\|b_1\|^2} b_1 \\ &= m_1 - (\|m_1\|e_1 + m_1) \frac{2(\|m_1\|e_1^T m_1 + \|m_1\|^2)}{2(\|m_1\|e_1^T m_1 + \|m_1\|^2)} \\ &= -\|m_1\|e_1 = -[\|m_1\|, 0, \ldots, 0]^T! \end{aligned}$$

This means that $\mathcal{H}_1$ has the effect of concentrating all "energy ($\|m_1\|$)" of the vector m_1 to a single coordinate while leaving the rest of the coordinates as zero. Now, examine matrix $\mathcal{M}_1 = \mathcal{H}_1 \mathcal{M}$, it must look like either

$$\mathcal{M}_1 = \begin{bmatrix} * & * & \ldots & * \\ 0 & 0 & \ldots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \ldots & 0 \end{bmatrix} \quad \text{or} \quad \mathcal{M}_1 = \begin{bmatrix} * & * & \ldots & * \\ 0 & * & \ldots & * \\ \vdots & \vdots & & \vdots \\ 0 & * & \ldots & * \end{bmatrix}.$$

The first case happens if all columns of $\mathcal{M}$ are parallel to m_1; in most cases, $\mathcal{M}_1$ has the second form where the problem has been reduced to a smaller $(l-1) \times (n-1)$ submatrix $\tilde{\mathcal{M}}_1$. The relation between $\mathcal{M}_1$ and $\tilde{\mathcal{M}}_1$ can be visually expressed as follows:

$$\mathcal{M}_1 = \begin{bmatrix} * & row_1 \\ 0_1 & \tilde{\mathcal{M}}_1 \end{bmatrix}$$

where $row_1 \in R^{(n-1)}$ denotes a nonzero sub-row and $0_1 \in R^{(l-1)}$ denotes an all zero sub-column. By repeating the above procedures, one can construct a second $(l-1) \times (l-1)$ Householder transform matrix $\tilde{\mathcal{H}}_2$ such that

$$\tilde{\mathcal{M}}_2 = \tilde{\mathcal{H}}_2 \tilde{\mathcal{M}}_1 = \begin{bmatrix} * & row_2 \\ 0_2 & \mathcal{G} \end{bmatrix}$$

where, again, "*" denotes a non-zero entry; $row_2 \in R^{(n-2)}$ a non-zero sub-row and $0_2 \in R^{(l-2)}$ an all-zero sub-column; $\mathcal{G}$ is the remaining matrix whose dimension is $(l-2) \times (n-2)$.

The two Householder transform matrices can be concatenated together by introducing an orthogonal matrix

$$\mathcal{H}_2 = \begin{bmatrix} 1 & 0_1^T \\ 0_1 & \tilde{\mathcal{H}}_2 \end{bmatrix}.$$

It then follows that

$$\mathcal{H}_2 \mathcal{H}_1 M = \begin{bmatrix} * & * & row_{21} \\ 0 & * & row_{22} \\ 0_2 & 0_2 & \mathcal{G} \end{bmatrix}$$

where row_{21} and row_{22} are two non-zero $(n-2)$-dimensional vectors. By induction, one can follow the same procedure to construct a series of Householder transforms $\mathcal{H}_i$, $1 \leq i \leq n$ such that

$$\mathcal{Q}^T \mathcal{M} = \mathcal{H}_n \mathcal{H}_{n-1} \ldots \mathcal{H}_2 \mathcal{H}_1 \mathcal{M} = \mathcal{R}$$

where $\mathcal{R}$ is the desired upper triangular matrix given in (7.67). Using the fact that $\mathcal{H}_i^T \mathcal{H}_i = \mathcal{H}_i \mathcal{H}_i = I$ and that

$$\mathcal{Q} = (\mathcal{H}_n \mathcal{H}_{n-1} \ldots \mathcal{H}_2 \mathcal{H}_1)^T = \mathcal{H}_1^T \mathcal{H}_2^T \ldots \mathcal{H}_{n-1}^T \mathcal{H}_n^T,$$

one can easily verify that $\mathcal{Q}^T \mathcal{Q} = I$ and the computation of the orthogonal matrix $\mathcal{Q}$ with respect to an arbitrary $l \times n$ matrix $\mathcal{M}$ becomes clear.

The construction procedure for the orthogonal matrix $\mathcal{Q}$ looks very complicated. In fact, there are many recursive algorithms to compute $\mathcal{Q}$ for $\mathcal{X}$ without saving all the samples of $\frac{\partial F}{\partial \zeta}$ [68]. These are topics beyond the scope of this study and will not be discussed here.

