Model-Based Adaptive Position and Force Control of Robot Manipulators

by

Qing-Hu Max Meng

BSEE, Luoyang Institute of Technology, China, 1982
MSEE, Beijing Institute of Technology, China, 1988

A DISSERTATION SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
in the
Department of Electrical and Computer Engineering

ACCEPTED
FACULTY OF GRADUATE STUDIES

We accept this dissertation as conforming
to the required standard

DEAN

DATE Sept. 28/92

Dr. W.-S. Lu, Supervisor, Dept. of Elec. & Comp. Engr.

Dr. P. Agathoklis, Department Member, Dept. of Elec. & Comp. Engr.

Dr. J. S. Collins, Department Member, Dept. of Elec. & Comp. Engr.

Dr. Y. Stepanenko, Outside Member, Dept. of Mechanical Engr.

Dr. M. Nahon, Outside Member, Dept. of Mechanical Engr.

Dr. C. C. H. Ma, External Examiner, University of British Columbia

Supervisor: Dr. Wu-Sheng Lu

ABSTRACT

This thesis is primarily concerned with motion control of robot manipulators with emphasis placed on adaptive impedance control and relevant computational issues. The general approach taken in our studies is a model-based approach, that is, the algorithms developed is based on the dynamic model of the robot(s) involved.

Two computational formulations are derived for the evaluation of the so-called regressor dynamics of a robot manipulator, which has played a key role in the development of popular stable adaptive control algorithms for robot manipulators. The closed-form version of the formulations is based on the Lagrangian dynamics formulation while the recursive version is based on the Newton-Euler dynamics.

As an application of the regressor dynamics formulation, the popular Slotine-Li adaptive control algorithm is modified and then implemented on a PUMA 560 robot. Satisfactory computational efficiency of the regressor formulas, especially the recursive formula, has been demonstrated in our experimental implementations.

To extend adaptive position control algorithms to force control, the concept of target impedance reference trajectory is introduced which makes it possible to inject two stable adaptive position control algorithms into Hogan's conventional impedance control. These two adaptive impedance control algorithms have been shown stable. Simulation and real-time implementation of the algorithms on a PUMA 560 robot are reported.

The last part of the thesis conducts a study on optimal load distribution and coordination of multiple robots. Optimal load distribution schemes using a p-norm type optimization approach are proposed. The algorithms are then adopted to dynamically link the two-level controllers in a proposed coordination framework. Simulation results are presented to show the performance of the proposed structure.

Examiners:

Dr. W.-S. Lu, Supervisor, Dept. of Elec. & Comp. Engr.

Dr. P. Agathoklis, Department Member, Dept. of Elec. & Comp. Engr.

Dr. J.S. Collins, Department Member, Dept. of Elec. & Comp. Engr.

Dr. Y. Stepanenko, Outside Member, Dept. of Mechanical Engr.

Dr. M. Nahon, Outside Member, Dept. of Mechanical Engr.

Dr. C. C. H. Ma, External Examiner, University of British Columbia

Contents

List of Figures

List of Tables

Acknowledgement

First and foremost I would like to thank Professor Wu-Sheng Lu, my supervisor, friend, and mentor, for his masterly guidance, constant encouragement, intelligent inspiration, and invaluable advice throughout my graduate education at the University of Victoria. His dedication and long-hour discussions with me are primarily responsible for the successful completion of this work. I thank him deeply.

I would like to thank Dr. P. Agathoklis, Dr. J. Collins, Dr. Y. Stepanenko, and Dr. M. Nahon for proof-reading this thesis and providing expert suggestions. I also thank Dr. C. C. H. Ma of UBC for serving as the external examiner and offering constructive suggestions.

I would also like to thank Mr. Kevin Jones and Mr. Roger Kelley for their expert support in hardware implementation and software development and their wisdom and skill to fix the experimental facilities as my deadlines approached.

I also offer my thanks to my fellow graduate students, faculty and staff members for creating the friendly environment at UVic Engineering where I have spent the last four years. Inspiring talks with many graduate students and faculty members during the past years are acknowledged.

I would also like to acknowledge the financial supports by NSERC and IRIS Center of Excellence program through grants to Professor Wu-Sheng Lu, and by the University of Victoria in the form of teaching assistantships and teaching scholarships and fellowships.

Last but not least, I owe a great deal of thanks to my family for their support during the past years. Especial thanks are due to my wife, Lily. Without her encouragement, patience, and taking care of almost all family matters, this work would be far from complete. I also thank my parents for taking care of my daughter and my parents-in-law for helping us and looking after my son, who arrived just after I finished the first draft of this thesis.

Dedication

To my parents,

for teaching me writing the word "MAN";

To my parents-in-law,

for introducing me to electrical engineering;

To my wife,

for burning her immense love to energize me;

To my daughter and son,

for bringing me lucks and uppermost happiness.

Chapter 1

Introduction

In the past two decades, a great deal of research has been carried out in the field of robotics. Even if confining ourselves only to those dealing with the control problems of robot manipulators, the vast volume of published work makes a concise review difficult if not impossible. Therefore only selected work concerning position control, force control, and coordinated control of rigid robot manipulators and the related topics in robot dynamics will be briefly reviewed.

The fundamental control problems encountered in robotics may be classified as position control, force control, and coordinated control. In what follows we review several basic approaches to these control problems.

1.1 Position Control

Position control is a control task where the robot end-effector (configuration) is required to move from one position (configuration) to another or along a given trajectory without contacting high-stiffness environment. In other words, a position controller is responsible for providing the actuators with proper amount of control torques that would move the end-effector to the positions that are sufficiently close to

the desired ones. Typical criteria for the performance evaluation of a position control algorithm include tracking errors in position and velocity, amplitude and power of the control torques required and algorithm robustness under model uncertainty and system disturbances.

Among the available algorithms, the proportional-derivative (PD) control, the computed torque control, and several stable adaptive position controls appear to be the basic choices for robot position control and they are fundamental in developing other position and force control schemes.

1.1.1 PD Control

The motion of an n degree-of-freedom (DOF) robot can be described by the following dynamics equation

$$\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau} \tag{1.1}$$

where $\mathbf{H}(\mathbf{q})$ is the $n \times n$ symmetric, positive definite mass matrix, $\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}}$ and $\mathbf{g}(\mathbf{q})$ represent torques due to centrifugal and gravity forces respectively, $\boldsymbol{\tau}$ is the $n \times 1$ vector of joint torques supplied by the actuators, and $\mathbf{q}$ is the $n \times 1$ vector of joint displacements.

An independent joint PD controller has the structure shown in Figure 1.1, where a desired position $\mathbf{q}_d$ and a desired velocity $\dot{\mathbf{q}}_d$ are compared to the actual position $\mathbf{q}$ and the actual velocity $\dot{\mathbf{q}}$, respectively, and then the differences are multiplied by a position gain $\mathbf{K}_p$ and a velocity gain $\mathbf{K}_v$ to generate the control torque:

$$\boldsymbol{\tau} = \mathbf{K}_p(\mathbf{q}_d - \mathbf{q}) + \mathbf{K}_v(\dot{\mathbf{q}}_d - \dot{\mathbf{q}}) = \mathbf{K}_p\mathbf{e} + \mathbf{K}_v\dot{\mathbf{e}} \tag{1.2}$$

where $\mathbf{e} = \mathbf{q}_d - \mathbf{q}$ and $\mathbf{K}_p$, $\mathbf{K}_v$ are chosen as positive definite diagonal matrices.

In the *absence* of gravity, that is, if $\mathbf{g}(\mathbf{q})$ is zero in (1.1), the PD control law (1.2) achieves asymptotic tracking of the desired joint position. To see this, consider the

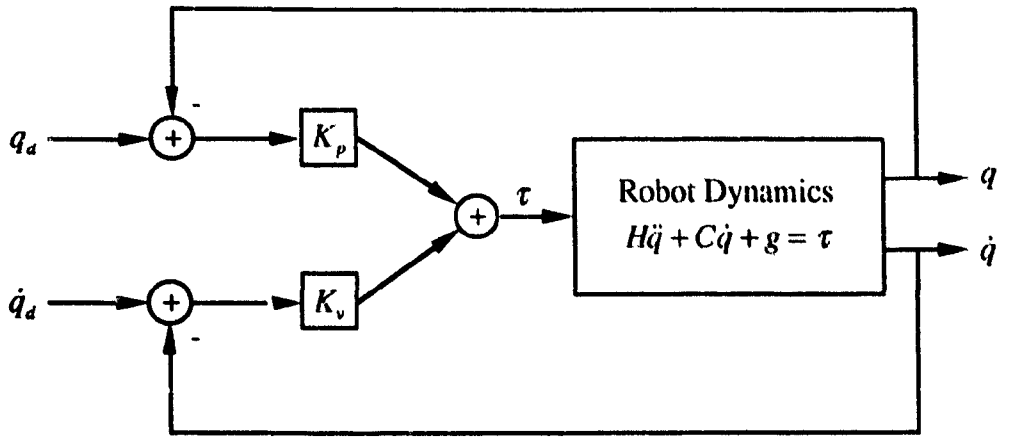


Figure 1.1: Structure of an Independent Joint PD Controller

Lyapunov function

$$\mathcal{V} = \frac{1}{2}\dot{\mathbf{q}}^T\mathbf{H}(\mathbf{q})\dot{\mathbf{q}} + \frac{1}{2}\mathbf{e}^T\mathbf{K}_p\mathbf{e} \tag{1.3}$$

Note that $\mathcal{V}$ represents the total kinetic energy that would result if the joint actuators were to be replaced by springs with stiffness $\mathbf{K}_p$ and with equilibrium position at $\mathbf{q}_d$.

The time derivative of $\mathcal{V}$ is given by

$$\dot{\mathcal{V}} = \dot{\mathbf{q}}^T\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \frac{1}{2}\dot{\mathbf{q}}^T\dot{\mathbf{H}}(\mathbf{q})\dot{\mathbf{q}} - \dot{\mathbf{q}}^T\mathbf{K}_p\mathbf{e} \tag{1.4}$$

Since we assume $\mathbf{g}(\mathbf{q}) = 0$, (1.1) and (1.4) yield

$$\begin{aligned}\dot{\mathcal{V}} &= \dot{\mathbf{q}}^T(\boldsymbol{\tau} - \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}}) + \frac{1}{2}\dot{\mathbf{q}}^T\dot{\mathbf{H}}(\mathbf{q})\dot{\mathbf{q}} - \dot{\mathbf{q}}^T\mathbf{K}_p\mathbf{e} \\ &= \dot{\mathbf{q}}^T(\boldsymbol{\tau} - \mathbf{K}_p\mathbf{e}) + \frac{1}{2}\dot{\mathbf{q}}^T(\dot{\mathbf{H}}(\mathbf{q}) - 2\mathbf{C}(\mathbf{q},\dot{\mathbf{q}}))\dot{\mathbf{q}} \\ &= \dot{\mathbf{q}}^T(\boldsymbol{\tau} - \mathbf{K}_p\mathbf{e})\end{aligned} \tag{1.5}$$

where we have used the fact that $\dot{\mathbf{H}} - 2\mathbf{C}$ is skew symmetric (see Property 3 of Section 2.2.4). Substituting PD control law (1.2) into (1.5) gives

$$\dot{\mathcal{V}} = -\dot{\mathbf{q}}^T\mathbf{K}_v\dot{\mathbf{q}} \leq 0 \tag{1.6}$$

The above analysis shows that $\mathcal{V}$ decreases as long as $\dot{\mathbf{q}}$ is nonzero. Now suppose $\dot{\mathcal{V}} = 0$, (1.6) then implies that $\dot{\mathbf{q}} \equiv 0$ and hence $\ddot{\mathbf{q}} \equiv 0$. From the equations of motion with PD control (1.2)

$$\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} = \mathbf{K}_p\mathbf{e} - \mathbf{K}_v\dot{\mathbf{q}} \tag{1.7}$$

we obtain

$$\mathbf{K}_p\mathbf{e} \equiv 0$$

which implies that $\mathbf{e} \equiv 0$. The well known LaSalle's Theorem then implies that the system is asymptotically stable, i.e. $\mathbf{e} \to 0$ and $\dot{\mathbf{q}} \to 0$ as $t \to \infty$.

In case the gravitational term is not zero but is exactly known, (1.2) can be modified to compensate the gravitational term [1, 2]

$$\boldsymbol{\tau} = \mathbf{K}_p\mathbf{e} + \mathbf{K}_v\dot{\mathbf{e}} + \mathbf{g}(\mathbf{q}) \tag{1.8}$$

and, obviously, the stability of the closed-loop system can be shown in the same way as above. If $\mathbf{g}(\mathbf{q})$ is not known, it can be shown [3, Chapter 10] that control (1.2) can still stabilize the system with a steady-state error $\mathbf{K}_p^{-1}\mathbf{g}(\mathbf{q})$.

1.1.2 Computed Torque Control

The computed torque controller, which was initiated in the work of Paul [4], Markiewicz [5], Bejczy [6], and Lewis [7], has been a popular model-based position controller. Freund is among the first to use the nonlinear dynamic model of a robot manipulator in a control algorithm [8, 9]. Other closely related approaches were given by Luh, Walker, and Paul [10] and Liegois, Fournier, and Aldon [11].

Consider again the dynamics equation (1.1). The idea of computed torque control[1] is to seek a nonlinear feedback control law which globally linearizes and

[1] In some literature computed torque control is also referred to as *inverse dynamics* control.

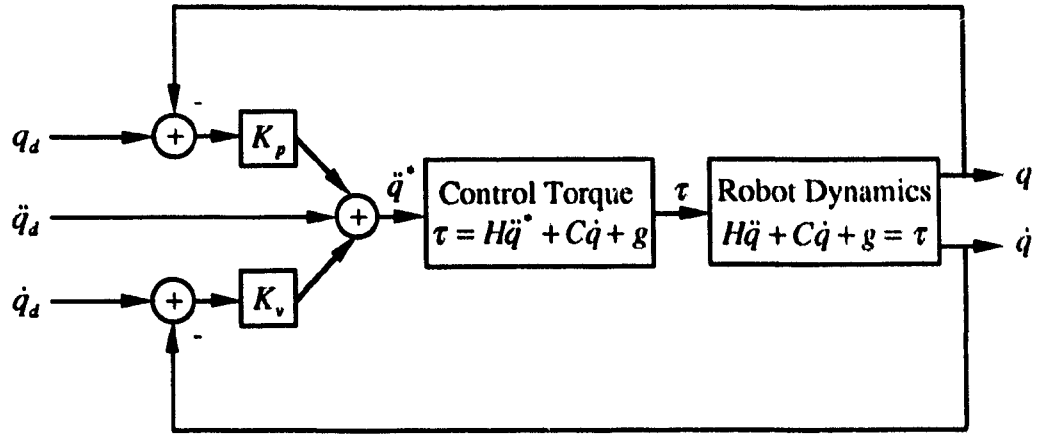


Figure 1.2: Structure of a Computed Torque Controller

decouples the nonlinear robotic system. This is done by choosing $\boldsymbol{\tau}$ as

$$\boldsymbol{\tau} = \mathbf{H}(\mathbf{q})\ddot{\mathbf{q}}^* + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) \tag{1.9}$$

that in conjunction with (1.1) yields

$$\ddot{\mathbf{q}} = \ddot{\mathbf{q}}^* \tag{1.10}$$

where the term $\ddot{\mathbf{q}}^*$ represents a new input to the system.

An obvious choice of $\ddot{\mathbf{q}}^*$ is

$$\ddot{\mathbf{q}}^* = \ddot{\mathbf{q}}_d + \mathbf{K}_v(\dot{\mathbf{q}}_d - \dot{\mathbf{q}}) + \mathbf{K}_p(\mathbf{q}_d - \mathbf{q}) \tag{1.11}$$

where $\ddot{\mathbf{q}}_d$ is the desired joint acceleration, $\mathbf{K}_p$ and $\mathbf{K}_v$ are positive definite diagonal matrices. The closed-loop system is now characterized by the linear error dynamics

$$\ddot{\mathbf{e}}(t) + \mathbf{K}_v\dot{\mathbf{e}}(t) + \mathbf{K}_p\mathbf{e}(t) = \mathbf{0} \tag{1.12}$$

where $\mathbf{e} = \mathbf{q}_d - \mathbf{q}$. An obvious choice for the gain matrices $\mathbf{K}_p$ and $\mathbf{K}_v$ is

$$\begin{aligned} \mathbf{K}_p &= \mathrm{diag}(\omega_1^2, \ldots, \omega_n^2) \\ \mathbf{K}_v &= \mathrm{diag}(2\omega_1, \ldots, 2\omega_n) \end{aligned} \tag{1.13}$$

which results in a decoupled and critically damped second order system with natural frequencies ω_i, $1 \leq i \leq n$.

Concerning the implementation of the computed torque algorithm, it is noted that $\ddot{\mathbf{q}}^*$ is treated as the joint acceleration, then equation (1.9) suggests that the control torque $\boldsymbol{\tau}$ can be computed by, for example, the recursive Newton-Euler formulation.

The structure of a computed torque controller is shown in Figure 1.2.

1.1.3 Adaptive Position Control

Adaptive control schemes are introduced in order to compensate for unknown dynamics. Although non-model-based adaptive controllers, such as adaptive PD control in joint space [12] and in Cartesian space [13], have attempted to deal with dynamics uncertainty, their performance is in general not as satisfactory as model-based adaptive position controllers.

Although adaptive control of robot manipulators has been a subject of study for many years [14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32], it is only recently that adaptive control results have appeared that provide rigorous proofs of global convergence for rigid robots. An excellent review on a number of globally stable adaptive controllers was given by Ortega and Spong [33].

Adaptive control laws may be classified according to their control objectives and the signals that drive the parameter update law. The control objective determines the controller structure whose parameters are to be updated on line. On the other hand, the update law is usually driven by a set of signals that measure either the error between the estimated and true parameters (prediction error) or the error between the actual and desired output (tracking error). Representatives of this class of adaptive laws include Craig et al [34], and Middleton and Goodwin [35]. The control structure adopted in [34, 35] is the computed torque control, and the

update law is driven by tracking error signals in the algorithm of [34].

An alternative approach to globally stable adaptive control of manipulators is to preserve the passivity properties of the robot. Good representatives of this approach are Slotine and Li [36, 37] and Sadegh and Horowitz [38]. The advantages of this approach are

- it avoids using the inverse of the estimated mass matrix $\hat{\mathbf{H}}(\mathbf{q})$;
- it avoids using the joint acceleration $\ddot{\mathbf{q}}$.

It is noted that these quantities need to be evaluated on line to implement Craig's algorithm [34].