7.5 A Simulation Example

In order to demonstrate the construction of $\frac{\partial F}{\partial \zeta}$, one may consider a two-link planar robot as an example. The three coordinate frames are illustrated in Fig. 7.1. In this case, the three z-axis are always parallel. That is $z_1 = z_2 = [0, 0, 1]^T$; consequently, $\dot{z}_1 = \dot{z}_2 = [0, 0, 0]^T$. The rotation matrix can be easily derived as

$$ {}^0_2R = \begin{bmatrix} \cos(q_1+q_2) & -\sin(q_1+q_2) & 0 \\ \sin(q_1+q_2) & \cos(q_1+q_2) & 0 \\ 0 & 0 & 1 \end{bmatrix}. $$

The other variables necessary for $\frac{\partial F}{\partial \zeta}$ are given by

$$ {}^2\omega_2 = \omega_2 = (\dot{q}_1 + \dot{q}_2)[0, 0, 1]^T, $$

$$ v_1 = [0, 0, 0]^T, \qquad v_2 = \dot{q}_1 l_1[-\sin(q_1), \cos(q_1), 0]^T, $$

$$ P_{12} = l_1[\cos(q_1), \sin(q_1), 0]^T \text{ and } P_{22} = [0, 0, 0]^T. $$

The above variables are substituted into (7.44) to obtain a_1 and a_2. In this case, it is not difficult to obtain

$$ z_1^T\, {}^0_2R\, \Lambda = z_2^T\, {}^0_2R\, \Lambda = [0, 0, (\dot{q}_1 + \dot{q}_2), 0, 0, 0]. \tag{7.70} $$

The other two terms in (7.44) can be computed using (7.45) and (7.46) as

$$ \kappa_1(\dot{q}, q) = z_1^T(P_{12} \times v_2) = \dot{q}_1 l_1^2 \qquad \kappa_2(\dot{q}, q) = 0, \tag{7.71} $$

$$ \begin{aligned} \eta_1(\dot{q}, q) &= z_1^T[S(P_{12})S(\omega_2) - S(v_2)] \\ &= [P_{12x}\omega_{2z} + v_{2y},\ P_{12y}\omega_{2z} - v_{2x},\ -(P_{12y}\omega_{2y} + P_{12x}\omega_{2x})] \\ &= (2\dot{q}_1 + \dot{q}_2)l_1[\cos(q_1),\ \sin(q_1),\ 0], \end{aligned} \tag{7.72} $$

and

$$ \begin{aligned} \eta_2(\dot{q}, q) &= [P_{22x}\omega_{2z} + v_{2y},\ P_{22z}\omega_{2z} - v_{2x},\ -(P_{22y}\omega_{2y} + P_{22x}\omega_{2x})] \\ &= \dot{q}_1 l_1[\cos(q_1),\ \sin(q_1),\ 0]. \end{aligned} \tag{7.73} $$

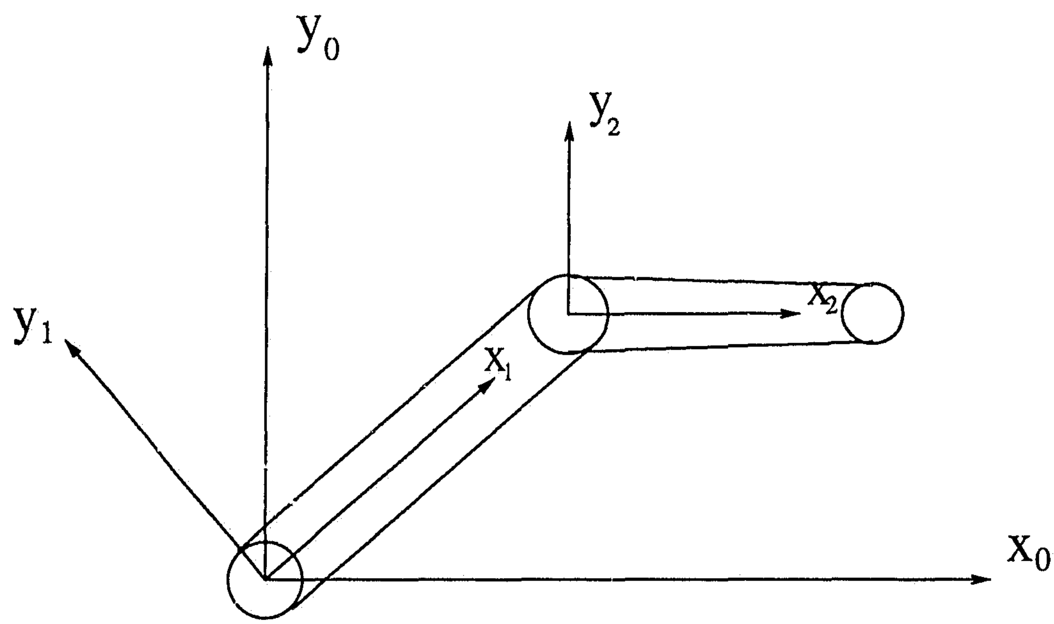

Figure 7.1: A two link planar robot.

From (7.72) and (7.73), one further obtains

$$\eta_1 \, {}^0_2R = (2\dot{q}_1 + \dot{q}_2)l_1[\cos(q_2), \ -\sin(q_2), \ 0], \tag{7.74}$$

$$\eta_2 \, {}^0_2R = \dot{q}_1 l_1[\cos(q_2), \ -\sin(q_2), \ 0]. \tag{7.75}$$

By substituting (7.70), (7.71), (7.74) and (7.75) into (7.48), one can write

$$A(\dot{q}, q, \zeta_m) = \begin{bmatrix} (\dot{q}_1 + \dot{q}_2) & \dot{q}_1 l_1^2 & (2\dot{q}_1 + \dot{q}_2)l_1\cos(q_2) & -(2\dot{q}_1 + \dot{q}_2)l_1\sin(q_2) \\ (\dot{q}_1 + \dot{q}_2) & 0 & \dot{q}_1 l_1 \cos(q_2) & -\dot{q}_1 l_1 \sin(q_2) \end{bmatrix} \zeta_m \tag{7.76}$$

where $\zeta_m^T = [\tilde{T}_{2zz}, m_2, m_2c_{2x}, m_2c_{2y}]$ has been reduced to a 4-dimensional vector because of the planar configuration. Physically, this simplification is straightforward because the other inertia parameters, such as $\tilde{T}_{xx}$ and c_{2z}, do not contribute any torques to the two joints which rotate about the z_0 axis.

Next, one may repeat the same way to derive

$$\begin{aligned}\alpha_1(\dot{q},q) &= z_1^T[S(P_{12})g - S(v_1)v_2] \\ &= z_1^T\begin{bmatrix} 0 & -P_{12z} & P_{12y} \\ P_{12z} & 0 & -P_{12x} \\ -P_{12y} & P_{12x} & 0 \end{bmatrix}\begin{bmatrix} 0 \\ -9.8 \\ 0 \end{bmatrix} \\ &= -9.8 l_1\cos(q_1), \end{aligned} \tag{7.77}$$

$$\alpha_2(\dot{q},q) = 0, \tag{7.78}$$

$$\beta_1(\dot{q},q) = -z_1^T[S(v_1)S(\omega_2) - S(g)] = -[9.8, 0, 0], \tag{7.79}$$

$$\begin{aligned}\beta_2(\dot{q},q) &= -z_2^T[S(v_2)S(\omega_2) - S(g)] \\ &= -[9.8 - (\dot{q}_1+\dot{q}_2)\dot{q}_1 l_1\sin(q_1),\ (\dot{q}_1+\dot{q}_2)\dot{q}_1 l_1\cos(q_1),\ 0]. \end{aligned} \tag{7.80}$$

Again, eqs. (7.79) and (7.80) leads to

$$\beta_1\,{}^0_2R = -[9.8\cos(q_1+q_2), -9.8\sin(q_1+q_2),\ 0] \tag{7.81}$$

$$\begin{aligned}\beta_2\,{}^0_2R &= -\beta_1\,{}^0_2R - [(\dot{q}_1+\dot{q}_2)\dot{q}_1 l_1\sin(q_2),\ (\dot{q}_1+\dot{q}_2)\dot{q}_1 l_1\cos(q_2),\ 0] \\ &= -[\tilde{\beta}_{2x},\ \tilde{\beta}_{2y},\ 0]\end{aligned} \tag{7.82}$$

where $\tilde{\beta}_{2x}$ and $\tilde{\beta}_{2y}$ are the x-y components of the row vector given by (7.82).

Now, one can substitute (7.77), (7.78), (7.81) and (7.82) into (7.53) and (7.54). In this particular case, $\dot{z}_1$ and $\dot{z}_2$ are zero vectors, which implies that $\Gamma(\dot{z})$ is an all-zero matrix. Consequently, $B(\dot{q},q,\zeta_m) = C(\dot{q},q,\zeta_m) = -\aleph\zeta_m$ which can be expressed as

$$B(\dot{q},q,\zeta_m) = -\begin{bmatrix} 0 & 9.8l_1\cos(q_1) & 9.8\cos(q_1+q_2) & -9.8\sin(q_1+q_2) \\ 0 & 0 & \tilde{\beta}_{2x} & \tilde{\beta}_{2y} \end{bmatrix}\zeta_m. \tag{7.83}$$

The dynamics due to the last link movement can be derived by

$$\begin{bmatrix} \tau_{21} \\ \tau_{22} \end{bmatrix} = \frac{d}{dt}A(\dot{q},q,\zeta_m) - B(\dot{q},q,\zeta_m) \tag{7.84}$$

while the dynamics contributed by the first link are given by

$$\tau_* = \begin{bmatrix} \tau_{1*} \\ 0 \end{bmatrix} = \begin{bmatrix} m_1(z_1 \times c_1)^T \frac{d}{dt}(v_1 + \omega_1 \times c_1) + z_1^T[\frac{d}{dt}(T_1\omega_1) + m_1(c_1 \times g_1)] \\ 0 \end{bmatrix}$$
$$= \begin{bmatrix} \ddot{q}_1[T_{1zz} + m_1(c_{1x}^2 + c_{1y}^2)] + 9.8m_1[c_{1x}\cos(q_1) - c_{1y}\sin(q_1)] \\ 0 \end{bmatrix}. \qquad (7.85)$$