It has been shown [39, 40] that the algorithms of Craig and Slotine-Li can be extended to control contact forces of a manipulator in a stable manner. Detailed discussions on the extensions will be given in Chapter 3. In what follows we outline the stable adaptive algorithms by Craig [34] and Slotine-Li [37], respectively.

Craig's Adaptive Algorithm

Consider the robot dynamics (1.1), i.e.

$$\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau} \tag{1.14}$$

To control the manipulator, the following control law is proposed by Craig et al [34, 41]

$$\boldsymbol{\tau} = \hat{\mathbf{H}}(\mathbf{q})\ddot{\mathbf{q}}^* + \hat{\mathbf{C}}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \hat{\mathbf{g}}(\mathbf{q}) \tag{1.15}$$

where $\hat{\mathbf{H}}(\mathbf{q})$, $\hat{\mathbf{C}}(\mathbf{q},\dot{\mathbf{q}})$, and $\hat{\mathbf{g}}(\mathbf{q})$ are $\mathbf{H}(\mathbf{q})$, $\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})$, and $\mathbf{g}(\mathbf{q})$ that are evaluated using estimated parameters, and

$$\ddot{\mathbf{q}}^* = \ddot{\mathbf{q}}_d + \mathbf{K}_v\dot{\mathbf{e}} + \mathbf{K}_p\mathbf{e} \tag{1.16}$$

where $\mathbf{e} = \mathbf{q}_d - \mathbf{q}$, and $\mathbf{K}_p > 0$ and $\mathbf{K}_v > 0$ are constant, diagonal gain matrices with diagonal elements $\{k_{pj}\}$ and $\{k_{vj}\}$, respectively.

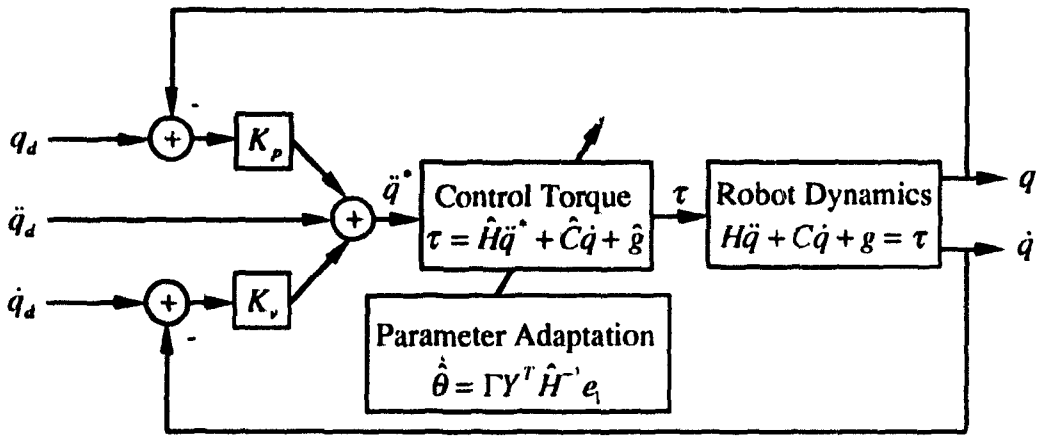


Figure 1.3: Structure of Craig's Adaptive Controller

The block diagram shown in Figure 1.3 indicates the structure of the adaptive controller that makes use of a dynamic model of the manipulator. An adaptive element is also indicated. This adaptive element observes tracking errors and adjusts the parameters that appear in the control law (1.15).

By equating (1.14) and (1.15) we obtain

$$\ddot{\mathbf{e}} + \mathbf{K}_v\dot{\mathbf{e}} + \mathbf{K}_p\mathbf{e} = \hat{\mathbf{H}}^{-1}(\mathbf{q})[\tilde{\mathbf{H}}(\mathbf{q})\ddot{\mathbf{q}} + \tilde{\mathbf{C}}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \tilde{\mathbf{g}}(\mathbf{q})] \tag{1.17}$$

where $\tilde{(\cdot)} = (\cdot) - \hat{(\cdot)}$.

Using property 2 of the robot dynamics (see Section 2.2.4), the error equation (1.17) can be written in the form

$$\ddot{\mathbf{e}} + \mathbf{K}_v\dot{\mathbf{e}} + \mathbf{K}_p\mathbf{e} = \hat{\mathbf{H}}^{-1}(\mathbf{q})\mathbf{Y}(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}})\tilde{\boldsymbol{\theta}} \tag{1.18}$$

where $\mathbf{Y}(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}})$, called the manipulator regressor, is an $n \times r$ matrix of known functions, $\tilde{\boldsymbol{\theta}} = \boldsymbol{\theta} - \hat{\boldsymbol{\theta}}$ is an $r \times 1$ vector representing parameter with

$$\boldsymbol{\theta} = [\theta_1 \quad \theta_2 \quad \dots \quad \theta_n]^T$$

the true parameters and

$$\hat{\boldsymbol{\theta}} = [\hat{\theta}_1 \quad \hat{\theta}_2 \quad \dots \quad \hat{\theta}_n]^T$$

an estimate of $\boldsymbol{\theta}$.

The adaptive law estimates the unknown parameters by using filtered servo error signals. The filtered servo error for the jth joint is

$$e_{1j}(s) = (s + \psi_j)e_j(s) \tag{1.19}$$

where the ψ_j is a positive constant. Hence,

$$\mathbf{e}_1 = \dot{\mathbf{e}} + \boldsymbol{\Psi}\mathbf{e} \tag{1.20}$$

where $\boldsymbol{\Psi}$ =diag($\psi_1 \quad \psi_2 \quad \dots \quad \psi_n$). Note that for manipulators equipped with position and velocity sensors, $\mathbf{e}_1$ can be computed simply from sensor measurements instead of using (1.20).

Constant ψ_j is chosen such that the transfer function

$$\frac{s + \psi_j}{s^2 + k_{vj}s + k_{pj}} \tag{1.21}$$

is strictly positive real [41] (see Appendix B). By the lossless version of the positive real lemma [42] there exist positive definite matrices $\mathcal{P}_j$ and $\mathcal{Q}_j$ such that

$$\begin{aligned} \mathbf{A}_j^T\mathcal{P}_j + \mathcal{P}_j\mathbf{A}_j &= -\mathcal{Q}_j \\ \mathcal{P}_j\mathbf{B}_j &= \mathbf{C}_j^T \end{aligned} \tag{1.22}$$

where matrices $\mathbf{A}_j$, $\mathbf{B}_j$, and $\mathbf{C}_j$ are the matrices of a minimal state space realization of the filtered error equation of the jth joint, i.e.

$$\begin{aligned} \dot{\mathbf{x}}_j &= \mathbf{A}_j\mathbf{x}_j + \mathbf{B}_j(\hat{\mathbf{H}}^{-1}\mathbf{Y}\tilde{\boldsymbol{\theta}})_j \\ e_{1j} &= \mathbf{C}_j\mathbf{x}_j \end{aligned} \tag{1.23}$$

with state vector $\mathbf{x}_j = [\mathbf{e}_j \quad \dot{\mathbf{e}}_j]^T$.

The filtered error equation of the entire system in state space form is now given by

$$\begin{aligned} \dot{\mathbf{X}} &= \mathbf{A}\mathbf{X} + \mathbf{B}\hat{\mathbf{H}}^{-1}\mathbf{Y}\tilde{\boldsymbol{\theta}} \\ \mathbf{e}_1 &= \mathbf{C}\mathbf{X} \end{aligned} \tag{1.24}$$

where $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ are all block diagonal (with $\mathbf{A}_j$, $\mathbf{B}_j$, and $\mathbf{C}_j$ on the diagonals, respectively) and $\mathbf{X} = [\mathbf{x}_1 \quad \mathbf{x}_2 \quad \dots \quad \mathbf{x}_n]^T$. Forming the $2n \times 2n$ matrices $\mathcal{P} = \text{diag}(\mathcal{P}_1 \quad \mathcal{P}_2 \quad \dots \quad \mathcal{P}_n)$ and $\mathcal{Q} = \text{diag}(\mathcal{Q}_1 \quad \mathcal{Q}_2 \quad \dots \quad \mathcal{Q}_n)$, we have $\mathcal{P} > 0$, $\mathcal{Q} > 0$, and

$$\begin{aligned} \mathbf{A}^T\mathcal{P} + \mathcal{P}\mathbf{A} &= -\mathcal{Q} \\ \mathcal{P}\mathbf{B} &= \mathbf{C}^T \end{aligned} \tag{1.25}$$

Lyapunov theory can now be adopted to derive an adaptation law. The Lyapunov function is defined by

$$\mathcal{V}(\mathbf{X}, \tilde{\boldsymbol{\theta}}) = \mathbf{X}^T\mathcal{P}\mathbf{X} + \tilde{\boldsymbol{\theta}}^T\boldsymbol{\Gamma}^{-1}\tilde{\boldsymbol{\theta}} \tag{1.26}$$

where $\boldsymbol{\Gamma} = \text{diag}(\gamma_1 \quad \gamma_2 \quad \dots \quad \gamma_n)$ and $\gamma_i > 0$ is nonnegative in both servo and parameter errors. Differentiation of $\mathcal{V}$ with respect to time leads to

$$\dot{\mathcal{V}}(\mathbf{X}, \tilde{\boldsymbol{\theta}}) = -\mathbf{X}^T\mathcal{Q}\mathbf{X} + 2\tilde{\boldsymbol{\theta}}^T(\mathbf{Y}^T\hat{\mathbf{H}}^{-1}\mathbf{e}_1 + \boldsymbol{\Gamma}^{-1}\dot{\tilde{\boldsymbol{\theta}}}) \tag{1.27}$$

If we choose

$$\dot{\tilde{\boldsymbol{\theta}}} = -\boldsymbol{\Gamma}\mathbf{Y}^T\hat{\mathbf{H}}^{-1}\mathbf{e}_1 \tag{1.28}$$

then

$$\dot{\mathcal{V}}(\mathbf{X}, \tilde{\boldsymbol{\theta}}) = -\mathbf{X}^T\mathcal{Q}\mathbf{X} \tag{1.29}$$

which is nonpositive because $\mathcal{Q}$ is positive definite. Since $\tilde{\boldsymbol{\theta}} = \boldsymbol{\theta} - \hat{\boldsymbol{\theta}}$, we have

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

and from (1.28) we have the adaptation law

$$\dot{\hat{\boldsymbol{\theta}}} = \boldsymbol{\Gamma}\mathbf{Y}^T\hat{\mathbf{H}}^{-1}\mathbf{e}_1 \tag{1.30}$$

Equations (1.26) and (1.29) imply that $\mathbf{X}$ and $\tilde{\boldsymbol{\theta}}$ are bounded. Control (1.15) and adaptation law (1.30) define a stable adaptive control algorithm.

Slotine-Li's Adaptive Algorithm

Consider the robot dynamics (1.1), i.e.

$$\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau} \tag{1.31}$$

Define

$$\dot{\mathbf{q}}_r = \dot{\mathbf{q}}_d - \boldsymbol{\Lambda}\tilde{\mathbf{q}} \tag{1.32}$$

where $\tilde{\mathbf{q}} = \mathbf{q} - \mathbf{q}_d$ denotes the position tracking error, and $\boldsymbol{\Lambda}$ is a positive constant matrix.

From Property 2 (see Section 2.2.4)

$$\tilde{\mathbf{H}}(\mathbf{q})\ddot{\mathbf{q}}_r + \tilde{\mathbf{C}}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}}_r + \tilde{\mathbf{g}} = \mathbf{Y}(\mathbf{q},\dot{\mathbf{q}},\dot{\mathbf{q}}_r,\ddot{\mathbf{q}}_r)\breve{\boldsymbol{\theta}} \tag{1.33}$$

The control and adaptation laws in the Slotine-Li algorithm [37] is given by

$$\boldsymbol{\tau} = \hat{\mathbf{H}}(\mathbf{q})\ddot{\mathbf{q}}_r + \hat{\mathbf{C}}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}}_r + \hat{\mathbf{g}}(\mathbf{q}) - \mathbf{K}\mathbf{s} \tag{1.34}$$

$$\dot{\hat{\boldsymbol{\theta}}} = -\boldsymbol{\Gamma}\mathbf{Y}^T(\mathbf{q},\dot{\mathbf{q}},\dot{\mathbf{q}}_r,\ddot{\mathbf{q}}_r)\mathbf{s} \tag{1.35}$$

where $\boldsymbol{\Gamma}$ is a constant positive definite matrix, $\mathbf{K}$ is a constant positive definite matrix, and $\mathbf{s}$, which can be considered as a measure of tracking accuracy, is defined by

$$\mathbf{s} = \dot{\mathbf{q}} - \dot{\mathbf{q}}_r = \dot{\tilde{\mathbf{q}}} + \boldsymbol{\Lambda}\tilde{\mathbf{q}} \tag{1.36}$$

The above control and adaptation laws guarantee the global convergence of the position and velocity tracking errors, as long as the desired trajectories $\mathbf{q}_d$, $\dot{\mathbf{q}}_d$, and $\ddot{\mathbf{q}}_d$ are bounded. To show this, consider the Lyapunov function

$$\mathcal{V} = \frac{1}{2}[\mathbf{s}^T\mathbf{H}\mathbf{s} + \tilde{\boldsymbol{\theta}}^T\boldsymbol{\Gamma}^{-1}\tilde{\boldsymbol{\theta}}] \tag{1.37}$$

and the differentiation of $\mathcal{V}(t)$ with respect to time leads to

$$\begin{aligned}\dot{\mathcal{V}} &= \mathbf{s}^T(\mathbf{H}\ddot{\mathbf{q}} - \mathbf{H}\ddot{\mathbf{q}}_r) + \tilde{\boldsymbol{\theta}}^T\boldsymbol{\Gamma}^{-1}\dot{\tilde{\boldsymbol{\theta}}} + \frac{1}{2}\mathbf{s}^T\dot{\mathbf{H}}\mathbf{s} \\ &= \mathbf{s}^T(\boldsymbol{\tau} - \mathbf{C}\dot{\mathbf{q}}_r - \mathbf{g} - \mathbf{H}\ddot{\mathbf{q}}_r) + \tilde{\boldsymbol{\theta}}^T\boldsymbol{\Gamma}^{-1}\dot{\tilde{\boldsymbol{\theta}}}\end{aligned}$$

where the fact that $\dot{\mathbf{H}} - 2\mathbf{C}$ is skew symmetric (Property 3 of Section 2.2.4) has been used to eliminate the term $\frac{1}{2}\mathbf{s}^T\dot{\mathbf{H}}\mathbf{s}$. Substituting the control law (1.34) into the above expression, and using the linearity property of robot dynamics (Property 2 of Section 2.2.4), we obtain

$$\dot{\mathcal{V}} = -\mathbf{s}^T\mathbf{K}\mathbf{s} + \tilde{\boldsymbol{\theta}}[\boldsymbol{\Gamma}^{-1}\dot{\tilde{\boldsymbol{\theta}}} + \mathbf{Y}^T\mathbf{s}] \tag{1.38}$$

The adaptation law (1.35) is chosen in order to remove the second term from the last expression. Indeed, substituting (1.35) into (1.38) leads to

$$\dot{\mathcal{V}} = -\mathbf{s}^T\mathbf{K}\mathbf{s} \leq 0 \tag{1.39}$$

Since $\mathcal{V}(t)$ is lower bounded by zero and decreases for any nonzero $\mathbf{s}$, as seen from (1.39), then $\mathcal{V}(t)$ and therefore (from (1.37)) the tracking error measure $\mathbf{s}$, must converge to zero. It can then be shown, from definition (1.36), that the convergence of $\mathbf{s}$ to zero in turn guarantees that $\tilde{\mathbf{q}}$ and $\dot{\tilde{\mathbf{q}}}$ also converge to zero. Therefore, both global stability of the system, and the convergence of the tracking error, are guaranteed by the above adaptive control algorithm.

As is noted in [37], if one chooses $\boldsymbol{\Lambda} = \lambda\mathbf{I}$ in (1.32) and $\mathbf{K} = \lambda\hat{\mathbf{H}}(\mathbf{q})$ in (1.34), then the time derivative of $\mathcal{V}$ is given by

$$\dot{\mathcal{V}} = -\lambda\mathbf{s}^T\mathbf{H}(\mathbf{q})\mathbf{s} \tag{1.40}$$

provided that the unknown parameters are updated according to

$$\dot{\hat{\boldsymbol{\theta}}}_u = -\boldsymbol{\Gamma}\mathbf{Y}^T(\mathbf{q}, \dot{\mathbf{q}}, \dot{\mathbf{q}}_r, (\ddot{\mathbf{q}}_r - \lambda\mathbf{s}))\mathbf{s} \tag{1.41}$$

Since $\mathbf{H(q)}$ is uniformly positive definite, $\dot{V}$ is negative if λ is sufficiently large. Note that the control torque in this case becomes

$$\tau = \hat{\mathbf{H}}(\mathbf{q})\ddot{\mathbf{q}}^* + \hat{\mathbf{C}}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}}_r + \hat{\mathbf{g}}(\mathbf{q}) \tag{1.42}$$

where

$$\ddot{\mathbf{q}}^* = \ddot{\mathbf{q}}_d - 2\lambda\dot{\tilde{\mathbf{q}}} - \lambda^2\tilde{\mathbf{q}} \tag{1.43}$$

Obviously, (1.42) and (1.43) represent a quasi-computed-torque controller.

Figure 1.4 shows the structure of Slotine-Li's adaptive controller.

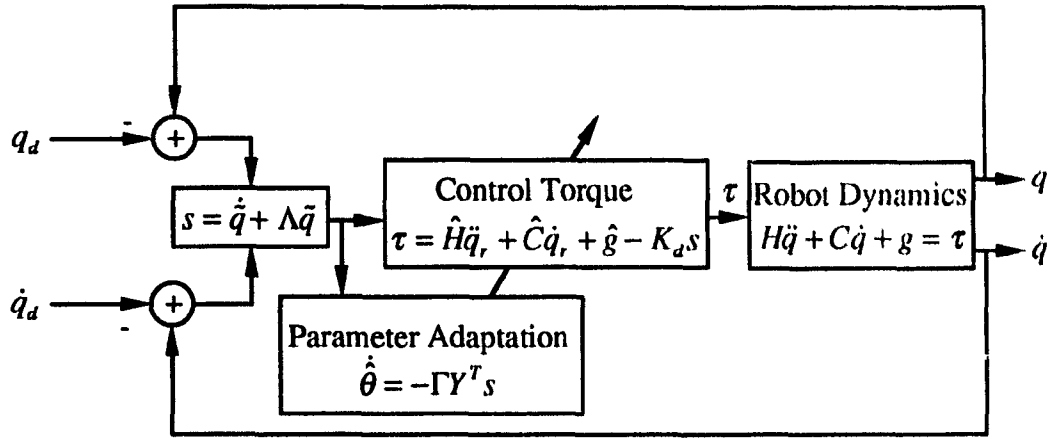


Figure 1.4: Structure of Slotine-Li's Adaptive Controller

1.2 Force Control

Many tasks we wish to accomplish using manipulators require not only controlling the position of a manipulator but also the force exerted by its end-effector on the object. Assembly, polishing, deburring, opening and closing a door, wiping a window, and turning a crank are examples of such tasks.