In the case of ideal point mass located on the end of the two links, T_1 and T_2 are all zero matrices; $c_{1x} = l_1$, $c_{1y} = 0$, $c_{2x} = l_2$ and $c_{2y} = 0$. One can verify that

$$\tau = \tau_* + \frac{d}{dt}A(\dot{q}, q, \zeta_m) + B(\dot{q}, q, \zeta_m) = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix}$$

where

$$\begin{aligned} \tau_1 &= [(2l_1\cos(q_2) + l_2)l_2m_2 + l_1^2(m_1 + m_2)]\ddot{q}_1 + [l_2^2m_2 + l_1l_2\cos(q_2)m_2]\ddot{q}_2 \\ &\quad -2l_1l_2m_2\sin(q_2)\dot{q}_2\dot{q}_1 - l_1l_2m_2\sin(q_2)\dot{q}_2^2 \\ &\quad +g(m_2l_2\cos(q_1 + q_2) + (m_1 + m_2)l_1\cos(q_1)) \end{aligned} \qquad (7.86)$$

$$\begin{aligned} \tau_2 &= [l_2^2m_2 + l_1l_2\cos(q_2)m_2]\ddot{q}_1 + l_2^2m_2\ddot{q}_2 + l_1l_2m_2\sin(q_2)\dot{q}_1^2 \\ &\quad +m_2l_2g\cos(q_1 + q_2) \end{aligned} \qquad (7.87)$$

The above equation describes the dynamics of a typical two-link planar robot under the ideal point mass assumption. A computer simulation experiment is conducted to test the self-tuning controller and the parameter estimator. The robot parameters are given by $m_1 = 10kg$, $m_2 = 5kg$, $l_1 = c_{1x} = 0.7M$ and $l_2 = c_{2x} = 0.5M$.

In order to test the adaptive controller, the two inertia parameters m_2 and c_{2x} for the second the link are fictitiously assumed to be not available to the controller. They are supposed to be identified by the parameter estimator.

The controller is synthesized by (7.2) with $K_v = 40I$ and $K_p = 400I$; the nonlinear feedback term $\hat{M}\ddot{q}_d + \hat{V}\dot{q} + \hat{G}$ is computed by (7.86) and (7.87) with $\hat{m}_2$ and $\hat{c}_{2x}$ being substituted by their estimated values instead of the true values.

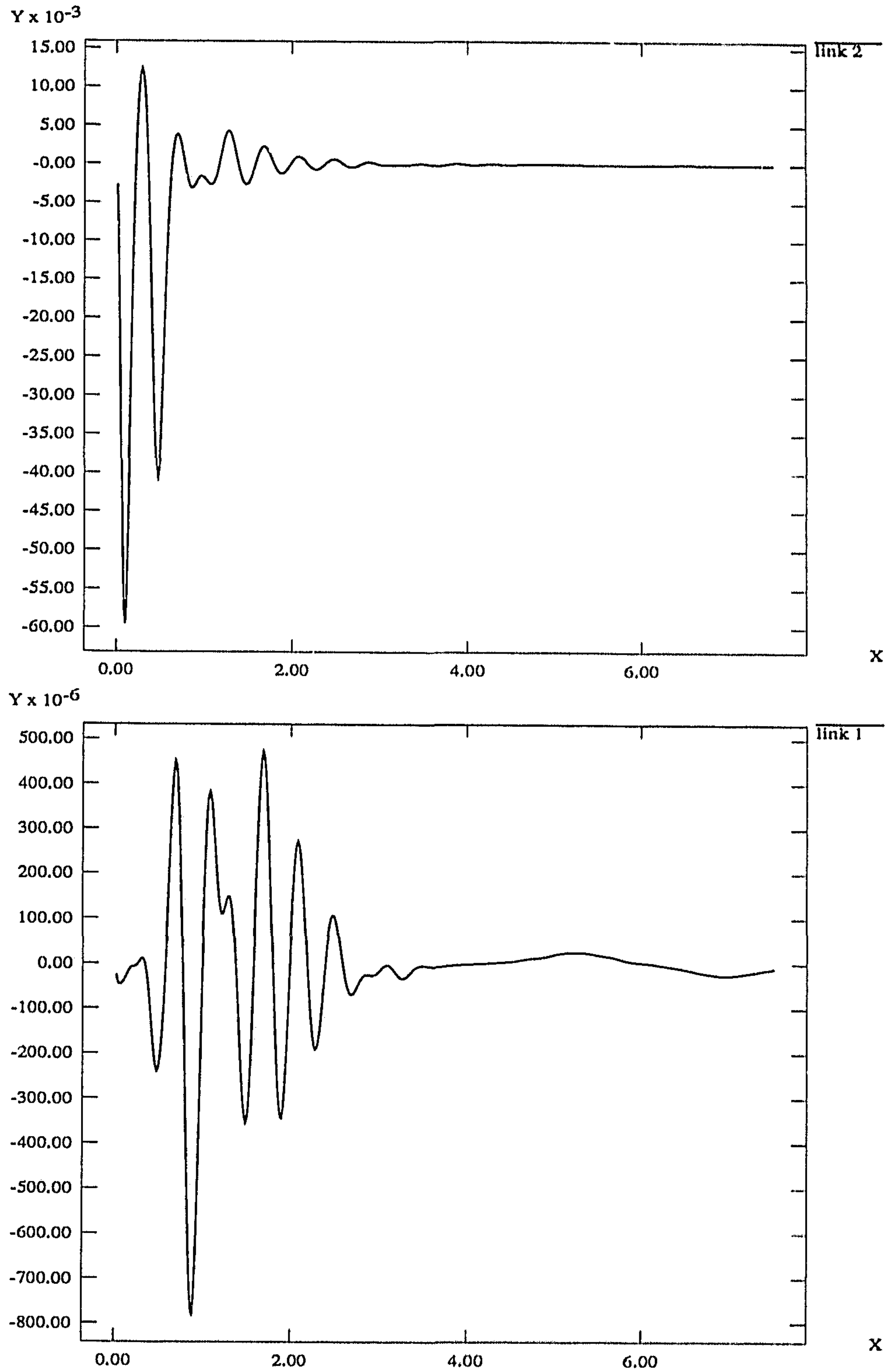

Figure 7.2: Performance of the self-tuning controller.

(These two parameters are initially set to zero.) The desired trajectories are given by $q_{d1} = q_{d2} = 1 - \cos(0.5\pi t)$. The tracking errors are plotted in Fig. 7.2. The first link has a much smaller tracking error because its inertia parameters are known. The tracking error of the second link converges very fast into practically zero.

7.6 Summary

The robustness of computed-torque controllers is studied in this chapter. These controllers are proven to be able to tolerate a certain degree of parameter uncertainties and external disturbances when the feedback gains are sufficiently large. A reduced-order regressor is derived for a general n-link robotic manipulator. The new regressor is derived by low-pass filtering the robotic dynamic equation. It requires minor increasing of computations compared with the traditional Newton-Euler algorthm. The use of a regressor enables one to express the robot dynamics linearly in terms of inertia parameters. This makes it possible to design adaptive algorithms to estimate the inertia parameters according to the feedback measurement. The combination of the two results is a robust indirect adaptive controller for robotic manipulators. Simulation results are presented to demonstrate the performance of the adaptive controller.

Chapter 8

Conclusions

This study addresses the problem of adaptive control of a class of NTV plants such as rigid body robotic manipulators. The dynamics of these plants are described by some nonlinear differential equations with varying coefficients.

The most common approach to control such plants is to compute and compensate part of the nonlinear dynamic effects. However, it is generally very difficult, if not impossible, to compute the exact nonlinear coefficients of a dynamic equation because the coefficients themselves depend on some system parameters which are poorly available to the controller. For example, the coefficient matrices of a rigid body robot dynamics depend on the inertia parameters of all links. The inertia parameters of the last link are generally inaccurate because they change with the unknown payload. In the control community, the inaccurate compensation of plant coefficients is called parameter uncertainty. Another difficulty of controlling a NTV plant is the effect of external disturbances. In many cases, the external disturbances include all possible dynamics which are conveniently neglected when one tries to build a mathematical model for a given plant. Adaptive controllers are intended to control NTV plants while being robust with respect to both parameter uncertainty and external disturbances.

Two different types of adaptive controllers are studied. The first one requires minimum knowledge about the plant dynamics. They will provide robust tracking as long as the order of the dynamic equation and the plant states are available to the controller. Design procedures and stability analysis of these adaptive controllers are given in Chapters 5 and 6. The second type of adaptive controllers require some detailed knowledge. They are plant dependent. When applied to robotic manipulator control, these adaptive controllers require a regressor to linearize the dynamics with respect to a properly defined system parameters. An innovative algorithm for the computation of a regressor for a general n-link robot is developed. Based on the algorithm, a robust controller is designed, which provides stable tracking under both parameter uncertainty and external disturbances. The robustness of the controller allows the adaptive law sufficient time to estimated the inertia parameters and optimize the closed-loop system performance. A detailed discussion of the adaptive controller is given in Chapter 7.