Force control[2] can be considered as a general form of motion control where

[2]The term *force control* used in this thesis is referred to as simultaneous control of both force and position.

the manipulator is required to contact the environment as it tracks a trajectory. Instead of only position variables to plan and control, there are now additionally force variables to plan and control. Since it deals with position and contact force at the tip of a robot manipulator, force control strategies are generally described in Cartesian space. As compared to position control, force control is a much more complex issue, and there have been relatively fewer theoretical results and successful implementations.

Since early 1970's several approaches to stable execution of contact tasks using robot manipulators have been proposed. These include the work by Groome [43], Whitney [44], Bolles and Paul [45], Nevins and Whitney [46], Drake [47], and Paul and Shimano [48]. An excellent survey of the important work in robot force control during 1967 to 1985 can be found in [49]. Tutorial [50] provides a good collection of research work accomplished during 1972-1984 including eight papers on force sensing and control. Chapter 5 of the book [51] gives an overview on several force-control methods as well as a collection of seven papers in this area. Part IV of a recent book [52] reviews progress and problems on motion and force control. It includes a survey on stable force-control methods and a survey on PD controller design using the concept of potential field. An in-depth force control study with stability analysis can be found in the monograph [53].

Based on our literature review and understanding of the development in robot force control, it is felt that the following list includes major approaches to robot force control:

- Impedance control
- Hybrid position/force control
- Stiffness control
- Compliance control

- Resolved acceleration based control
- Potential field based control
- Open-loop joint torque control

In what follows, we briefly review these approaches. In addition, work on adaptive force control will also be reviewed.

1.2.1 Impedance Control

Hogan and his colleagues [54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66] initiated the development of the impedance control. Non-model-based impedance control can be formed by combining a stiffness controller with a damping controller [67, Chapter 11]. On the other hand, model-based impedance control regulates the dynamic relation between the robot end effector and the contact environment [62, 63].

Lawrence [68] studied the stability properties of impedance control in common implementations. Anderson and Spong [69] proposed a hybrid impedance control scheme which combines hybrid position/force control with impedance control. Recent work in adaptive impedance control has been reported by Kelly, Carelli, Amestegul, and Ortega [70, 71] and by Lu and Meng [72, 39] respectively. Meng and Lu also extended their adaptive impedance control algorithm for redundant robot case [73]. Lu and Goldenberg [74] investigated the implementation of robust impedance control and Abdallah, Jordan, and Horne [75] reported their design of positive-real impedance controller.

It appears that impedance control is one of the successful algorithms for stable execution of contact tasks as it provides a unified approach to both free motion control and constrained motion control (force control) in a stable manner.

1.2.2 Hybrid Position/Force Control

The hybrid position/force control method has been developed based on the observation that a force control task can be partitioned into two sub-tasks in two subspaces - a position subspace and a force subspace. The sub-task in the position subspace requires pure position control while the sub-task in the force subspace requires only force control. This partition is dynamic and the two subspaces may change during the task, but once partitioned, the sub-task in one subspace requires only *one* type of control so that the algorithm development in each subspace is simplified.

The hybrid position/force control was first proposed by Raibert and Craig [76] and it was based on the theoretical framework of Mason [77]. Khatib formulated a hybrid control scheme as an unified approach for motion and force control in [78]. Zhang and Paul proposed their hybrid control scheme in [79]. West and Asada developed a hybrid controller which was implemented on a commercially available robot [80, 81]. Yoshikawa introduced dynamics into hybrid control and studied dynamic hybrid control [82, 83]. In a recent work [84], Mills linked constrained motion to hybrid position/force control by using constrained dynamics formulation that describes the dynamic behavior of the manipulator during force controlled tasks to design a hybrid controller. Wedel and Saridis implemented a hybrid controller on a PUMA 600 industrial robot to study the position and force tracking abilities of the algorithm [85]. In addition, several adaptive and robust hybrid controllers were proposed by Slotine and his co-workers [86, 87, 88], Seraji [89], Mo and Bayoumi [90], Chung and Leininger [91], Kuo and Wang [92], De Luca, Manes and Ulivi [93], etc. Paul summarized the problems and research topics associated with hybrid control in a tutorial [94]. Duffy [95] pointed out that the hybrid control theory, which is based on "orthogonal complements" of twist and wrench spaces, is "fallacious".

1.2.3 Stiffness Control

Stiffness is the rate at which the forces (and torques) on the end effector increase as it is deflected from its nominal position. Stiffness control was originated in the work of Salibury [96], in which the gain of the position control loops is adjusted to control the apparent stiffness of a manipulator. Stiffness control is a special case of impedance control without the damping control part. And impedance control can be considered as an enhanced version of stiffness control where robot dynamic model may be used.

1.2.4 Compliance Control

Compliance is the inverse of stiffness. To allow the end effector to move relatively to a fixed wrist, the compliance controller acts like a spring where the compliance is the inverse of the spring constant. There are both passive and active compliance control algorithms. Early work on active and passive compliance robot control was reported by Whitney and Nevins [44, 46, 97], Drake [47], Paul and Shimano [48], Mason [77], Hanafusa and Asada [98, 99]. The subject was further studied by El Maraghy and Johns [100], DeSchutter and Van Brussel [101], Kazerooni and his co-workers [102, 103, 104, 105, 106], and Shin and Lee [107].

1.2.5 Resolved Acceleration Based Force Control

Motivated by the resolved acceleration control for robot free motion by Luh, Walker, and Paul [10], the concept was used to combine with either stiffness or compliance control to handle force control problems [108, 107]. Similar results were also reported by McClamroch and Wang [109, 110], in which a linear feedback controller was used to control the position and compliance of a nonlinear constrained mechanism. Impedance control can be considered as a resolved-acceleration control combined

with an enhanced version of stiffness plus damping control algorithms. Therefore stiffness control, impedance control, and resolved-acceleration control share the same underlying mechanism.

1.2.6 Potential-Field-Based Force Control

Potential field is a concept originally proposed by Khatib et al [111, 112], Hogan [61], and Miyazaki and Arimoto [2] to deal with obstacle avoidance problem. It appears to be a *concept* rather than an *algorithm*. It shares the underlying computational algorithms with PD control, stiffness control, and impedance control as shown by Hogan [59, 60] and Neculasau [113].

1.2.7 Open-Loop Joint Torque Control

Open-loop joint torque control was first proposed by Wu and Paul [114] and Luh, Fisher, and Paul [115]. Experimental studies of this control approach on a direct-drive robot were described by An, Atkeson, and Hollerbach in [53].

With force control relying on force feedback from a wrist force sensor, a high force feedback gain and a stiff environment can cause instability in the force feedback loop. On the other hand, stability can be improved by lowering the force feedback gain with deteriorated force resolution. The open-loop joint torque control approach achieves force control by relying on the measurements of joint torques and the ability to command them accurately. The wrist force feedback gain is zero with this method, which is an open-loop control from the point of view of interaction forces at the end effector. On the other hand, a closed-loop torque servo is implemented at each joint so that the joint torque can be regulated accurately.

Wu and Paul [114] presented a good comparison between wrist force sensing and joint torque sensing in implementing force control. Wrist sensing provides accurate force/torque measurements at the hand, but because the robot structure is

inherently a low bandwidth flexible system and the sensor is situated at the end of this structure, a high gain feedback will produce instability. Therefore, only a slow closed-loop system can be implemented stably using a wrist force sensor. On the other hand, since joint torque sensors are situated before the low bandwidth robot structure, a high bandwidth torque inner loop can be implemented around each joint. Since the sensors are not installed at the tip, the tip forces and torques cannot be inferred as accurately as by wrist sensing.

Since both wrist sensing and joint sensing have positive and negative features, proper combination of these two sensing systems may provide a stable high bandwidth force control system. As was found in [53], the high bandwidth open-loop joint torque control with an inner torque servo loop will provide stability and fast response, and the lower bandwidth outer loop with wrist force sensing will provide steady state accuracy.

1.2.8 Adaptive Force Control

Slotine and Li [86] reformulated their adaptive algorithms for robot free motion control [37] to deal with end-effector dynamics which results in an adaptive hybrid position/force control. The computational aspects of this force control strategy were later discussed by Niemeyer and Slotine [88]. Other types of adaptive hybrid controllers were proposed by Mo and Bayoumi [90] and Chung and Leininger [91]. Experimental implementation results were described in [91]. Kelly, Carelli, Amestegui, and Ortega [70, 71] proposed an adaptive force control algorithm based on impedance control. Liu, Han, Lingarkar, and El Bestawi [116] combined the force control method of [109] and the adaptive law of [117] to form an adaptive force control. Lu and Meng proposed two adaptive impedance control algorithms in [39] and reported the implementation results of these algorithms as applied to a PUMA 560 manipulator in [118, 119].

1.3 Coordinated Control

Objects having large volume or heavy weight can be handled using coordinating robots more efficiently as opposed to using a single-arm robot. In many assembly tasks, using a multi-arm robotic system might be the only choice to achieve the desired efficient automation. Here the terminology *coordinated control* is referred to the motion coordination of a robotic system where the number of participating robots is greater than or equal to two.

The major issues in controlling multi-arm systems, which do not exist in controlling single-arm robots, include motion coordination and load distribution among the participating robots. Obviously, the resolution of the former requires an appropriate model of the multi-arm system with a non-point-mass payload included, a mechanism to deal with internal forces among the manipulators, and possibly a mechanism to deal with relative motion between the object and the end-effectors. And the resolution of the latter requires the formulation of an objective function whose minimization will lead to an optimal distribution of the instant load over the participating robots. In what follows, several approaches to these issues are reviewed.

1.3.1 Master-Slave Approach

The master-slave approach was initiated by Ishia [120], and Alford and Belyeu [121]. In this approach, only two coordinating robots are considered, where one is called the master robot and the other is the slave robot. The master robot is controlled to move along a specified trajectory while the slave robot is controlled to move along a trajectory which is determined by the *actual* trajectory of the master robot.

1.3.2 Parallel Control Approach

The master-slave approach does not consider constrained motions. Zheng and Luh proposed an approach [122, 123], which is generally referred to as the parallel control approach. In this approach, dynamic constraint relations are established as the basis for the coordinated control. Typical Cartesian space position and force control strategies such as PD control, computed torque control, hybrid control and impedance control for a single robot can then be extended to the coordinated control.

1.3.3 Object Control Approach

Recently, a new approach called object control approach emerged which are based on the work by Mason [77] and Tarn, Bejczy, and Yun [124]. The control strategy in this approach is specified at the object level in Cartesian space and is then passed on to the coordinating robots in the system through certain mechanism such as a load distribution algorithm which dynamically relates the object motion to the motion of each participating robot.

Recent work using object control approach was reported by Alberts and Soloway [125], Nakamura, Nagai, and Yoshikawa [126], Hsu [127], Wen [128], Schneider and Cannon [129], and Hu and Goldenberg [130].

1.3.4 Adaptive Coordinated Control

Several researchers have proposed adaptive techniques to deal with multi-arm systems with unknown system parameters. Pittelkau presented an adaptive load sharing force control scheme for two coordinating manipulators [131]. Walker, Kim, and Dionise developed an adaptive control scheme for two manipulator arms using an object control approach [132]. Hu and Goldenberg derived an adaptive algorithm using Popov's hyperstability theory [130].

1.3.5 Optimal Load Distribution

Load distribution has been one of the important issues in the analysis of multi-robot systems [133]. In fact it plays a key role in dynamically relating the manipulated object to the coordinating robots in a multirobot system. For systems with two coordinating robots handling a single object, the problem has been investigated by Orin and Oh [134], Zheng and Luh [135, 136], and Pittelkau [131]. Force distribution problem in multiple-chain robotic systems is studied by Nakamura, Nagai, and Yoshikawa [137], Kumar and Waldron [138], and Cheng and Orin [139]. Lu and Meng proposed p-norm type optimal load distribution algorithms in [140, 141].

In practice, the magnitude of each joint force for a mechanical manipulator has its upper bound. Consequently, the load distribution problem may be described as a *constrained* optimization problem. Methods used to solve the load distribution problem include linear programming [134, 139], nonlinear programming [135, 136], p-norm type optimization [140, 141] and adaptive approach [131].

1.4 Problems to be Studied

This thesis is primarily concerned with motion control of robot manipulators with emphasis placed on adaptive impedance control and relevant computational issues. The general approach taken in our studies is a *model-based* approach, that is, the algorithms to be developed will be based on the dynamic model of the robot(s) involved. The rationale behind this approach is that in general control signals generated by a good model-based algorithm are able to steer the robot to achieve the desired motion with satisfactory accuracy. Although implementing model-based control algorithms often require intensive computation, the needed computing power is now available with fairly moderate cost. Above all, as soon as the model-based approach is adopted, numerically efficient implementation of the algorithms obtained

has been a source of motivation that has eventually led to a detailed study on the so-called regressor dynamics formulation. Within the framework of robot motion control, specific problems studied in this thesis can be described as follows.

1. We shall study the impedance control of single-arm robotic systems where some of system parameters are not known. Adaptive mechanisms shall be injected into the conventional impedance control initiated by Hogan in order to regulate contact forces in a stable manner and possibly identify the values of the unknown parameters.

2. Most of stable adaptive algorithms for motion and force control of robots require evaluating the manipulator regressor on line. It is realized that this evaluation is computationally equivalent to the evaluation of manipulator dynamics, but the conventional dynamics formulations such as recursive Newton-Euler or Lagrangian formulations are not suitable for many adaptive algorithms. The problem of efficient computation of manipulator regressor will be studied in detail.

3. To verify feasibility of the algorithms developed in real-time motion/force control, we shall carry out experimental studies by applying the algorithms to a PUMA 560 manipulator, where robot dynamics as well as parameter adaptation law will be implemented using the recursive regressor dynamics formulation.

4. We shall present a preliminary study on motion coordination of a multi-arm system, which includes a model set-up for an arm-payload system and a proposal for a coordination framework. In addition, we shall study the problem of optimal load sharing for multi-arm systems using some optimization techniques.

1.5 Contributions of this Dissertation

The contributions of this thesis can be summarized as follows.

1. Two adaptive impedance control algorithms are developed and their stability in contact-force regulation are proved. More specifically, the concept of target impedance reference trajectory (TIRT) is introduced and used to extend the stable adaptive algorithms proposed by Craig et al [34] and Slotine-Li [36] to constrained motion control.

2. Two approaches for the evaluation of manipulator regressor are presented. The first approach is based on a reformulation of the Lagrangian formulation of robot dynamics by vector analysis techniques and the result is a closed-form solution for the manipulator regressor. A key property used in the solution derivation is that link Lagrangians are additive. The second approach is based on a reformulation of the Newton-Euler formulation of robot dynamics and offers a recursive scheme for the regressor dynamics. The derivation is essentially a step-by-step re-organization of the Newton-Euler formulation to extract system parameters using vector analysis techniques.

3. The proposed adaptive impedance control algorithms are implemented on a PUMA 560 by using the recursive algorithm of the regressor dynamics. Comparison studies are conducted to show the importance of utilizing robot dynamics in controlling a manipulator. It will be demonstrated that adaptive position and force control algorithms using regressor dynamics can be efficiently implemented on general open-chain robot manipulators by using recursive regressor dynamics.

4. A preliminary study on the coordination of a multiple robot system is presented. By formulating the system load as a p-norm objective function, op-

timal load distribution algorithms are proposed and used as the core of a coordination framework. Simulation studies are carried out to show the performance of the proposed scheme.

1.6 Organization of this Dissertation

Chapter 2 begins with a brief review of the conventional Newton-Euler dynamics and Lagrangian dynamics and their properties. Then the regressor dynamics of a robot manipulator is introduced and two approaches for its computation are developed. Examples are included to illustrate the algorithms. Issues on real-time implementation of the regressor dynamics in an adaptive position control algorithm are also addressed.

In Chapter 3 we develop two adaptive impedance control algorithms. These algorithms are developed by introducing adaptive update laws for parameter estimation into the conventional impedance control via the concept of target impedance reference trajectory. Algorithm stability is shown for each of the proposed impedance controls. A comparative study is carried out for the proposed algorithms and the nonadaptive impedance control. Simulation and experimental results are included to illustrate the proposed algorithms.

In Chapter 4 several problems in coordinated control of robot manipulators are addressed. We start with a description of the problems associated with coordination such as the dynamics of multirobot systems, dynamic load distribution, internal force control. Two p-norm type optimization algorithms are developed for optimal load distribution, and they are compared to the existing load distribution schemes. Finally we propose a coordination framework which uses the optimal load distribution algorithm as a core to dynamically link the two-level controllers in the framework. A simulation study is included to illustrate the proposed scheme.

Chapter 5 summarizes the work presented and discusses possible future research projects in position, force, and coordinated control and their applications in industries and space exploration.

Chapter 2

Regressor Dynamics of Robot Manipulators

2.1 Introduction

The dynamic behavior of a robot manipulator can be described in terms of the arm configuration variations in relation to the torques exerted by its actuators. From a control point of view, the problem that we would like to investigate is to compute the required joint torques[1] for a given trajectory $\mathbf{q}(t)$, $\dot{\mathbf{q}}(t)$ and $\ddot{\mathbf{q}}(t)$. Two most commonly used descriptions for robot dynamics are the Lagrangian formulation and the Newton-Euler formulation. Excellent treatment of these formulations can be found in several textbooks, e.g. [3, 142]. Although these formulations have played a very important role in many non-adaptive control algorithms such as computed torque and feedforward control [3], they are unlikely the best choice when a model-based adaptive motion control algorithm needs to be implemented [33]. As a matter of fact, most of the established model-based adaptive control algorithms require the

[1] Unless indicated otherwise in this thesis, we use the word *torque* to mean a vector of forces and torques in joint space and *force* to mean a vector of forces and torques in Cartesian space.

use of the regressor dynamics for algorithm derivation as well as implementation [33, 143].

This chapter intends to provide a detailed discussion on the regressor dynamics formulation which is found suitable for implementing many established stable adaptive control algorithms. As the proposed regressor dynamics has two versions that can be viewed as the counterparts of the conventional Lagrangian and Newton-Euler formulations respectively, for the sake of comparison we include a concise review of these conventional dynamics.