The adaptive controllers developed in this study are only suitable for LTV or NTV plants with complete state feedbacks. Adaptive control of LTV/NTV plants with incomplete state feedbacks remains a challenging topic. Particularly, robust control of robotic manipulators without directly measuring the joint velocities is being studied by many researchers. Adaptive control of flexible robots is another topic to be further explored.

Bibliography

[1] R.C. Paul, "Modeling, Trajectory Calculation and Servoing of a Computer Controlled Arm", *AIM-177, Stanford University Artificial Intelligence Laboratory*, 1972.

[2] A. K. Bejczy, "Robot Arm Dynamics and Control," *JPL NASA Technical Memor.* 33-669, Feb. 1974.

[3] R. A. Lewis, "Autonomous Manipulation on a Robot: Summary of Manipulator Software Functions," *JPL Technical Memor.* 33-679, Mar. 1974.

[4] B. R. Markiewicz, "Analysis of the Computed Torque Drive Method and Comparison with Conventional Position Servo for a Computer-Controlled Manipulator," *JPL technical Memor.* 33-601, Mar. 1973.

[5] J.Y.S. Luh, M.W. Walker and R. P. Paul, "Resolved-Acceleration Control of Mechanical Manipulators," *IEEE Trans. Auto. Contr.* 1980; AC-25(3).

[6] J. Y. S. Luh, M. W. Walker and R. P. Paul, "On-Line Computation Scheme for Mechanical Manipulators," *Trans. ASME J. Dynam. Syst. Meas. Contr.* 1980.

[7] Y. Stepanenko, "State-Feedback Robotic Control" In R.C. Dorf (Ed.) *Encyclopedia of Robotics*, John Wiley and Sons, New York, 1988, pp. 1667-1684.

[8] Y. Stepanenko and T.S. Sankar, "Vibro-Impact Analysis of Control Systems with Mechanical Clearance and Application to Robotic Actuators", Trans.

of ASME, *J. Dynamic Systems, Measurement and Control*, vol. 108, pp. 9-16, 1986.

[9] J. Reed and P.A. Ioannou, "Instability Analysis and Robust Adaptive Control of Robotic Manipulators", *IEEE Trans. Robotics and Automation*, vol. 5, no. 3, June 1989.

[10] K.J. Astrom, "Theory and applications of adaptive control - a survey", *Automatica*, vol. 19, no. 5, pp 471-486, 1983.

[11] M.A. Duarte and K. S. Narendra, "A new Approach To Model Reference Adaptive Control", Int. J. Adaptive Control and Signal Processing, Vol. 3, pp. 53-73, 1989.

[12] G. Kreisselmeier and K. S. Narendra, "Stable Model Reference Adaptive Control in the Presence of Bounded Disturbances," *IEEE Trans. Auto. Contr.*, vol. AC-27, pp. 1161–1168, Dec. 1982

[13] B.D.O. Anderson, "Exponential Stability of Linear Equations Arising in Adaptive Identification," *IEEE Trans. Auto. Contr.*, 1977; AC-22: pp. 83–88.

[14] P.A. Ioannou and P.V. Kokotovic, "Instability Analysis and Improvement of Robustness of Adaptive Control," *Automatica*, vol. 20, pp. 583–594, Sept. 1984

[15] K.S. Narendra and A. M. Annaswamy, "A New Adaptive Law for Robust Adaptation Without Persistent Excitation", *IEEE Trans. Automat. Control*, vol. 32, no. 2, pp. 134-145, 1987.

[16] Y.D. Landau, "*Adaptive Control: The Model Reference Approach,*" Marcel Dekker, N.Y., 1979.

[17] E.D. Sontay, "Smooth Stabilization Implies Corprime Factorization", *IEEE Trans. Automat. Control*, vol. 34, no. 4, pp. 435-443, 1989.

[18] D.G. Taylor, P. V. Kokotovic, R. Marino and I. Kanellakopoulos, "Adaptive Regulation of Nonlinear Systems With Unmodeled Dynamics", *IEEE Trans. Automat. Control*, vol. 34, no. 4, pp. 405-412, 1988.

[19] K. Nam and A. Arapostatthis, "A Model Reference Adaptive Control Scheme for Pure-Feedback Nonlinear Systems", IEEE Trans. Automat. Control, Vol.33, no. 9 pp. 803-811, 1988.

[20] R.H. Middleton and G. C. Goodwin, "Adaptive Control of Time-varying Linear Systems", *IEEE Trans. Automat. Control*, vol. 33, no. 2, pp. 150-155, 1988.

[21] R.H. Middleton, G. C. Goodwin, D. J. Hill and D. Q. Mayne, " Design Issues in Adaptive Control", *IEEE Trans Automat. Control*, vol.33 no. 1, pp. 50-58, 1989.

[22] M.A. Rotea and P. P. Khargonekar, "Stability of Linear Time-Varying and Uncertain Linear Systems", *IEEE Trans. Automat. Control*, vol. 33, no. 9, pp.884-887, 1988.