2.2 Conventional Dynamics Formulations

In this section we give a brief review of the Lagrangian and the Newton-Euler dynamics formulations. Detailed treatment of these conventional equations of motion can be found in, for example, [3, 142].

2.2.1 Newton-Euler Dynamics Formulation

Newton's Equation

Consider each link of a robot manipulator as a rigid body and assume that the center of mass of a rigid body is accelerating with acceleration $\dot{v}_c$, then the force exerted at the center of mass which causes this acceleration is given by

$$F = m\dot{v}_c \tag{2.1}$$

where m is the total mass of the body.

Euler's Equation

Assume that a rigid body is rotating with angular velocity ω and with angular acceleration $\dot{\omega}$ then the moment N, exerted on the body is given by

$$\mathbf{N} = {}^{c}\mathbf{I}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times {}^{c}\mathbf{I}\boldsymbol{\omega}, \tag{2.2}$$

where ${}^{c}I$ is the inertia tensor of the body written in a frame $\{C\}$ whose origin is located at the center of mass.

Recursive Newton-Euler Dynamics Formulation

The Newton-Euler approach to this problem has been a subject for study for many years [144, 145, 146, 147, 148]. Consider the problem of computing the torques required to move the robot along a given trajectory $(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$. With this knowledge, and the knowledge of the kinematics and mass distribution information of the robot, we can calculate the joint torques required to cause this motion. The algorithm described here is based upon the method published by Luh, Walker, and Paul in [147].

The algorithm is composed of two parts. First, link velocities and accelerations are recursively computed from link 1 to link n. Second, equation (2.1) and (2.2) are applied to each link to obtain forces and torques of interaction and joint actuator torques recursively from link n back to link 1. The algorithm summarized below is for the case of rotational joints.

The Recursive Newton-Euler Dynamics Agorithm

Outward recursions: i : 0 → n − 1

$$ {}^{i+1}\boldsymbol{\omega}_{i+1} = {}^{i+1}_{i}\mathbf{R}\, {}^{i}\boldsymbol{\omega}_{i} + \dot{q}_{i+1}\mathbf{e}_z \tag{2.3}$$

$$ {}^{i+1}\dot{\boldsymbol{\omega}}_{i+1} = {}^{i+1}_{i}\mathbf{R}\, {}^{i}\dot{\boldsymbol{\omega}}_{i} + {}^{i+1}_{i}\mathbf{R}\, {}^{i}\boldsymbol{\omega}_{i} \times \dot{q}_{i+1}\mathbf{e}_z + \ddot{q}_{i+1}\mathbf{e}_z \tag{2.4}$$

$$ {}^{i+1}\dot{\mathbf{v}}_{i+1} = {}^{i+1}_{i}\mathbf{R}({}^{i}\dot{\boldsymbol{\omega}}_{i} \times {}^{i}\mathbf{P}_{i+1} + {}^{i}\boldsymbol{\omega}_{i} \times ({}^{i}\boldsymbol{\omega}_{i} \times {}^{i}\mathbf{P}_{i+1}) + {}^{i}\dot{\mathbf{v}}_{i}) \tag{2.5}$$

$$
\begin{aligned}
{}^{i+1}\dot{\mathbf{v}}_{C_{i+1}} &= {}^{i+1}\dot{\boldsymbol{\omega}}_{i+1} \times {}^{i+1}\mathbf{P}_{C_{i+1}} \\
&\quad + {}^{i+1}\boldsymbol{\omega}_{i+1} \times ({}^{i+1}\boldsymbol{\omega}_{i+1} \times {}^{i+1}\mathbf{P}_{C_{i+1}}) + {}^{i+1}\dot{\mathbf{v}}_{i+1} \qquad (2.6)\\
{}^{i+1}\mathbf{F}_{i+1} &= m_{i+1}\, {}^{i+1}\dot{\mathbf{v}}_{C_{i+1}} \qquad (2.7)\\
{}^{i+1}\mathbf{N}_{i+1} &= {}^{C_{i+1}}\mathbf{I}_{i+1}\, {}^{i+1}\dot{\boldsymbol{\omega}}_{i+1} + {}^{i+1}\boldsymbol{\omega}_{i+1} \times {}^{C_{i+1}}\mathbf{I}_{i+1}\, {}^{i+1}\boldsymbol{\omega}_{i+1} \qquad (2.8)
\end{aligned}
$$

Inward recursions: i : n → 1

$$
\begin{aligned}
{}^{i}\mathbf{f}_i &= {}^{i}_{i+1}\mathbf{R}\, {}^{i+1}\mathbf{f}_{i+1} + {}^{i}\mathbf{F}_i \qquad (2.9)\\
{}^{i}\mathbf{n}_i &= {}^{i}\mathbf{N}_i + {}^{i}_{i+1}\mathbf{R}\, {}^{i+1}\mathbf{n}_{i+1} + {}^{i}\mathbf{P}_{C_i} \times {}^{i}\mathbf{F}_i \\
&\quad + {}^{i}\mathbf{P}_{i+1} \times {}^{i}_{i+1}\mathbf{R}\, {}^{i+1}\mathbf{f}_{i+1} \qquad (2.10)\\
\tau_i &= \mathbf{e}_z^T\, {}^{i}\mathbf{n}_i \qquad (2.11)
\end{aligned}
$$

In the above equations, ${}^{i}\boldsymbol{\omega}_i$ is the angular velocity vector of the origin of frame $\{i\}$ with reference to the universe frame expressed in frame $\{i\}$; ${}^{i}\mathbf{v}_i$ is the linear velocity vector of the origin of frame $\{i\}$ with reference to the universe frame expressed in the link frame $\{i\}$; ${}^{i}\mathbf{v}_{C_i}$ is the linear velocity vector of the the origin of the frame $\{C_i\}$, which is attached at the center of mass of link $\{i\}$ and whose orientation is the same as that of the link frame $\{i\}$, with reference to the universe frame expressed in frame $\{i\}$; ${}^{i}\mathbf{F}_i$ and ${}^{i}\mathbf{N}_i$ are the inertial force and torque vectors exerted at the center of mass of link $\{i\}$; ${}^{i}\mathbf{f}_i$ and ${}^{i}\mathbf{n}_i$ are the force and torque vectors exerted on link i by link $i-1$; τ_i is the torque of joint i; ${}^{i}_{i+1}\mathbf{R} = {}^{i+1}_{i}\mathbf{R}^T$ is the rotation matrix describing the rotation of frame $\{i+1\}$ relative to $\{i\}$; ${}^{i}\mathbf{P}_{i+1}$ is the position vector of the frame $\{i+1\}$ in frame $\{i\}$; ${}^{i}\mathbf{P}_{C_i}$ is the position vector of center of mass of link $\{i\}$ in frame $\{i\}$; m_i is the mass of the link i; ${}^{C_i}\mathbf{I}_i$ is the inertia tensor of link i written in the frame $\{C_i\}$; q_i is i-th joint angular displacement; and $\mathbf{e}_z = [0 \;\; 0 \;\; 1]^T$.

The above equations are valid for a robot whose joints are rotational. If joint $i+1$ is prismatic, equations (2.3),(2.4), (2.5), and (2.11) need to be modified as

$$
{}^{i+1}\boldsymbol{\omega}_{i+1} = {}^{i+1}_{i}\mathbf{R}\, {}^{i}\boldsymbol{\omega}_i \qquad (2.3a)
$$

and

$$ {}^{i+1}\dot{\boldsymbol{\omega}}_{i+1} = {}^{i+1}_{i}\mathbf{R}\, {}^{i}\dot{\boldsymbol{\omega}}_{i} \tag{2.4a} $$

$$ \begin{aligned} {}^{i+1}\dot{\mathbf{v}}_{i+1} = {}^{i+1}_{i}\mathbf{R}({}^{i}\dot{\boldsymbol{\omega}}_{i} \times {}^{i}\mathbf{P}_{i+1} + {}^{i}\boldsymbol{\omega}_{i} \times ({}^{i}\boldsymbol{\omega}_{i} \times {}^{i}\mathbf{P}_{i+1}) + {}^{i}\dot{\mathbf{v}}_{i}) \\ + 2\,{}^{i+1}\boldsymbol{\omega}_{i+1} \times \dot{d}_{i+1}\mathbf{e}_z + \ddot{d}_{i+1}\mathbf{e}_z \end{aligned} \tag{2.5a} $$

$$ \tau_i = \mathbf{e}_z^T\, {}^{i}\mathbf{f}_i \tag{2.11a} $$

2.2.2 Lagrangian Dynamics Formulation

The Newton-Euler approach is based on the elementary dynamic formulations (2.1) and (2.2), and on the analysis of forces and moments of constraint acting between the links. As an alternative, the Lagrangian formulation is an "energy based" approach. The kinetic energy of the i-th link, $k^{(i)}$, can be expressed as

$$ k^{(i)} = \frac{1}{2} m_i \mathbf{v}_{C_i}^T \mathbf{v}_{C_i} + \frac{1}{2}\, {}^{i}\boldsymbol{\omega}_i^T\, {}^{C_i}\mathbf{I}_i\, {}^{i}\boldsymbol{\omega}_i \tag{2.12} $$

where the first term is kinetic energy due to linear velocity of the mass center of the link, and the second term is kinetic energy due to angular velocity of the link. The total kinetic energy of the manipulator is the sum of the kinetic energy in the individual links, that is,

$$ \mathcal{K} = \sum_{i=1}^{n} k^{(i)} \tag{2.13} $$

Since $\mathbf{v}_{C_i}$ and ${}^{i}\boldsymbol{\omega}_i$ in equation (2.12) are functions of $\mathbf{q}$ and $\dot{\mathbf{q}}$, we see that the kinetic energy of a manipulator can be described as a quadratic form $\mathcal{K}(\mathbf{q}, \dot{\mathbf{q}})$. In fact, the kinetic energy of a manipulator is given by

$$ \mathcal{K}(\mathbf{q}, \dot{\mathbf{q}}) = \frac{1}{2}\dot{\mathbf{q}}^T \mathbf{H}(\mathbf{q})\dot{\mathbf{q}} \tag{2.14} $$

where $\mathbf{H}(\mathbf{q})$ is the $n \times n$ manipulator mass matrix. Since the total kinetic energy is always positive, the manipulator mass matrix is a *positive definite* matrix. Note that equation (2.14) is analogous to the familiar expression for the kinetic energy of

a point mass,

$$k = \frac{1}{2}mv^2$$

and the fact that a manipulator mass matrix must be positive definite is analogous to the fact that a point mass is always a positive number.

The potential energy of the ith link, $u^{(i)}$, can be expressed as

$$u^{(i)} = -m_i\, {}^0\mathbf{g}_i^T\, {}^0\mathbf{P}_{C_i} + u_{ref_i} \tag{2.15}$$

where ${}^0\mathbf{g}_i$ is the 3×1 gravity vector of ith link expressed in the base frame $\{0\}$, ${}^0\mathbf{P}_{C_i}$ is the vector locating the center of mass of the ith link in the base frame $\{0\}$, and u_{ref_i} is a constant chosen so that the minimum value of $u^{(i)}$ is zero.[2] The total potential energy stored in the manipulator is the sum of the potential energy in the individual links, that is,

$$\mathcal{U} = \sum_{i=1}^{n} u^{(i)} \tag{2.16}$$

Since the ${}^0\mathbf{P}_{C_i}$ in equation (2.15) are functions of $\mathbf{q}$, we see that the potential energy of a manipulator can be described as a function of joint position, $\mathcal{U}(\mathbf{q})$.

Defining the Lagrangian as

$$\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}}) = \mathcal{K}(\mathbf{q}, \dot{\mathbf{q}}) - \mathcal{U}(\mathbf{q}) \tag{2.17}$$

the equations of motion are given by

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\mathbf{q}}}\right) - \frac{\partial \mathcal{L}}{\partial \mathbf{q}} = \boldsymbol{\tau} \tag{2.18}$$

or

$$\frac{d}{dt}\left(\frac{\partial \mathcal{K}}{\partial \dot{\mathbf{q}}}\right) - \frac{\partial \mathcal{K}}{\partial \mathbf{q}} + \frac{\partial \mathcal{U}}{\partial \mathbf{q}} = \boldsymbol{\tau} \tag{2.19}$$

where $\boldsymbol{\tau}$ is the $n \times 1$ vector of actuator torques.

[2] Actually, since only the partial derivative of the potential energy with respect to $\mathbf{q}$ will appear in the dynamics, this constant is arbitrary. This corresponds to defining the potential energy relative to an arbitrary zero reference height.

2.2.3 The Structure of the Robot Dynamics Equations

When the Newton-Euler or Lagrangian dynamics equations are evaluated symbolically for any manipulator, the dynamic equation can be written in the form

$$\boldsymbol{\tau} = \mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) \tag{2.20}$$

where $\mathbf{H}(\mathbf{q})$ is the $n \times n$ mass matrix of the manipulator, $\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}}$ is an $n \times 1$ vector of centrifugal and Coriolis terms, and $\mathbf{g}(\mathbf{q})$ is an $n \times 1$ vector of gravity terms.

In general, entries of $\mathbf{H}(\mathbf{q})$, $\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})$, and $\mathbf{g}(\mathbf{q})$ are quite complicated functions of $\mathbf{q}$ and $\dot{\mathbf{q}}$ [149].

2.2.4 Some Properties of the Robot Dynamics Equation

Although the dynamics equations of a robot manipulator (2.20) are coupled, nonlinear equations in general, they have several properties which are found useful in facilitating analysis and design of control system. In what follows we state these properties without proofs. Derivations and proofs of these properties can be found in, for example, [142, 150].

Property 1 The mass matrix $\mathbf{H}(\mathbf{q})$ in equation (2.20) is symmetric, positive definite, and both $\mathbf{H}(\mathbf{q})$ and $\mathbf{H}(\mathbf{q})^{-1}$ are uniformly bounded for $\mathbf{q} \in \mathbf{R}^n$.

Property 2 All robot parameters such as link masses, moments of inertias, etc., appear as coefficients of known functions of $(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}})$. By defining each coefficient as a separate parameter, we can write (2.20) as

$$\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \mathbf{Y}(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}})\boldsymbol{\theta} = \boldsymbol{\tau} \tag{2.21}$$

where $\mathbf{Y}(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}})$ is an $n \times r$ matrix of known functions which has been known as the *manipulator regressor*, and $\boldsymbol{\theta}$ is an r dimensional vector of parameters.

Property 3 Matrix

$$\mathbf{S}(\mathbf{q},\dot{\mathbf{q}}) \;=\; \dot{\mathbf{H}}(\mathbf{q}) - 2\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})$$

is skew symmetric, i.e., it satisfies

$$\mathbf{S}(\mathbf{q},\dot{\mathbf{q}}) \;=\; -\mathbf{S}(\mathbf{q},\dot{\mathbf{q}})^T$$

or equivalently,

$$\mathbf{x}^T\mathbf{S}\mathbf{x} \;=\; \mathbf{x}^T(\dot{\mathbf{H}} - 2\mathbf{C})\mathbf{x} \;=\; 0.$$

2.3 Regressor Dynamics Formulation

In this section, we formulate the robot dynamics by using the regressor dynamics formulation. As was mentioned earlier, this formulation is useful in derivation and implementation of many adaptive control algorithms [151].

2.3.1 The Regressor Dynamics

The manipulator regressor, defined by (2.21) has been a key quantity in derivation as well as implementation of the many established adaptive position and force control algorithms [33]. This is because the availability of the regressor enables one to express the dynamics of a robot manipulator as

$$\mathbf{Y}(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}})\boldsymbol{\theta} \;=\; \boldsymbol{\tau} \tag{2.22}$$

with $\boldsymbol{\theta} \in \mathbf{R}^r$ representing the manipulator parameters, thus a Lyapunov approach may lead to a *linear* law for updating the parameters. Studies on this linear parameter-dependence issue from an identification point of view can be found in [150, 152, 36] among others.

In principle the regressor can be obtained using a two-step approach. The first step is to formulate the manipulator dynamics as

$$\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} \;+\; \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} \;+\; \mathbf{g}(\mathbf{q}) \;=\; \boldsymbol{\tau} \tag{2.23}$$

This can be accomplished by for example using the Newton-Euler or Lagrange formulation, see section 2.2 for the details of these formulations and their computation complexity. Having done this, the second step of the approach defines a parameter vector $\boldsymbol{\theta}$ and then works on each entry on the left-hand side of (2.23) to extract vector $\boldsymbol{\theta}$, leading to the regressor formulation $\mathbf{Y}\boldsymbol{\theta} = \boldsymbol{\tau}$. So we see that computationally this is an indirect approach that requires formulating (2.23) plus a parameter extraction procedure. As the entries of $\boldsymbol{\theta}$ are in general spread over all the entries of $\mathbf{H}(\mathbf{q})$, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$, and $\mathbf{g}(\mathbf{q})$, the second step is also computationally complicated.

In this section we propose two methods that compute the regressor of a general n degrees of freedom (DOF) robot *without* using (2.23). Our first method provides a closed-form solution [153], which is obtained by extracting parameter $\boldsymbol{\theta}$ from link Lagrangians *during* the Lagrangian formulation; our second method gives a recursive type solution [154], which is obtained by extracting parameter $\boldsymbol{\theta}$ from joint velocities, accelerations, forces, and torques *during* the Newton-Euler formulation. As opposed to the conventional two-step approach, in which one derives (2.23) with the entries of $\boldsymbol{\theta}$ spread widely over the terms and then extracts these parameters term by term, the proposed methods perform the parameter extraction and dynamics formulation simultaneously and, therefore, more efficiently. Furthermore, unlike the two-step approach which defines parameter vector $\boldsymbol{\theta}$ by trial-and-error until $\boldsymbol{\theta}$ can be extracted from every term on the left-hand side of (2.23), the formation of $\boldsymbol{\theta}$ in the proposed methods has an explicit rule to follow. It is found that if for a specified link the mass, mass center, and inertia tensor are the parameters to be extracted, then this portion of vector $\boldsymbol{\theta}$ has dimension 16 although the number of physical parameters that are related to the link is 10. In addition, the so-called filtered regressor adopted in several adaptive motion control algorithms [35, 155, 156] can readily be obtained from the proposed formula.

2.3.2 Preliminaries

$\mathbf{Y}_u(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$ – The Regressor Associated with Unknown Parameters

It follows from Property 2 that equations of motion (2.23) can be written as

$$\mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})\boldsymbol{\theta} = \boldsymbol{\tau} \tag{2.24}$$

where $\mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) \in \mathbf{R}^{n \times r}$ is the manipulator regressor and $\boldsymbol{\theta} \in \mathbf{R}^{r \times 1}$ is the vector formed by the dynamic parameters of the manipulator in a certain manner. Denoting

$$\boldsymbol{\theta} = \begin{bmatrix} \boldsymbol{\theta}_k \\ \boldsymbol{\theta}_u \end{bmatrix} \tag{2.25}$$

with $\boldsymbol{\theta}_k \in \mathbf{R}^{r_1}$ and $\boldsymbol{\theta}_u \in \mathbf{R}^{r_2}$ representing the known and unknown parameters, respectively, and

$$\mathbf{Y} = [\mathbf{Y}_k \quad \mathbf{Y}_u] \tag{2.26}$$

with $\mathbf{Y}_k \in \mathbf{R}^{n \times r_1}$ and $\mathbf{Y}_u \in \mathbf{R}^{n \times r_2}$, equations (2.23) and (2.24) imply that

$$\mathbf{H}\ddot{\mathbf{q}} + \mathbf{C}\dot{\mathbf{q}} + \mathbf{g} = \mathbf{Y}_k\boldsymbol{\theta}_k + \mathbf{Y}_u\boldsymbol{\theta}_u \tag{2.27}$$

If $\hat{\boldsymbol{\theta}}_u$ is an estimate of $\boldsymbol{\theta}_u$, then

$$\hat{\mathbf{H}}\ddot{\mathbf{q}} + \hat{\mathbf{C}}\dot{\mathbf{q}} + \hat{\mathbf{g}} = \mathbf{Y}_k\boldsymbol{\theta}_k + \mathbf{Y}_u\hat{\boldsymbol{\theta}}_u \tag{2.28}$$

where $\hat{\mathbf{H}}, \hat{\mathbf{C}}, \hat{\mathbf{g}}$ assume the same forms as $\mathbf{H}$, $\mathbf{C}$ and $\mathbf{g}$ respectively, with $\boldsymbol{\theta}_u$ replaced by $\hat{\boldsymbol{\theta}}_u$. It follows that

$$\tilde{\mathbf{H}}\ddot{\mathbf{q}} + \tilde{\mathbf{C}}\dot{\mathbf{q}} + \tilde{\mathbf{g}} = \mathbf{Y}_u\tilde{\boldsymbol{\theta}}_u \tag{2.29}$$

where $(\tilde{*}) = (\hat{*}) - (*)$. It is equation (2.29) that plays a role in the establishment of the many stable algorithms for adaptive control of robots.

The Relation of $\mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$ to the Lagrangians of Manipulator Links and Load

Consider an n DOF manipulator grasping firmly a non-point-mass load. Denote by $k^{(i)}$ and $u^{(i)}$ $(1 \leq i \leq n)$ the kinetic and potential energy of link i , and by $k^{(n+1)}$ and $u^{(n+1)}$ the kinetic and potential energy of the load, respectively. If the load is treated as link $n+1$ and the link Lagrangian of link i is defined as (see Section 2.2.2)

$$l^{(i)} = k^{(i)} - u^{(i)} \qquad 1 \leq i \leq n+1 \tag{2.30}$$

then the manipulator Lagrangian is

$$\mathcal{L} = \sum_{i=1}^{n+1} l^{(i)} \tag{2.31}$$

It follows that the manipulator Lagrangian is *linkwise additive*, that is, if a new link is added to the robot, its manipulator Lagrangian is then equal to the original $\mathcal{L}$ plus the Lagrangian of the new link. This property turns out to be a key fact in the subsequent derivation of the regressor as it allows one to separate the parameters of a specific link from the parameters of other links. From (2.24),(2.31), and the Lagrange's equations of motion

$$\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{\mathbf{q}}}\right) - \frac{\partial \mathcal{L}}{\partial \mathbf{q}} = \boldsymbol{\tau} \tag{2.32}$$

we obtain

$$\sum_{i=1}^{n+1}\left[\frac{d}{dt}\left(\frac{\partial l^{(i)}}{\partial \dot{\mathbf{q}}}\right) - \frac{\partial l^{(i)}}{\partial \mathbf{q}}\right] = \mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})\boldsymbol{\theta} \tag{2.33}$$

Now if $\boldsymbol{\theta}$ is partitioned as

$$\boldsymbol{\theta} = \begin{bmatrix} \boldsymbol{\theta}^{(1)} \\ \boldsymbol{\theta}^{(2)} \\ \vdots \\ \boldsymbol{\theta}^{(n+1)} \end{bmatrix} \tag{2.34}$$

with $\boldsymbol{\theta}^{(i)}$ representing the dynamic parameters of link i, i.e. its mass, mass center and inertia tensor and if $\mathbf{Y}$ is partitioned into $n+1$ blocks

$$\mathbf{Y} = [\mathbf{Y}^{(1)} \quad \mathbf{Y}^{(2)} \quad \dots \quad \mathbf{Y}^{(n+1)}] \tag{2.35}$$

with dimensions consistent to (2.34), then

$$\frac{d}{dt}(\frac{\partial l^{(i)}}{\partial \dot{\mathbf{q}}}) - \frac{\partial l^{(i)}}{\partial \mathbf{q}} = \mathbf{Y}^{(i)}\boldsymbol{\theta}^{(i)} \quad 1 \le i \le n+1 \tag{2.36}$$

In section 2.3.3, (2.36) will be used to obtain a formula for $\mathbf{Y}^{(i)}$.

Once $\mathbf{Y}$ is computed, $\mathbf{Y}_k$ and $\mathbf{Y}_u$ in (2.26) can readily be found as follows. Vector $\boldsymbol{\theta}$ in (2.34) can be regrouped as in (2.25), where $\boldsymbol{\theta}_u$ is formed by correcting the entries, each of which involves at least one of the unknown parameters, and $\boldsymbol{\theta}_k$ is simply the complement of $\boldsymbol{\theta}_u$ in $\boldsymbol{\theta}$. Obviously, this regrouping can be done by premultiplying $\boldsymbol{\theta}$ by an elementary transformation matrix $\mathbf{T}$, i.e.

$$\mathbf{T}\boldsymbol{\theta} = \begin{bmatrix} \boldsymbol{\theta}_k \\ \boldsymbol{\theta}_u \end{bmatrix}_{(r_1+r_2)\times 1} \tag{2.37}$$

From

$$\mathbf{Y}\boldsymbol{\theta} = \mathbf{Y}\mathbf{T}^T\mathbf{T}\boldsymbol{\theta} = \mathbf{Y}\mathbf{T}^T \begin{bmatrix} \boldsymbol{\theta}_k \\ \boldsymbol{\theta}_u \end{bmatrix} = [\mathbf{Y}_k \quad \mathbf{Y}_u] \begin{bmatrix} \boldsymbol{\theta}_k \\ \boldsymbol{\theta}_u \end{bmatrix} \tag{2.38}$$

it follows that

$$\mathbf{Y}_k(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) = \text{the first } r_1 \text{ columns of } \mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})\mathbf{T}^T \tag{2.39}$$

and

$$\mathbf{Y}_u(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) = \text{the last } r_2 \text{ columns of } \mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})\mathbf{T}^T \tag{2.40}$$

Useful Identities from Vector Analysis

The identities

$$(\mathbf{a} \times \mathbf{b})^T\mathbf{c} = (\mathbf{b} \times \mathbf{c})^T\mathbf{a} = (\mathbf{c} \times \mathbf{a})^T\mathbf{b} \tag{2.41}$$

$$(\mathbf{a} \times \mathbf{b}) \times \mathbf{c} = (\mathbf{c}^T\mathbf{a})\mathbf{b} - (\mathbf{b}^T\mathbf{c})\mathbf{a} \tag{2.42}$$

with $\mathbf{a}, \mathbf{b}, \mathbf{c} \in \mathbf{R}^{3\times 1}$ are well known in vector analysis. In Appendix A, we consider several types of expressions that will be used in the subsequent sections and show how identities (2.41) and (2.42) can be applied to extract various unknown parameters from the expressions.

2.3.3 Closed-Form Formulation of Regressor Dynamics

Denote by $\{i\}$ the frame attached to link i according to the Denavit-Hartenberg convention [3, Section 3.4], and by $\{C_i\}$ the frame attached at the mass center of the link, which differs from $\{i\}$ only by a translation. Denote by m_i, ${}^i\mathbf{P}_{C_i} = [p_x\ p_y\ p_z]^T$, and ${}^c\mathbf{I}_i$ the mass, mass center, and inertia tensor of the link, respectively, where ${}^i\mathbf{P}_{C_i}$ is expressed in $\{i\}$, and ${}^c\mathbf{I}_i$ is expressed with respect to $\{C_i\}$ and is explicitly given by (A.9). Further denote by ${}^0_i\mathbf{R}$ the rotation matrix describing frame $\{i\}$ in the base frame $\{0\}$ and by $\mathbf{J}^{(i)} \in \mathbf{R}^{6\times n}$ the i-th link *Jacobian* [3, Chapter 5] expressed in frame $\{0\}$, which will often be partitioned into $\mathbf{J}_v^{(i)} \in R^{3\times n}$ – the Jacobian associated with the translational velocity of frame $\{i\}$ and $\mathbf{J}_\omega^{(i)} \in R^{3\times n}$ – the Jacobian associated with the rotational velocity of frame $\{i\}$, i.e.

$$\mathbf{J}^{(i)} = \begin{bmatrix} \mathbf{J}_v^{(i)} \\ \mathbf{J}_\omega^{(i)} \end{bmatrix}$$

For the sake of simplicity, in what follows the superscript i in the Jacobian will be omitted. In the sequel, $\mathbf{J}_v$ and $\mathbf{J}_\omega$ will sometimes be used in the form

$$\mathbf{J}_v = \mathbf{J}_v^{(i)} = [\mathbf{J}_{v1}\ \ \mathbf{J}_{v2}\ \ \ldots\ \ \mathbf{J}_{vn}], \qquad \mathbf{J}_\omega = \mathbf{J}_\omega^{(i)} = [\mathbf{J}_{\omega 1}\ \ \mathbf{J}_{\omega 2}\ \ \ldots\ \ \mathbf{J}_{\omega n}]$$

where $\mathbf{J}_{vj}$ and $\mathbf{J}_{\omega j}$ are the j-th column of $\mathbf{J}_v$ and $\mathbf{J}_\omega$, respectively.

The Formula

By (2.30), the left-hand side of (2.36) is equal to

$$\frac{d}{dt}\left(\frac{\partial k^{(i)}}{\partial \dot{\mathbf{q}}}\right) - \frac{\partial k^{(i)}}{\partial \mathbf{q}} + \frac{\partial u^{(i)}}{\partial \mathbf{q}} \tag{2.43}$$

where the kinetic energy of the link i is given by

$$k^{(i)} = \frac{1}{2} m \| {}^0\mathbf{v}_{C_i} \|^2 + \frac{1}{2} {}^i\boldsymbol{\omega}_i^T \, {}^c\mathbf{I}_i \, {}^i\boldsymbol{\omega}_i \tag{2.44}$$

with

$${}^0\mathbf{v}_{C_i} = \mathbf{J}_v \dot{\mathbf{q}} + \mathbf{J}_\omega \dot{\mathbf{q}} \times {}^0_i\mathbf{R} \, {}^i\mathbf{P}_{C_i} \tag{2.45}$$

$${}^i\boldsymbol{\omega}_i = {}^i_0\mathbf{R}\mathbf{J}_\omega(\mathbf{q})\dot{\mathbf{q}} \tag{2.46}$$

and the potential energy of the link is given by

$$u^{(i)} = -m \, {}^0\mathbf{P}_{C_i}^T \, {}^0\mathbf{g}_i \tag{2.47}$$

with ${}^0\mathbf{g}_i$ the gravity vector and ${}^0\mathbf{P}_{C_i}$ the position vector from the origin of frame $\{0\}$ to the link's mass center. From (2.44) (2.46) it follows that

$$\frac{\partial k^{(i)}}{\partial \dot{\mathbf{q}}} = m \frac{\partial \, {}^0\mathbf{v}_{C_i}^T}{\partial \dot{\mathbf{q}}} \, {}^0\mathbf{v}_{C_i} + \mathbf{J}_\omega^T \, {}^0_i\mathbf{R} \, {}^c\mathbf{I}_i \, {}^i_o\mathbf{R}\mathbf{J}_\omega \dot{\mathbf{q}} \tag{2.48}$$

With ${}^0\mathbf{v}_{C_i}$ given by (2.45), formula (A.31) can be used to express the first term in (2.48) as $\mathbf{W}_1(\mathbf{q}, \dot{\mathbf{q}})\boldsymbol{\theta}_5$ with $\mathbf{W}_1(\mathbf{q}, \dot{\mathbf{q}})$ defined by (A.32) and $\boldsymbol{\theta}_5$ defined by (A.30). Further let

$$\mathbf{d} = {}^i_0\mathbf{R}\mathbf{J}_\omega \dot{\mathbf{q}} \equiv [d_1 \;\; d_2 \;\; d_3]^T \tag{2.49}$$

formula (A.11) gives

$$\mathbf{J}_\omega^T \, {}^0_i\mathbf{R} \, {}^c\mathbf{I}_{C_i} \, {}^i_0\mathbf{R}\mathbf{J}_\omega^T \dot{\mathbf{q}} = \mathbf{J}_\omega^T \, {}^0_i\mathbf{R}\mathbf{B}(\mathbf{d})\boldsymbol{\theta}_3 \equiv \mathbf{W}_2(\mathbf{q}, \dot{\mathbf{q}})\boldsymbol{\theta}_3 \tag{2.50}$$

where $\boldsymbol{\theta}_3$ is defined by (A.10) and $\mathbf{B}(\mathbf{d})$ is given by (A.12). Equation (2.48) can now be written as

$$\frac{\partial k^{(i)}}{\partial \dot{\mathbf{q}}} = [\mathbf{W}_1(\mathbf{q}, \dot{\mathbf{q}}) \;\; \mathbf{W}_2(\mathbf{q}, \dot{\mathbf{q}})]\boldsymbol{\theta}^{(i)} \tag{2.51}$$

with

$$\boldsymbol{\theta}^{(i)} = \begin{bmatrix} \boldsymbol{\theta}_5 \\ \boldsymbol{\theta}_3 \end{bmatrix} \tag{2.52}$$

By (2.44)-(2.46) the second term in (2.43) becomes

$$-\frac{\partial k^{(i)}}{\partial \mathbf{q}} = -m\frac{\partial {}^0\mathbf{v}_{C_i}^T}{\partial \mathbf{q}} {}^0\mathbf{v}_{C_i} - \frac{\partial(\dot{\mathbf{q}}^T \mathbf{J}_\omega^T {}_i^0\mathbf{R})}{\partial \mathbf{q}} {}^c\mathbf{I}_i {}_0^i\mathbf{R}\mathbf{J}_\omega\dot{\mathbf{q}} \tag{2.53}$$

By repeatedly using (2.41) and (2.42), it is found that

$$\begin{aligned} -m\frac{\partial {}^0\mathbf{v}_{C_i}^T}{\partial \mathbf{q}} {}^0\mathbf{v}_{C_i} &= [\mathbf{Y}_{11}(\mathbf{q},\dot{\mathbf{q}}) \;\; \mathbf{Y}_{12}(\mathbf{q},\dot{\mathbf{q}}) \;\; \mathbf{Y}_{13}(\mathbf{q},\dot{\mathbf{q}}) \;\; \mathbf{Y}_{14}(\mathbf{q},\dot{\mathbf{q}})]\boldsymbol{\theta}_5 \\ &\equiv \mathbf{Y}_1(\mathbf{q},\dot{\mathbf{q}})\boldsymbol{\theta}_5 \end{aligned} \tag{2.54}$$

where

$$\mathbf{Y}_{11}(\mathbf{q},\dot{\mathbf{q}}) = -\mathbf{D}_v^T\mathbf{J}_v\dot{\mathbf{q}} \tag{2.55}$$

$$\mathbf{Y}_{12}(\mathbf{q},\dot{\mathbf{q}}) = -[\mathbf{D}_v \times \mathbf{J}_\omega\dot{\mathbf{q}} + \mathbf{J}_v\dot{\mathbf{q}} \times \mathbf{D}_\omega]^T {}_i^0\mathbf{R} - \mathbf{D}_0 \tag{2.56}$$

$$\mathbf{Y}_{13}(\mathbf{q},\dot{\mathbf{q}}) = -[\frac{\partial \mathbf{J}_\omega^T}{\partial q_1}\mathbf{J}_\omega\dot{\mathbf{q}} \;\; \cdots \;\; \frac{\partial \mathbf{J}_\omega^T}{\partial q_n}\mathbf{J}_\omega\dot{\mathbf{q}}]^T\dot{\mathbf{q}} \tag{2.57}$$

with

$$\mathbf{D}_v = [\frac{\partial \mathbf{J}_v}{\partial q_1}\dot{\mathbf{q}} \;\; \cdots \;\; \frac{\partial \mathbf{J}_v}{\partial q_n}\dot{\mathbf{q}}] \tag{2.58}$$

$$\mathbf{D}_\omega = [\frac{\partial \mathbf{J}_\omega}{\partial q_1}\dot{\mathbf{q}} \;\; \cdots \;\; \frac{\partial \mathbf{J}_\omega}{\partial q_n}\dot{\mathbf{q}}] \tag{2.59}$$

$$\mathbf{D}_0 = [\frac{\partial\, {}_0^i\mathbf{R}}{\partial q_1}(\mathbf{J}_v\dot{\mathbf{q}} \times \mathbf{J}_\omega\dot{\mathbf{q}}) \;\; \cdots \;\; \frac{\partial\, {}_0^i\mathbf{R}}{\partial q_n}(\mathbf{J}_v\dot{\mathbf{q}} \times \mathbf{J}_\omega\dot{\mathbf{q}})]$$

and $\mathbf{Y}_{14}(\mathbf{q},\dot{\mathbf{q}})$ is determined by

$$\begin{bmatrix} {}^i\mathbf{P}_{C_i}^T\mathbf{E}_1\, {}^i\mathbf{P}_{C_i} \\ \vdots \\ {}^i\mathbf{P}_{C_i}^T\mathbf{E}_n\, {}^i\mathbf{P}_{C_i} \end{bmatrix} = \mathbf{Y}_{14}(\mathbf{q},\dot{\mathbf{q}})\boldsymbol{\theta}_4 \tag{2.60}$$

where $\boldsymbol{\theta}_4$ is defined by (A.17), and

$$\mathbf{E}_j = {}_0^i\mathbf{R}\mathbf{J}_\omega\dot{\mathbf{q}}\dot{\mathbf{q}}^T\frac{\partial \mathbf{J}_\omega^T}{\partial q_j} {}_i^0\mathbf{R} - \frac{\partial\, {}_0^i\mathbf{R}}{\partial q_j}(\|\mathbf{J}_\omega\dot{\mathbf{q}}\|\, {}^2\mathbf{I}_3 - \mathbf{J}_\omega\dot{\mathbf{q}}\dot{\mathbf{q}}^T\mathbf{J}_\omega^T) {}_i^0\mathbf{R} \tag{2.61}$$