[23] G. Kreisselmeier, "Adaptive Control of a Class of Slowly Time-Varying Plants", *Syst. Contr. Lett.*, vol. 8, Dec. 1986.

[24] K.S. Tsakalis and P.A. Ioannou, "Adaptive Control of Linear Time-Varying Plants: A New Model Reference Controller Structure," *IEEE Trans. Auto. Contr.* vol. AC–34, no. 10, pp. 1038–1046, Oct. 1989.

[25] K.S. Tsakalis and P.A. Ioannou, "Adaptive Control of Linear Time-Varying Plants," *Automatica*, vol. 23, pp. 459–468, July 1987.

[26] K. J. Aström and B. Wittenmark, "*Adaptive Control*", Addison-Wesley, 1989.

[27] S. Dubowsky and D.T. DesForges, "The Application of Model Reference Adaptive Control To Robotic Manipulators", ASME *J. Dyn. Syst. Meas. Contr.*, vol. 101, no. 3, pp. 193–200, 1979.

[28] A. Liegois, A. Fournier and M. J. Aldon, "Model Reference Control of High Velocity Industrial Robots," *Proc. 1980 JACC*, San Francisco, Aug. 1980.

[29] H. Seraji, "A New Approach To Adaptive Control of Manipulators," *ASME J. Dyn. Syst. Meas. Contr.*, vol. 109, no. 3, pp. 193–202.

[30] H. Seraji, "Direct Adaptive Control of Manipulators in Cartesian Space," *J. Robotic Syst.*, vol. 4, no. 1, pp. 157–178, 1987.

[31] R.P. Anex, Jr. and M. Hubbard, "Modeling and Adaptive Control of Mechanical Manipulator," ASME J. Dyn. Syst. Meas. Contr., vol. 106, pp. 211–217, Spt. 1984.

[32] A Balestrino, G. De Maria and L. Sciavicco, "An Adaptive Model Reference Following Control For Robot Manipulators," ASME J. Dyn. Syst. Meas. Contr., vol. 105, pp. 143–151, Spt. 1983.

[33] C.S.G. Lee and B.H. Lee, "Resolved Motion Adaptive Control For Mechanical Manipulators," ASME j. Dyn. Syst., Meas. Contr., vol. 106, pp. 134–142, Spt. 1984.

[34] V.A. Malyshev and A.V. Timofeev, "Manipulator Dynamics and Adaptive Control," Automat. Remote Contr., vol. 42, pp. 1069–1076. 1981.

[35] K.Y. Lim and M. Eslami, "Robust Adaptive Controller For Manipulator Systems", *IEEE Trans. Robotics and Automation*, vol. RA-3, no. 1, pp. 54–66, Fed. 1987.

[36] H. Seraji, "Decentralized Adaptive Control of Manipulators: Theory, Simulation and Experimentation", *IEEE Trans. Robotics and Automation*, vol. RA-5, no. 2, pp. 183–201, April 1989.

[37] J.J. Craig, "*Adaptive Control of Mechanical Manipulators*", Addison-Wesley, 1988.

[38] J. Craig, P. Hsu and S. Sastri, "Adaptive control of mechanical manipulators", *Proc. IEEE Conference on Robotics and Automation*, San Francisco, Calif., 1986, 243-248.

[39] J.J. E. Slotine and W. Li, "Composite Adaptive Control of Robot Manipulators", *Automatica*, vol. 25, no. 4, pp.509–519, 1989.

[40] J.E. Slotine and W. Li, "Adaptive manipulator control: a case study", *IEEE Trans. Automat. Control*, vol. 33, no. 11, pp. 995-1003, 1988.

[41] P. Hsu, M. Bodson, S. Sastry and B. Paden, "Adaptive Identification and Control for Manipulators without Using Joint Accelerations," *Proc. IEEE Int. Conf. Robotics and Automation*, pp. 1210–1215, Raleigh, N.C. 1986.

[42] R.H. Middleton and G. C. Goodwin, "Adaptive Computed Torque Control for Rigid Link Manipulators," *Proc. IEEE Conf. Decision and Control*, pp. 68–73, Athens, Dec. 1986.

[43] C. Atkeson, C. An and J. Hollerbach, "Estimation of Inertial Parameters of Manipulator Loads and Links", *Robotics Resaerch*, eds. O. Faugeras and G. Giralt. Cambridge: MIT Press.

[44] P. Khosla and T. Kanade, "Parameter Identification of Robot Dynamics", *Proc. IEEE Conf. Decision and Control*, Fort Lauderdale, Fla., 1985.

[45] A.P. Morgan and K.S. Narendra, "On the Uniform Asymptotic Stability of Certain Linear Nonautonomous Differential Equations," *SIAM J. Contr. Optim.*, 1977, p. 15.