With $\mathbf{d}$ defined by (2.49) and formula (A.11), the second term in (2.53) can be written as

$$-\frac{\partial(\dot{\mathbf{q}}^T \mathbf{J}_{\omega\,i}^T\, {}^0_i\mathbf{R})}{\partial \mathbf{q}}\, {}^c\mathbf{I}_i\, {}^i_0\mathbf{R}\mathbf{J}_\omega\dot{\mathbf{q}} \;=\; \mathbf{Y}_2(\mathbf{q},\dot{\mathbf{q}})\boldsymbol{\theta}_3 \tag{2.62}$$

where

$$\mathbf{Y}_2(\mathbf{q},\dot{\mathbf{q}}) \;=\; -[\frac{\partial({}^i_0\mathbf{R}\mathbf{J}_\omega)}{\partial q_1}\dot{\mathbf{q}} \;\;\ldots\;\; \frac{\partial({}^i_0\mathbf{R}\mathbf{J}_\omega)}{\partial q_n}\dot{\mathbf{q}}]\mathbf{B}(\mathbf{d}) \tag{2.63}$$

To compute the last term in (2.43), note that

$${}^0\mathbf{v}_{C_i} \;=\; \frac{d}{dt}({}^0\mathbf{P}_{C_i}) \;=\; (\frac{\partial\, {}^0\mathbf{P}_{C_i}^T}{\partial \mathbf{q}})^T\dot{\mathbf{q}}$$

which, in conjunction with (2.45) and (A.3), leads to

$$-m\dot{\mathbf{q}}^T\frac{\partial\, {}^o\mathbf{P}_{C_i}^T}{\partial \mathbf{q}}\, {}^o\mathbf{g} \;=\; \dot{\mathbf{q}}^T\mathbf{Y}_3(\mathbf{q})\boldsymbol{\theta}_2 \tag{2.64}$$

where $\boldsymbol{\theta}_2$ is defined by (A.5) and

$$\mathbf{Y}_3(\mathbf{q}) \;=\; [\mathbf{Y}_{31}(\mathbf{q}) \quad \mathbf{Y}_{32}(\mathbf{q})] \tag{2.65}$$

with

$$\mathbf{Y}_{31}(\mathbf{q}) \;=\; -\mathbf{J}_v^T\, {}^o\mathbf{g} \tag{2.66}$$

$$\mathbf{Y}_{32}(\mathbf{q}) \;=\; (\mathbf{J}_\omega \times {}^o\mathbf{g})^T\, {}^o_i\mathbf{R} \tag{2.67}$$

Using equations (2.36),(2.43),(2.51),(2.53), (2.54),(2.62), and (2.64), we obtain

$$\mathbf{Y}^{(i)}(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}}) \;=\; [\dot{\mathbf{W}}_1(\mathbf{q},\dot{\mathbf{q}}) + \hat{\mathbf{Y}}_1(\mathbf{q},\dot{\mathbf{q}}) \quad \dot{\mathbf{W}}_2(\mathbf{q},\dot{\mathbf{q}}) + \mathbf{Y}_2(\mathbf{q},\dot{\mathbf{q}})] \tag{2.68}$$

where

$$\hat{\mathbf{Y}}_1(\mathbf{q},\dot{\mathbf{q}}) \;=\; [\mathbf{Y}_{11}+\mathbf{Y}_{31} \quad \mathbf{Y}_{12}+\mathbf{Y}_{32} \quad \mathbf{Y}_{13} \quad \mathbf{Y}_{14}] \tag{2.69}$$

Remarks

1. From (2.68) and (2.52) it is observed that although there are only 10 physical parameters involved in a link or load, in general the dimension of $\boldsymbol{\theta}^{(i)}$ is 16.

This parameter redundancy, presented in vector $\boldsymbol{\theta}_5$ which is defined by (A.30), appears to be necessary in order to reform the dynamics so that a *linear* appearance of vector $\boldsymbol{\theta}^{(i)}$ in the dynamics is achieved. On the other hand, if the link (load) has a regular geometry, the dimension of $\boldsymbol{\theta}^{(i)}$ will very likely be reduced, leading to a simplified solution.

2. Another feature of (2.68) is that the formula as it stands is suitable to serve as a starting point to further derive a closed-form solution for the so-called filtered regressor which has been used in several globally stable adaptive control algorithms [155, 35]. This is due to the fact that in (2.68) $\mathbf{Y}^{(i)}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$ depends *implicitly* on $\ddot{\mathbf{q}}$ through the time derivative of $\mathbf{W}_1$ and $\mathbf{W}_2$.

2.3.4 Special Cases

Point-Mass Link (or Load)

If the link (or load) can be treated as a point mass, then $\boldsymbol{\theta}^{(i)} = \boldsymbol{\theta}_5$ which is defined by (A.30). Consequently $\mathbf{Y}^{(i)}$ is given by

$$\mathbf{Y}^{(i)} = \dot{\mathbf{W}}_1(\mathbf{q}, \dot{\mathbf{q}}) + \dot{\mathbf{Y}}_1(\mathbf{q}, \dot{\mathbf{q}}) \tag{2.70}$$

When Axes of $\{C_i\}$ are the Principle Axes of the Link

If link i has a regular geometry, such that the axes of $\{C_i\}$ coincide with the principal axes of the link, i.e.

$${}^{c}\mathbf{I}_i = \text{diag}\{I_{xx}, \ I_{yy}, \ I_{zz}\} \tag{2.71}$$

and if the mass center of the link lies on the x-axis or on the z-axis of frame $\{C_i\}$, then $\boldsymbol{\theta}^{(i)}$ is reduced to a 6-dimensional vector of the form

$$\boldsymbol{\theta}^{(i)} = \begin{cases} [m \ \ mp_x \ \ mp_x^2 \ \ I_{xx} \ \ I_{yy} \ \ I_{zz}]^T, & \text{if the mass center is on the } x\text{-axis} \\ [m \ \ mp_z \ \ mp_z^2 \ \ I_{xx} \ \ I_{yy} \ \ I_{zz}]^T, & \text{if the mass center is on the } z\text{-axis} \end{cases} \tag{2.72}$$

and expressions for matrices $\mathbf{W}_1$, $\mathbf{W}_2$, $\hat{\mathbf{Y}}_1$ and $\mathbf{Y}_2$ can be simplified considerably.

For ${}^i\mathbf{p}_{C_i} = [p_x \;\; 0 \;\; 0]^T$, matrix $\mathbf{W}_1(\mathbf{q}, \dot{\mathbf{q}}) \in R^{n\times 3}$ becomes

$$\mathbf{W}_1(\mathbf{q}, \dot{\mathbf{q}}) = [\mathbf{J}_v^T \mathbf{J}_v \dot{\mathbf{q}} \quad (\mathbf{J}_v \times \mathbf{J}_\omega \dot{\mathbf{q}} + \mathbf{J}_v \dot{\mathbf{q}} \times \mathbf{J}_\omega)^T \, {}^0_i\mathbf{r}_1 \quad \mathbf{J}_\omega^T \mathbf{J}_\omega \dot{\mathbf{q}} - \mathbf{b}] \tag{2.73}$$

where ${}^0_i\mathbf{r}_1$ denotes the first column of ${}^0_i\mathbf{R}$ and $\mathbf{b}$ is defined by $\mathbf{b} = [b_1 \;\; \dots \;\; b_n]^T$ with

$$b_j = {}^0_i\mathbf{r}_1^T \mathbf{J}_{\omega_j} \dot{\mathbf{q}}^T \mathbf{J}_\omega^T \, {}^0_i\mathbf{r}_1, \qquad 1 \le j \le n \tag{2.74}$$

Matrix $\mathbf{W}_2(\mathbf{q}, \dot{\mathbf{q}})$ defined by (2.50) becomes a 3×3 matrix with $\mathbf{B}(\mathbf{d}) \in R^{3\times 3}$ given by

$$\mathbf{B}(\mathbf{d}) = \mathrm{diag}(\, {}^i_o\mathbf{R} \mathbf{J}_\omega \dot{\mathbf{q}}) \tag{2.75}$$

i.e. the 3×3 diagonal matrix with ${}^i_o\mathbf{R}\mathbf{J}_\omega\dot{\mathbf{q}}$ as entries along its main diagonal. Matrix $\hat{\mathbf{Y}}_1$ becomes

$$\hat{\mathbf{Y}}_1 = [\mathbf{Y}_{11} + \mathbf{Y}_{31} \quad \mathbf{y}_{121} + \mathbf{y}_{321} \quad \mathbf{Y}_{13} + \mathbf{y}_{141}] \tag{2.76}$$

where

$$\begin{aligned}
\mathbf{y}_{121} &= -\{[\frac{\partial \mathbf{J}_v}{\partial q_1}\dot{q} \;\; \dots \;\; \frac{\partial \mathbf{J}_v}{\partial q_n}\dot{\mathbf{q}}] \times \mathbf{J}_\omega \dot{\mathbf{q}} + \mathbf{J}_v \dot{\mathbf{q}} [\frac{\partial \mathbf{J}_\omega}{\partial q_1}\dot{\mathbf{q}} \;\; \dots \;\; \frac{\partial \mathbf{J}_\omega}{\partial q_n}\dot{\mathbf{q}}]\}^T \, {}^0_i\mathbf{r}_1 \\
&\quad -[\frac{\partial \, {}^0_i\mathbf{r}_1^T}{\partial q_1}(\mathbf{J}_v \dot{\mathbf{q}} \times \mathbf{J}_\omega \dot{\mathbf{q}}) \;\; \dots \;\; \frac{\partial \, {}^0_i\mathbf{r}_1^T}{\partial q_n}(\mathbf{J}_v \dot{\mathbf{q}} \times \mathbf{J}_\omega \dot{\mathbf{q}})]^T \qquad (2.77) \\
\mathbf{y}_{321} &= -(\mathbf{J}_\omega \times {}^o\mathbf{g})^T \, {}^0_i\mathbf{r}_1 \qquad (2.78) \\
\mathbf{y}_{141} &= [e_{11} \;\; \dots \;\; e_{1n}]^T \qquad (2.79) \\
e_{1j} &= {}^0_i\mathbf{r}_1^T \mathbf{J}_\omega \dot{\mathbf{q}} \dot{\mathbf{q}}^T \frac{\partial \mathbf{J}_\omega^T}{\partial q_j} \, {}^0_i\mathbf{r}_1 - \frac{\partial \, {}^0_i\mathbf{r}_1^T}{\partial q_j}(\|\mathbf{J}_\omega \dot{\mathbf{q}}\|^2 \mathbf{I}_3 - \mathbf{J}_\omega \dot{\mathbf{q}} \dot{\mathbf{q}}^T \mathbf{J}_\omega^T) \, {}^0_i\mathbf{r}_1 \qquad (2.80)
\end{aligned}$$

where $1 \le j \le n$ and $\mathbf{Y}_2 \in R^{n\times 3}$ is given by (2.63) with $\mathbf{B}(\mathbf{d})$ given by (2.75).

For ${}^i\mathbf{p}_{C_i} = [0 \;\; 0 \;\; p_z]^T$, a simplified formula can also be established which is almost identical to (2.73)-(2.80) except that ${}^0_i\mathbf{r}_1$ there should be replaced by ${}^0_i\mathbf{r}_3$ –the third column of ${}^0_i\mathbf{R}$.

2.3.5 An Example

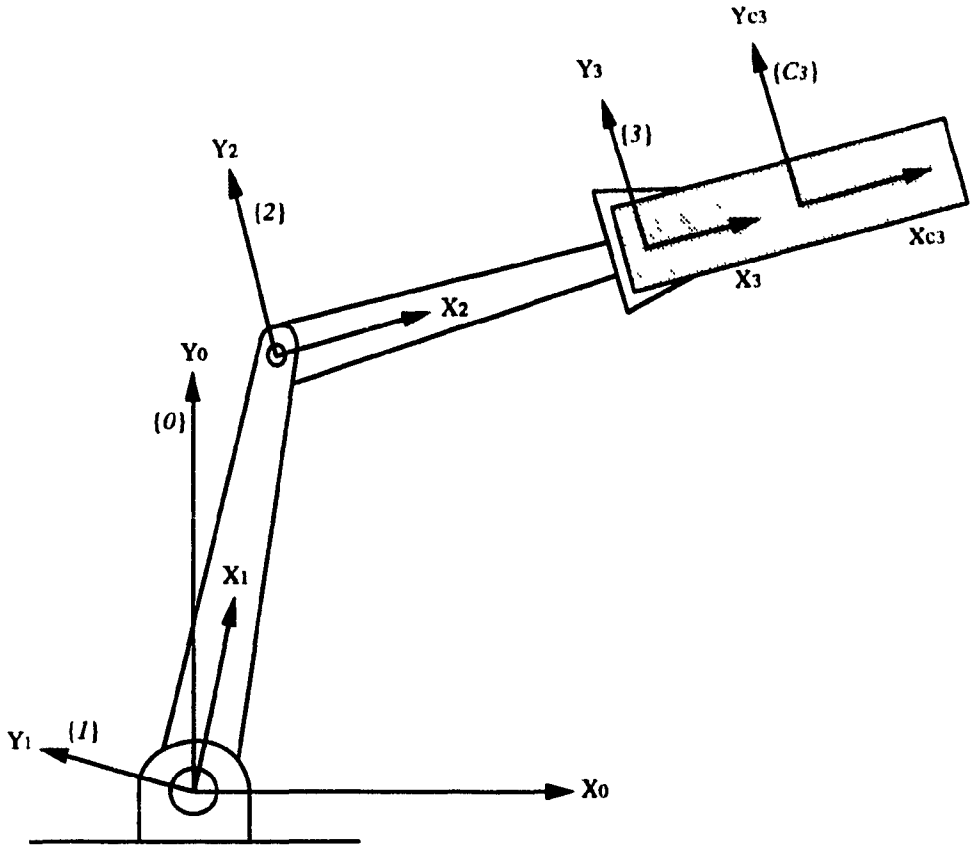


Figure 2.1: A 2 DOF Planar Robot

Consider the 2 DOF vertical planar arm shown in Figure 2.1, where the length of link i is denoted by l_i. It is assumed that the links are of point-mass type with each mass center at the origin of its link frame. The robot handles a rectangular bar with uniform material density. The parameters of the load include mass m_3, mass center ${}^3\mathbf{p}_{C_3}$ and inertia tensor ${}^c\mathbf{I}_{C_3}$ where

$$ {}^3\mathbf{p}_{C_3} = [p_x \quad 0 \quad 0] \quad \text{and} \quad {}^c\mathbf{I}_3 = \text{diag}\,(I_{xx},\, I_{yy},\, I_{zz}) $$

For the comparison purpose, the manipulator regressor will be evaluated using

the conventional approach described in Section 2.2 and the approach proposed in last section.

The Conventional Approach

As mentioned earlier, there are two steps that need to be carried out in this indirect approach.

Step 1. *Establishing the Robot Dynamics*

Following [142], we compute

$$\mathbf{H}(\mathbf{q}) = \sum_{i=1}^{3} [m_i (\hat{\mathbf{J}}_v^{(i)})^T \hat{\mathbf{J}}_v^{(i)} + (\mathbf{J}_\omega^{(i)})^T \, {}^c\mathbf{I}_i \mathbf{J}_\omega^{(i)}]$$

where $\hat{\mathbf{J}}_v^{(i)}$ and $\hat{\mathbf{J}}_\omega^{(i)}$ form the link Jacobian for link i that relates the joint velocity to the velocity of the frame $\{C_i\}$ which is obtained by translating $\{i\}$ to the mass center of link i. In our case, $\{C_1\} = \{1\}$, $\{C_2\} = \{2\}$ and ${}^c\mathbf{I}_1 = {}^c\mathbf{I}_2 = \mathbf{0}$, so $\hat{\mathbf{J}}_v^{(1)} = \mathbf{J}_v^{(1)}$, $\hat{\mathbf{J}}_v^{(2)} = \mathbf{J}_v^{(2)}$ where

$$\mathbf{J}_v^{(1)} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad \text{and} \quad \mathbf{J}_v^{(2)} = \begin{bmatrix} -l_1 s_1 & 0 \\ l_1 c_1 & 0 \\ 0 & 0 \end{bmatrix} \tag{2.81}$$

Since the linear velocity of $\{C_3\}$ is

$$\mathbf{v}_{C_3} = \begin{bmatrix} -l_1 s_1 \dot{\theta}_1 - (l_2 + p_x) s_{12} (\dot{\theta}_1 + \dot{\theta}_2) \\ l_1 c_1 \dot{\theta}_1 + (l_2 + p_x) c_{12} (\dot{\theta}_1 + \dot{\theta}_2) \end{bmatrix}$$

we obtain

$$\hat{\mathbf{J}}_v^{(3)} = \begin{bmatrix} -l_1 s_1 - (l_2 + p_x) s_{12} & -(l_2 + p_x) s_{12} \\ l_1 c_1 + (l_2 + p_x) c_{12} & (l_2 + p_x) c_{12} \end{bmatrix}$$

Further notice that

$$\mathbf{J}_\omega^{(3)} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{bmatrix}$$

hence

$$\mathbf{H(q)} = m_2(\mathbf{J}_v^{(2)})^T\mathbf{J}_v^{(2)} + m_3(\hat{\mathbf{J}}_v^{(3)})^T\hat{\mathbf{J}}_v^{(3)} + (\mathbf{J}_\omega^{(3)})^T\ {}^c\mathbf{I}_3\mathbf{J}_\omega^{(3)}$$

$$= \begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix}$$

where

$$\begin{aligned} H_{11} &= l_1^2 m_2 + [l_1^2 + (l_2+p_x)^2 + 2l_1(l_2+p_x)c_2]m_3 + I_{zz} \\ H_{12} &= [(l_2+p_x)^2 + l_1(l_2+p_x)c_2]m_3 + I_{zz} \\ H_{21} &= [(l_2+p_x)^2 + l_1(l_2+p_x)c_2]m_3 + I_{zz} \\ H_{22} &= (l_2+p_x)^2 m_3 + I_{zz} \end{aligned}$$

It follows from [142] that the term $\mathbf{C}\dot{\mathbf{q}}$ in (1.1) is given by

$$\mathbf{C(q,\dot{q})\dot{q}} = \begin{bmatrix} \sum_{j=1}^{2}(\frac{\partial H_{1j}}{\partial q_1} - \frac{1}{2}\frac{\partial H_{j1}}{\partial q_1})\dot{q}_j & \sum_{j=1}^{2}(\frac{\partial H_{1j}}{\partial q_2} - \frac{1}{2}\frac{\partial H_{j2}}{\partial q_1})\dot{q}_j \\ \sum_{j=1}^{2}(\frac{\partial H_{2j}}{\partial q_1} - \frac{1}{2}\frac{\partial H_{j1}}{\partial q_2})\dot{q}_j & \sum_{j=1}^{n}(\frac{\partial H_{2j}}{\partial q_2} - \frac{1}{2}\frac{\partial H_{j2}}{\partial q_2})\dot{q}_j \end{bmatrix}\dot{\mathbf{q}}$$

$$= \begin{bmatrix} -l_1(l_2+p_x)s_2 m_3 \dot{q}_2(2\dot{q}_1 + \dot{q}_2) \\ l_1(l_2+p_x)s_2 m_3 \dot{q}_1^2 \end{bmatrix}$$

Finally, we compute the total potential energy of the system as

$$\mathcal{U} = m_2 g l_1 s_1 + m_3 g[l_1 s_1 + (l_2+p_x)s_{12}]$$

hence

$$\mathbf{g} = \begin{bmatrix} \partial\mathcal{U}/\partial q_1 \\ \partial\mathcal{U}/\partial q_2 \end{bmatrix} = \begin{bmatrix} (m_2+m_3)l_1 g c_1 + m_3 g(l_2+p_x)c_{12} \\ m_3 g(l_2+p_x)c_{12} \end{bmatrix}$$

Step 2. *Parameter Extraction*

As mentioned in the introduction section, this step begins with defining a parameter vector $\boldsymbol{\theta}$. Suppose that the parameters of interest are link mass m_2 (note that the link mass m_1 plays no role in the dynamics), and the parameters associated with the load, i.e. m_3, p_x, and I_{zz}. From what we have done in step 1 it is observed that every entry of $\mathbf{H}$, $\mathbf{C}$ and $\mathbf{g}$ involves some of these parameters. In addition, in a number of entries parameters present themselves in a nonlinear manner such as $m_3 p_x^2$. It takes a while to figure out that

$$\boldsymbol{\theta} = [m_2 \quad m_3 \quad m_3 p_x \quad m_3 p_x^2 \quad I_{zz}]^T$$

defines a parameter vector with minimum dimension such that for every entry of $\mathbf{H}$, $\mathbf{C}$ and $\mathbf{g}$, the parameters of interests can all be extracted. Note that one does not need to include p_x as a separate component in $\boldsymbol{\theta}$ as parameter p_x in all the terms involved always presents itself together with m_3. Obviously, such a parameter vector cannot be adequately defined before a careful inspection of all entries is done. For a robot with more non-point-mass links, this step will become quite involved and time consuming.