[46] J. J. Uicker, "One the Dynamic Analysis of Spatial Linkages Using 4 × 4 Matrices", Ph. D. dissertation, Evanston, Ill.: Northwestern University, 1965.

[47] M. Kahn and B. Roth, "The Near Minimum Time Control of Open Loop Articulated Chains," *Trans. ASME J. Dynam. Syst. Meas. Contr.* Sept. 1971.

[48] M. Renaud, "Contribution à l'Etude de la Modélisation et de la Commande des Systèms Mécaniques Articulés", Toulouse, France: Université Paul Sabatier, Dec. 1975, Thèse de Docteur Ingénieur.

[49] A. Liegois, W. Khalil, J. M. Dumas and M. Renaud, "Mathematical and Models of Interconnected Mechanical Systems," *Symposium on the Theory and Practice of Robots and Manipulators*, Warsaw, Poland, 1976.

[50] Y. Stepanenko, "A Method of Dynamic Analysis of Spatial Mechanisms," *Mechanics of Machines (USSR)*, (23), 117–124 (1970) (in Russian).

[51] D. E. Orin *et al.* "Kinematic and Kinetic Analysis of Open-Chain Linkage Utilizing Newton-Euler methods," *Math-Biosciences*, 1979; vol. 43: pp. 107–130.

[52] W. W. Armstrong, "Recursive Solution to the Equations of Motion of an N-Link Manipulator," *Proc. 5th World Cong. on the Theory of Machines and Mechanisms*, Montreal, Quebec, Canada, July 1979.

[53] J. M. Hollerbach, "A Recursive Lagrangian Formulation of Manipulator Dynamics and a Comparative Study of Dynamic Formulation Complexity", *IEEE Trans. Syst. Man Cybernet.* Nov. 1980.

[54] D. B. Silver, "On the Equivalence of Lagrangian and Newton-Euler Dynamics for Manipulators," *Int. J. Robotics Res.* 1982; 1 and 2.

[55] J. M. Hollerbach and G. Sahar, "Wrist-Partitioned Inverse Accelerations and Manipulator Dynamics," Cambridge, Mass.: *MIT AI Memo* no. 717, April 1983.

[56] T. K. Kanade, P. K. Khosla and N. Tanaka, "Real-Time Control of the CMU Direct Drive Arm II Using Customized Inverse Dynamics", *Proc. 23rd IEEE Conf. on Decision and Control*, Las Vegas, Nevada, Dec. 1984.

[57] A. Izaguirre and P. P. Paul, "Computation of the Inertial and Gravitational Coefficients of the Dynamic Equation for a Robot Manipulator with a Load,"

Proc. 1985 Inter. Conf. on Robotics and Automation. St. Louis, Mar. 1985: pp. 1024–1032.

[58] B. Armstrong, O. Khatib and J. Burdick, "The Explicit Dynamic Model and Inertial Parameters of the PUMA 560 Arm," *Proc. 1986 IEEE Inter. Conf. on Robotics and Automation,* San Francisco, April 1986: pp. 510–518.

[59] E. Freund, "The Structure of Decoupled Non-Linear Systems," *Int. J. Contr.* 1975: 21(3): pp. 443–450.

[60] Idem, "A Non-Linear Control Concept for Computer-Controlled Manipulators," *Proc. 7th ISIR*, Tokyo, 1977.

[61] J. Zabala Iturralde, "Commande des Robots Manipulateurs à partir de la Modélisation de leur Dynamique," Toulouse, France: Université of Paul Sabatier, July 1978. Thèse de 3^{eme} Cycle.

[62] O. Khatib, M. Llibre and R. Mampey, "Fonction Décision-Commande d'un Robot Manipulateur," *Rapport Scientifique no. 2/7156, DERA-CERT*, Toulouse, France, July 1978.

[63] J.J. Craig, "*Introduction to Robotics Mechanics and Control*", Addison-Wesliey, 1986.

[64] M.W. Spong and M. Vidyasagar, "*Robot Dynamics and Control*", J. Wiley and Son., 1989.

[65] J.B. Griffiths, "*The Theory of Classical Dynamics*" Cambridge University Press, 1985.

[66] M. Vidyasagar, *Nonlinear Systems Analysis*, pp. 189, Prentice-Hall, *Englewood Cliffs*, New Jersey, 1978.

[67] A. Michel and R. Miller, em Qualitative Analysis of Large Scale Dynamical Systems, New York: Academic, 1977.

[68] A. O. Steinhardt, "Householder Transforms in Signal Processing", *IEEE ASSP Magazine*, vol 5, no. 3, pp. 4–12, July 1988.

[69] S. Arimoto and F. Miyazaki, "Stability and Robustness of PID Feedback Control for Robot Manipulators of Sensory Capability", *Third International Symposium of Robotics Research,* Gouvieux, France, July 1985.

[70] C.F. Gerald, and P.O. Wheatley, *Applied Numerical Analysis*, Addison-Wesley Pub. Co., 1989.