With $\boldsymbol{\theta}$ defined above, we can now extract it from the entries of $\mathbf{H}$, $\mathbf{C}$ and $\mathbf{g}$ as follows

$$\begin{aligned}
H_{11} &= [l_1^2 \quad l_1^2 + 2l_1 l_2 c_2 + l_2^2 \quad 2(l_1 c_2 + l_2) \quad 1 \quad 1]\boldsymbol{\theta} \triangleq \mathbf{h}_{11}\boldsymbol{\theta} \\
H_{12} &= [0 \quad l_2^2 + l_1 l_2 c_2 \quad l_1 c_2 + 2l_2 \quad 1 \quad 1]\boldsymbol{\theta} \triangleq \mathbf{h}_{12}\boldsymbol{\theta} \\
H_{22} &= [0 \quad l_2^2 \quad 2l_2 \quad 1 \quad 1]\boldsymbol{\theta} \triangleq \mathbf{h}_{22}\boldsymbol{\theta}
\end{aligned}$$

$$\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} = \begin{bmatrix} 0 & -l_1 l_2 s_2 \dot{q}_2 (2\dot{q}_1 + \dot{q}_2) & -l_1 s_2 \dot{q}_2 (2\dot{q}_1 + \dot{q}_2) & 0 & 0 \\ 0 & l_1 l_2 s_2 \dot{q}_1^2 & l_1 s_2 \dot{q}_1^2 & 0 & 0 \end{bmatrix} \boldsymbol{\theta} \triangleq \mathbf{Y}_c \boldsymbol{\theta}$$

$$\mathbf{g} = \begin{bmatrix} l_1 c_1 g & (l_1 c_1 + l_2 c_{12}) g & c_{12} g & 0 & 0 \\ 0 & l_2 c_{12} g & c_{12} g & 0 & 0 \end{bmatrix} \boldsymbol{\theta} \triangleq \mathbf{Y}_g \boldsymbol{\theta}$$

Therefore, the manipulator regressor associated with θ is given by

$$\begin{aligned}\mathbf{Y} &= \begin{bmatrix}\mathbf{h}_{11}\ddot{q}_1 + \mathbf{h}_{12}\ddot{q}_2 \\ \mathbf{h}_{12}\ddot{q}_1 + \mathbf{h}_{11}\ddot{q}_2\end{bmatrix} + \mathbf{Y}_c + \mathbf{Y}_g \\ &= \begin{bmatrix}Y_{(11)} & Y_{(12)} & Y_{(13)} & Y_{(14)} & Y_{(15)} \\ Y_{(21)} & Y_{(22)} & Y_{(23)} & Y_{(24)} & Y_{(25)}\end{bmatrix}\end{aligned} \tag{2.82}$$

where

$$\begin{aligned}
Y_{(11)} &= l_1(\ddot{q}_1 + c_1 g) \\
Y_{(12)} &= (l_1^2 + 2l_1l_2 + l_2^2)\ddot{q}_1 + l_2(l_1c_2 + l_2)\ddot{q}_2 - l_1l_2s_2\dot{q}_2(2\dot{q}_1 + q_2) + (l_1c_1 + l_2c_{12})g \\
Y_{(13)} &= 2(l_1c_2 + l_2)\ddot{q}_1 + (l_1c_2 + 2l_2)\ddot{q}_2 - l_1s_2\dot{q}_2(2\dot{q}_1 + \dot{q}_2) + c_{12}g \\
Y_{(14)} &= \ddot{q}_1 + \ddot{q}_2 \\
Y_{(15)} &= \ddot{q}_1 + \ddot{q}_2 \\
Y_{(21)} &= 0 \\
Y_{(22)} &= l_2(l_1c_2 + l_2)\ddot{q}_1 + l_2^2\ddot{q}_2 + l_1l_2s_2\dot{q}_1^2 + l_2c_{12}g \\
Y_{(23)} &= (l_1c_2 + 2l_2)\ddot{q}_1 + 2l_2^2\ddot{q}_2 + l_1s_2\dot{q}_1^2 + c_{12}g \\
Y_{(24)} &= \ddot{q}_1 + \ddot{q}_2 \\
Y_{(25)} &= \ddot{q}_1 + \ddot{q}_2
\end{aligned}$$

The proposed Approach

The problem of computing $\mathbf{Y}^{(3)}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$ falls obviously within the special case addressed in previous section. Using

$$\mathbf{J}_v(\mathbf{q}) = \begin{bmatrix}-l_1s_1 - l_2s_{12} & -l_2s_{12} \\ l_1c_1 + l_2c_{12} & l_2c_{12} \\ 0 & 0\end{bmatrix} \quad \text{and} \quad \mathbf{J}_\omega(q) = \begin{bmatrix}0 & 0 \\ 0 & 0 \\ 1 & 1\end{bmatrix}$$

(2.73) becomes

$$\mathbf{W}_1(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix}W_{1(11)} & W_{1(12)} & W_{1(13)} \\ W_{1(21)} & W_{1(22)} & W_{1(23)}\end{bmatrix}$$

where

$$
\begin{aligned}
W_{1(11)} &= (l_1^2 + l_2^2 + 2l_1l_2c_2)\dot{q}_1 + (l_2^2 + l_1l_2c_2)\dot{q}_2 \\
W_{1(12)} &= 2(l_1c_2 + l_2)\dot{q}_1 + (l_1c_2 + 2l_2)\dot{q}_2 \\
W_{1(13)} &= \dot{q}_1 + \dot{q}_2 \\
W_{1(21)} &= (l_2^2 + l_1l_2c_2)\dot{q}_1 + l_2^2\dot{q}_2 \\
W_{1(22)} &= (l_1c_2 + 2l_2)\dot{q}_1 + 2l_2\dot{q}_2 \\
W_{1(23)} &= \dot{q}_1 + \dot{q}_2
\end{aligned}
$$

Since $\mathbf{B}(\mathbf{d}) = \text{diag}(0,\ 0,\ \dot{q}_1 + \dot{q}_2)$, equation (2.50) gives

$$
\mathbf{W}_2(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} 0 & 0 & \dot{q}_1 + \dot{q}_2 \\ 0 & 0 & \dot{q}_1 + \dot{q}_2 \end{bmatrix}
$$

By (2.55), (2.66) and (2.77)-(2.80), we obtain $\hat{\mathbf{Y}}_1$ with

$$
\begin{aligned}
\mathbf{Y}_{11} + \mathbf{Y}_{31} &= \begin{bmatrix} (l_1c_1 + l_2c_{12})g \\ l_1l_2s_2\dot{q}_1(\dot{q}_1 + \dot{q}_2) + l_2c_{12}g \end{bmatrix} \\
\mathbf{y}_{121} + \mathbf{y}_{321} &= \begin{bmatrix} c_{12}g \\ l_1s_2\dot{q}_1(\dot{q}_1 + \dot{q}_2) + c_{12}g \end{bmatrix} \\
\mathbf{y}_{141} &= \begin{bmatrix} 0 \\ 0 \end{bmatrix}
\end{aligned}
$$

Further, by (2.63) it is found that $\mathbf{Y}_2(\mathbf{q}, \dot{\mathbf{q}})$ is a 2×3 zero matrix.

Since $\mathbf{Y}_2$ and the first two columns of $\mathbf{W}_2$ are all zero, $\boldsymbol{\theta}^{(3)}$ can be re-defined as

$$
\boldsymbol{\theta}^{(3)} = [m_3 \quad m_3p_x \quad m_3p_x^2 \quad I_{zz}]^T \tag{2.83}
$$

and $\mathbf{Y}^{(3)}$ is now obtained as

$$\mathbf{Y}^{(3)} = \begin{bmatrix} (l_1^2 + 2l_1l_2 + l_2^2)\ddot{q}_1 & 2(l_1c_2 + l_2)\ddot{q}_1 & & \\ +l_2(l_1c_2 + l_2)\ddot{q}_2 & +(l_1c_2 + 2l_2)\ddot{q}_2 & \ddot{q}_1 + \ddot{q}_2 & \ddot{q}_1 + \ddot{q}_2 \\ -l_1l_2s_2\dot{q}_2(2\dot{q}_1 + q_2) & -l_1s_2\dot{q}_2(2\dot{q}_1 + \dot{q}_2) & & \\ +(l_1c_1 + l_2c_{12})g & +c_{12}g & & \\ \\ l_2(l_1c_2 + l_2)\ddot{q}_1 + l_2^2\ddot{q}_2 & (l_1c_2 + 2l_2)\ddot{q}_1 + 2l_2^2\ddot{q}_2 & & \\ +l_1l_2s_2\dot{q}_1^2 + l_2c_{12}g & +l_1s_2\dot{q}_1^2 + c_{12}g & \ddot{q}_1 + \ddot{q}_2 & \ddot{q}_1 + \ddot{q}_2 \end{bmatrix} \tag{2.84}$$

To compute $\mathbf{Y}^{(1)}$ and $\mathbf{Y}^{(2)}$, note that each link is of point-mass involving only one parameter – its link mass. The link Jacobians for link 1 and 2 are given by (2.81), and (2.70) gives

$$\mathbf{Y}^{(1)} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad \mathbf{Y}^{(2)} = \begin{bmatrix} l_1^2\ddot{q}_1 + l_1c_1g \\ 0 \end{bmatrix} \tag{2.85}$$

Since the link mass m_1 does not play a role in the dynamics, $\mathbf{Y}_1$ (the zero column) can be dropped in $\mathbf{Y}$, i.e.

$$\mathbf{Y} = [\mathbf{Y}^{(2)} \quad \mathbf{Y}^{(3)}] \tag{2.86}$$

with $\boldsymbol{\theta} = [m_2 \quad m_3 \quad m_3p_x \quad m_3p_x^2 \quad I_{zz}]^T$.

On comparing these two approaches and their results namely (2.84)-(2.86) and (2.82), it is observed that the proposed approach finds the regressor by directly and therefore more efficiently manipulating the link Jacobians. As our method is evolved from Lagrange's equation of motion, it may be viewed as a counterpart of the Lagrangian formulation for the evaluation of regressor dynamics.

2.3.6 Recursive Formulation of Regressor Dynamics

We begin our analysis by re-formulating the Newton-Euler recursion introduced in section 2.2.1, using the techniques developed in Appendix A. For the sake of

simplicity, all joints are assumed to be rotational. Notation adopted below will be consistent with those used in section 2.2.1.

Outward recursions (i: $0 \rightarrow n$)

As the computation of ${}^{i+1}\boldsymbol{\omega}_{i+1}$, ${}^{i+1}\dot{\boldsymbol{\omega}}_{i+1}$, ${}^{i+1}\dot{\mathbf{v}}_{i+1}$ involves no dynamic parameters (i.e, the link mass, position of the link's mass center and inertia tensor), we only need to write

$$\begin{aligned}
{}^{i+1}\dot{\mathbf{v}}_{c_{i+1}} &= {}^{i+1}\boldsymbol{\omega}_{i+1} \times {}^{i+1}\mathbf{P}_{c_{i+1}} + {}^{i+1}\boldsymbol{\omega}_{i+1} \times ({}^{i+1}\boldsymbol{\omega}_{i+1} \times {}^{i+1}\mathbf{P}_{c_{i+1}}) + {}^{i+1}\dot{\mathbf{v}}_{i+1} \\
{}^{i+1}\mathbf{F}_{i+1} &= m_{i+1}\, {}^{i+1}\dot{\mathbf{v}}_{c_{i+1}} \\
{}^{i+1}\mathbf{N}_{i+1} &= {}^{c}\mathbf{I}_{i+1}\dot{\boldsymbol{\omega}}_{i+1} + {}^{i+1}\boldsymbol{\omega}_{i+1} \times {}^{c}\mathbf{I}_{i+1}\boldsymbol{\omega}_{i+1}
\end{aligned}$$

By (A.8), the outward recursion [page 29] implies that

$${}^{i+1}\mathbf{F}_{i+1} = \mathbf{A}_{i+1}\boldsymbol{\theta}_2^{(i+1)} \tag{2.87}$$

where $\boldsymbol{\theta}_2^{(i+1)}$ is defined by $\boldsymbol{\theta}_2$ in (A.5) with the understanding that m_{i+1} is the mass of link $i+1$, ${}^{i+1}\mathbf{P}_{c_{i+1}} = [p_x \quad p_y \quad p_z]^T$, and

$$\mathbf{A}_{i+1} = [{}^{i+1}\dot{\mathbf{v}}_{i+1} \quad \mathbf{H}_{i+1}] \tag{2.88}$$

$$\mathbf{H}_{i+1} = \boldsymbol{\Omega}({}^{i+1}\dot{\boldsymbol{\omega}}_{i+1}) + {}^{i+1}\mathbf{U}_{i+1} - \|{}^{i+1}\boldsymbol{\omega}_{i+1}\|^2\mathbf{I} \tag{2.89}$$

$${}^{i+1}\mathbf{U}_{i+1} = {}^{i+1}\boldsymbol{\omega}_{i+1}\, {}^{i+1}\boldsymbol{\omega}_{i+1}^T$$

By (A.11), we obtain

$${}^{i+1}\mathbf{N}_{i+1} = \mathbf{E}_{i+1}\boldsymbol{\theta}_3^{(i+1)} \tag{2.90}$$

where $\boldsymbol{\theta}_3^{(i+1)}$ is defined by $\boldsymbol{\theta}_3$ in (A.10) with $I_{xx}, \ldots, I_{yz}$ from ${}^{c}\mathbf{I}_{i+1}$, and

$$\mathbf{E}_{i+1} = \mathbf{B}({}^{i+1}\dot{\boldsymbol{\omega}}_{i+1}) + \boldsymbol{\Omega}({}^{i+1}\boldsymbol{\omega}_{i+1})\mathbf{B}({}^{i+1}\boldsymbol{\omega}_{i+1}) \tag{2.91}$$

Inward recursions (i: $n+1 \rightarrow 1$)

If no contact occurs between the load and its environment, we can begin the inward recursion with zero boundary conditions

$$ {}^{n+2}\mathbf{f}_{n+2} = 0, \qquad {}^{n+2}\mathbf{n}_{n+2} = 0 $$

By (2.87) and (2.90), the inward recursions in section 2.2.1 imply that

$$ \begin{aligned} {}^{n+1}\mathbf{f}_{n+1} &= {}^{n+1}\mathbf{F}_{n+1} = \mathbf{A}_{n+1}\boldsymbol{\theta}_2^{(n+1)} \\ {}^{n+1}\mathbf{n}_{n+1} &= \mathbf{E}_{n+1}\boldsymbol{\theta}_3^{(n+1)} + {}^{n+1}\mathbf{P}_{c_{n+1}} \times \mathbf{A}_{n+1}\boldsymbol{\theta}_2^{(n+1)} \end{aligned} \tag{2.92} $$

where

$$ {}^{n+1}\mathbf{P}_{c_{n+1}} \times \mathbf{A}_{n+1}\boldsymbol{\theta}_2^{(n+1)} = m_{n+1}\, {}^{n+1}\mathbf{P}_{c_{n+1}} \times ({}^{n+1}\dot{\mathbf{v}}_{n+1} + \mathbf{H}_{n+1}\, {}^{n+1}\mathbf{P}_{c_{n+1}}) $$

with m_{n+1} the mass of link $n+1$ (i.e. the load) and $\mathbf{H}_{n+1}$ is defined in (2.89). From (A.13)-(A.21) it follows that

$$ {}^{n+1}\mathbf{P}_{c_{n+1}} \times \mathbf{A}_{n+1}\boldsymbol{\theta}_2^{(n+1)} = \boldsymbol{\Phi}_{n+1} \begin{bmatrix} \boldsymbol{\theta}_1^{(n+1)} \\ \boldsymbol{\theta}_4^{(n+1)} \end{bmatrix} \tag{2.93} $$

where $\boldsymbol{\theta}_1^{(n+1)}$ and $\boldsymbol{\theta}_4^{(n+1)}$ are defined by (A.2) and (A.17) respectively, with

$$ m = m_{n+1}, \qquad [p_x \;\; p_y \;\; p_z]^T = {}^{n+1}\mathbf{P}_{c_{n+1}}, $$

and

$$ \boldsymbol{\Phi}_{n+1} = [-\boldsymbol{\Omega}({}^{n+1}\dot{\mathbf{v}}_{n+1}) \quad \boldsymbol{\Psi}_{n+1}] \tag{2.94} $$

with $\boldsymbol{\Psi}_{n+1}$ defined by (A.19). Hence

$$ {}^{n+1}\mathbf{n}_{n+1} = \boldsymbol{\Pi}_{n+1}\boldsymbol{\theta}^{(n+1)} \tag{2.95} $$

where

$$ \boldsymbol{\Pi}_{n+1} = [\boldsymbol{\phi} \quad \boldsymbol{\Phi}_{n+1} \quad \mathbf{E}_{n+1}] \tag{2.96} $$

with $\boldsymbol{\phi} = [0 \;\; 0 \;\; 0]^T$.

To obtain the recursive relation of $\boldsymbol{\Pi}_{i+1}$ to $\boldsymbol{\Pi}_i$ for $i = n, n-1, \ldots, 1$, we first write the vector $\boldsymbol{\theta}$ in (2.34) as

$$\boldsymbol{\theta} = [\boldsymbol{\theta}_2^{(1)'} \quad \boldsymbol{\theta}_4^{(1)'} \quad \boldsymbol{\theta}_3^{(1)'} \quad \boldsymbol{\theta}_2^{(2)'} \quad \boldsymbol{\theta}_4^{(2)'} \quad \boldsymbol{\theta}_3^{(2)'} \quad \ldots \quad \boldsymbol{\theta}_2^{(n+1)'} \quad \boldsymbol{\theta}_4^{(n+1)'} \quad \boldsymbol{\theta}_3^{(n+1)'}]^T \tag{2.97}$$

and write (2.92) and (2.95) as

$$^{n+1}\mathbf{f}_{n+1} = \tilde{\mathbf{A}}_{n+1}\boldsymbol{\theta} \tag{2.98}$$

with

$$\tilde{\mathbf{A}}_{n+1} = [0 \quad 0 \quad \ldots \quad 0 \quad \mathbf{A}_{n+1} \quad 0] \tag{2.99}$$

and

$$^{n+1}\mathbf{n}_{n+1} = \tilde{\boldsymbol{\Pi}}_{n+1}\boldsymbol{\theta} \tag{2.100}$$

with

$$\tilde{\boldsymbol{\Pi}}_{n+1} = [0 \quad 0 \quad \ldots \quad 0 \quad \boldsymbol{\Pi}_{n+1}] \tag{2.101}$$

where $\mathbf{A}_{n+1}$ is placed in a position consistent with that of $\boldsymbol{\theta}_2^{(n+1)}$ in (2.97), and $\boldsymbol{\Pi}_{n+1}$ is placed in a position consistent with that of $\boldsymbol{\theta}^{(n+1)}$. By (2.98) and (2.87), the inward recursions for $1 \leq i \leq n$ give

$$\begin{aligned} ^{i}\mathbf{f}_i &= {}^{i}_{i+1}\mathbf{R}\ {}^{i+1}\mathbf{f}_{i+1} + {}^{i}\mathbf{F}_i \\ &= {}^{i}_{i+1}\mathbf{R}\tilde{\mathbf{A}}_{i+1}\boldsymbol{\theta} + \mathbf{A}_i\boldsymbol{\theta}_2^{(i)} \\ &= \tilde{\mathbf{A}}_i\boldsymbol{\theta} \end{aligned} \tag{2.102}$$

where

$$\tilde{\mathbf{A}}_i = {}^{i}_{i+1}\mathbf{R}\tilde{\mathbf{A}}_{i+1} + [0 \quad \ldots \quad 0 \quad \mathbf{A}_i \quad 0 \quad \ldots \quad 0] \tag{2.103}$$

with $\mathbf{A}_i$ placed in a position consistent with that of $\boldsymbol{\theta}_2^{(i)}$ in (2.97). Further, by (2.90), (2.100), (2.93), (A.6), and (2.102), the inward iterations give

$$^{i}\mathbf{n}_i = \tilde{\boldsymbol{\Pi}}_i\boldsymbol{\theta} \tag{2.104}$$

where

$$\tilde{\boldsymbol{\Pi}}_i = {}^i_{i+1}\mathbf{R}\tilde{\boldsymbol{\Pi}}_{i+1} + \boldsymbol{\Omega}({}^i\mathbf{P}_{i+1})\,{}^i_{i+1}\mathbf{R}\tilde{\mathbf{A}}_{i+1} + [0 \quad \ldots \quad 0 \quad \boldsymbol{\Phi}_i \quad \mathbf{E}_i \quad 0 \quad \ldots \quad 0] \tag{2.105}$$

with $\boldsymbol{\Phi}_i$ placed in a position consistent with that of $\boldsymbol{\theta}_1^{(i)}$ (which is the last 3 components of $\boldsymbol{\theta}_2^{(i)}$) and $\boldsymbol{\theta}_4^{(i)}$, and $\mathbf{E}_i$ placed in a position consistent with that of $\boldsymbol{\theta}_3^{(i)}$, respectively.

We now obtain the manipulator dynamics as $\boldsymbol{\tau} = \mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})\boldsymbol{\theta}$ where

$$\mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) = \begin{bmatrix} \mathbf{e}_z^T \tilde{\boldsymbol{\Pi}}_1 \\ \vdots \\ \mathbf{e}_z^T \tilde{\boldsymbol{\Pi}}_n \end{bmatrix} \tag{2.106}$$

with $\mathbf{e}_z^T = [0 \quad 0 \quad 1]$ and $\tilde{\boldsymbol{\Pi}}_i$ given by (2.105) for $1 \leq i \leq n$. Obviously, if joint i is not rotational but prismatic, then the ith row of $\mathbf{Y}$ in (2.106) should be $\mathbf{e}_z^T \tilde{\mathbf{A}}_i$.

2.3.7 Computation of Regressor Dynamics: Algorithm 2.1

The computation of $\mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$ in (2.106) can be accomplished by following the algorithm listed below.

Algorithm 2.1 Computation of $\mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$

Step 1: Compute $\mathbf{A}_{n+1}$ using (2.88), (2.89);

Step 2: Compute $\mathbf{E}_{n+1}$ using (2.90), (2.91);

Step 3: Compute $\boldsymbol{\Phi}_{n+1}$ using (2.94);

Step 4: Form $\tilde{\mathbf{A}}_{n+1}$ and $\tilde{\boldsymbol{\Pi}}_{n+1}$ using (2.99) and (2.96), (2.101), respectively.

Step 5: For $i = n, n-1, \ldots, 1$

- Compute $\mathbf{A}_i$ using (2.88), (2.89);
- Compute $\mathbf{E}_i$ using (2.90), (2.91);
- Compute $\boldsymbol{\Phi}_i$ using (2.94);
- Compute $\tilde{\mathbf{A}}_i$ using (2.103);
- Compute $\tilde{\boldsymbol{\Pi}}_i$ using (2.105).

Step 6: Form $\mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$ where

$$\text{The } i\text{th row of } \mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) \;=\; \begin{cases} \mathbf{e}_z^T \tilde{\boldsymbol{\Pi}}_i, & \text{if joint } i \text{ is rotational.} \\ \mathbf{e}_z^T \tilde{\mathbf{A}}_i, & \text{if joint } i \text{ is prismatic.} \end{cases}$$

As the above algorithm is a re-formulation of the Newton-Euler recursion, it is therefore $O(n)$ in complexity for numerical evaluation of the regressor [3, Chapter 6]. In a simulation study, we have utilized the MATLAB codes of the recursive algorithm in two adaptive motion control algorithms to compute the regressor at each control instant, and the results will be presented in the next section. Note that, formula (2.106) can also be used to derive a symbolic solution for the regressor without explicitly formulating the link Jacobians.

2.3.8 An Example

We consider the robot used in previous example (see Figure 2.1) and perform the recursive algorithm to compute its regressor as follows.

Following the steps given in previous section, we compute

$$\mathbf{A}_3 \;=\; [{}^3\dot{\mathbf{v}}_3 \quad \boldsymbol{\Omega}({}^3\dot{\omega}_3) + {}^3\mathbf{U}_3 - \|{}^3\omega_3\|^2 b\mathbf{I}]$$

$$= \begin{bmatrix} \beta_1 & -(\dot{q}_1+\dot{q}_2)^2 & -(\ddot{q}_1+\ddot{q}_2) & 0 \\ \beta_2 & q_1+\ddot{q}_2 & -(\dot{q}_1+\dot{q}_2)^2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

where

$$\begin{aligned} \beta_1 &= l_1 s_2 \ddot{q}_1 - l_1 c_2 \dot{q}_1^2 + s_{12} g - l_2(\dot{q}_1+\dot{q}_2)^2 \\ \beta_2 &= l_1 c_2 \ddot{q}_1 + l_1 s_2 \dot{q}_1^2 + c_{12} g + l_2(\ddot{q}_1+\ddot{q}_2) \end{aligned}$$

Due to the regular geometry of the load, we have $p_y = p_z = 0$, hence $\boldsymbol{\theta}_2^{(3)}$ can be regarded as a two-dimensional vector

$$\boldsymbol{\theta}_2^{(3)} = [m_3 \quad m_3 p_x]^T$$

Consequently, the last two columns of matrix $\mathbf{A}_3$ are not needed in the computation and $\mathbf{A}_3$ can be re-defined as

$$\mathbf{A}_3 = \begin{bmatrix} \beta_1 & -(\dot{q}_1+\dot{q}_2)^2 \\ \beta_2 & (\ddot{q}_1+\ddot{q}_2) \\ 0 & 0 \end{bmatrix}$$

Similarly, we compute

$$\mathbf{E}_3 = \begin{bmatrix} 0 \\ 0 \\ \ddot{q}_1+\ddot{q}_2 \end{bmatrix}$$

with

$$\boldsymbol{\theta}_3^{(3)} = I_{zz}$$

Next note that

$$\begin{bmatrix} \theta_1^{(3)} \\ \theta_4^{(3)} \end{bmatrix} = \begin{bmatrix} m_3 p_x \\ m_3 p_x^2 \end{bmatrix}$$

hence we only need the first and fourth columns of $\boldsymbol{\Phi}_3$. By (2.94) it is found

$$\boldsymbol{\Phi}_3 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \beta_2 & \ddot{q}_1 + \ddot{q}_2 \end{bmatrix}$$

Using (2.99), (2.96) and (2.101), we have

$$\tilde{\mathbf{A}}_3 = \begin{bmatrix} 0 & 0 & \beta_1 & -(\dot{q}_1 + \dot{q}_2)^2 & 0 & 0 \\ 0 & 0 & \beta_2 & \ddot{q}_1 + \ddot{q}_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

and

$$\tilde{\boldsymbol{\Pi}}_3 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \beta_2 & \ddot{q}_1 + \ddot{q}_2 & \ddot{q}_1 + \ddot{q}_2 \end{bmatrix}$$

Furthermore, using (2.105), it is found that

$$\tilde{\boldsymbol{\Pi}}_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -(\ddot{q}_1 + \ddot{q}_2) & 0 & 0 \\ 0 & 0 & l_2\beta_2 & l_2(\ddot{q}_1 + \ddot{q}_2) & \ddot{q}_1 + \ddot{q}_2 & \ddot{q}_1 + \ddot{q}_2 \end{bmatrix}$$

and

$$\tilde{\boldsymbol{\Pi}}_1 = \begin{bmatrix} 0 & 0 & 0 & * & 0 & 0 \\ 0 & 0 & 0 & * & 0 & 0 \\ 0 & \pi_{32} & \pi_{33} & \pi_{34} & \ddot{q}_1 + \ddot{q}_2 & \ddot{q}_1 + \ddot{q}_2 \end{bmatrix}$$

where

$$\begin{aligned}\pi_{32} &= l_1^2\ddot{q}_1 + l_1 c_1 g \\ \pi_{33} &= l_1(s_2\beta_1 + c_2\beta_2) + l_2\beta_2 \\ \pi_{34} &= (l_2 + l_1 c_2)(\ddot{q}_1 + \ddot{q}_2) - l_1 s_2(\dot{q}_1 + \dot{q}_2)^2 + \beta_2\end{aligned}$$

The parameter vector associated with $\tilde{\boldsymbol{\Pi}}_i$ is

$$\boldsymbol{\theta} = [m_2 \quad m_3 \quad m_3 p_x \quad m_3 p_x^2 \quad I_{zz}]$$

Using (2.106), the corresponding regressor $\mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$ is given by

$$\mathbf{Y} = \begin{bmatrix} \pi_{32} & \pi_{33} & \pi_{34} & \ddot{q}_1 + \ddot{q}_2 & \ddot{q}_1 + \ddot{q}_2 \\ 0 & l_2\beta_2 & l_2(\ddot{q}_1 + \ddot{q}_2) + \beta_2 & \ddot{q}_1 + \ddot{q}_2 & \ddot{q}_1 + \ddot{q}_2 \end{bmatrix}$$

Note that the $\mathbf{Y}$ obtained is identical to the one given by (2.86).

2.3.9 Application to Adaptive Control: Modified Slotine-Li Adaptive Algorithm

From Section 1.1.3, it is obvious that Craig's adaptive algorithm [34] can be directly implemented using regressor dynamics formulation. But Slotine-Li's adaptive algorithm needs to be modified before it can be implemented using regressor dynamics formula.

In this section, a modified version of the Slotine-Li adaptive scheme [37] is presented, and its global stability is shown through a Lyapunov approach. Unlike the algorithm in [37], the modified version enables us to directly use the regressor $\mathbf{Y}$ in controller implementation. Contrary to Craig's algorithm introduced in previous

section, however, the approach proposed below does not require the use of joint acceleration measurements $\ddot{\mathbf{q}}$ and inversion of the estimated mass matrix $\hat{\mathbf{H}}$.

The Dynamic Model

The equations of motion of a robot manipulator as introduced earlier in (1.1) is

$$\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau} \tag{2.107}$$

The Error Equation

Define

$$\dot{\mathbf{q}}_r = \dot{\mathbf{q}}_d - \boldsymbol{\Lambda}\tilde{\mathbf{q}} \tag{2.108}$$

where $\dot{\mathbf{q}}_d$ is the desired velocity, $\tilde{\mathbf{q}} = \mathbf{q} - \mathbf{q}_d$, and $\boldsymbol{\Lambda} > 0$, and recall that

$$\tilde{\mathbf{H}}\ddot{\mathbf{q}}_r + \tilde{\mathbf{C}}\dot{\mathbf{q}} + \tilde{\mathbf{g}} = \mathbf{Y}(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}}_r)\tilde{\boldsymbol{\theta}}_u \tag{2.109}$$

The Control and Adaptive Algorithm

If a control torque vector $\boldsymbol{\tau}$ is assigned as

$$\boldsymbol{\tau} = \hat{\mathbf{H}}\ddot{\mathbf{q}}_r + \hat{\mathbf{C}}\dot{\mathbf{q}} + \hat{\mathbf{g}} - \mathbf{K}\mathbf{s} \tag{2.110}$$

$$= \mathbf{Y}(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}}_r)\hat{\boldsymbol{\theta}} - \mathbf{K}\mathbf{s} \tag{2.111}$$

with $\mathbf{s} = \dot{\mathbf{q}} - \dot{\mathbf{q}}_r$ and if $\hat{\boldsymbol{\theta}}_u$ is updated according to

$$\dot{\hat{\boldsymbol{\theta}}}_u = -\boldsymbol{\Gamma}\mathbf{Y}^T(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}}_r)\mathbf{s}, \qquad \boldsymbol{\Gamma} > 0 \tag{2.112}$$

then for any $\mathbf{s}(0)$, $\hat{\boldsymbol{\theta}}_u(0)$ and a bounded $\mathbf{q}_d(t)$, there exists a $\mathbf{K} > 0$ such that the position and velocity tracking errors converge to zero. To show this, consider the Lyapunov function candidate

$$\mathcal{V} = \frac{1}{2}[\mathbf{s}^T\mathbf{H}\mathbf{s} + \tilde{\boldsymbol{\theta}}_u^T\boldsymbol{\Gamma}^{-1}\tilde{\boldsymbol{\theta}}_u] \tag{2.113}$$

and compute its time derivative along trajectories of (2.107) as

$$\begin{aligned}\dot{\mathcal{V}} &= \mathbf{s}^T(\mathbf{H}\ddot{\mathbf{q}} - \mathbf{H}\ddot{\mathbf{q}}_r) + \tilde{\boldsymbol{\theta}}_u^T \boldsymbol{\Gamma}^{-1}\dot{\tilde{\boldsymbol{\theta}}}_u + \frac{1}{2}\mathbf{s}^T\dot{\mathbf{H}}\mathbf{s} \\ &= \mathbf{s}^T(\boldsymbol{\tau} - \mathbf{C}\dot{\mathbf{q}}_r - \mathbf{g} - \mathbf{H}\ddot{\mathbf{q}}_r) + \tilde{\boldsymbol{\theta}}_u^T \boldsymbol{\Gamma}^{-1}\dot{\tilde{\boldsymbol{\theta}}}_u\end{aligned}$$

where the fact (property 3 of Section 2.2.4) that $\dot{\mathbf{H}} - 2\mathbf{C}$ is skew symmetric has been used. If control (2.111) and parameter update law (2.112) are employed, then

$$\begin{aligned}\dot{\mathcal{V}} &= -\mathbf{s}^T[\mathbf{K} - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})]\mathbf{s} \\ &= -\mathbf{s}^T[\mathbf{K} - \bar{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}})]\mathbf{s} \qquad (2.114)\end{aligned}$$

where $\bar{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}}) = [\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{C}^T(\mathbf{q}, \dot{\mathbf{q}})]/2$ is a symmetric matrix. By (2.114) in conjunction with the same argument as was adopted in [37] and [117], it can be shown that both $\tilde{\mathbf{q}}$ and $\dot{\tilde{\mathbf{q}}}$ converge to zero.

Implementation Consideration

Concerning the algorithm implementation, note that generating a new control torque vector requires that $\mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}_r)$ be evaluated to update parameter vector $\hat{\boldsymbol{\theta}}_u$ in (2.112) and (2.111) to obtain $\boldsymbol{\tau}$ as (2.111) can be written in the form

$$\boldsymbol{\tau} = \mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}_r) \begin{bmatrix} \boldsymbol{\theta}_k \\ \hat{\boldsymbol{\theta}}_u \end{bmatrix} - \mathbf{K}\mathbf{s} \qquad (2.115)$$

As is noted in [37], if one chooses $\boldsymbol{\Lambda} = \lambda\mathbf{I}$ in (2.108) and $\mathbf{K} = \lambda\hat{\mathbf{H}}(\mathbf{q})$ in (2.111), then the time derivative of $\mathcal{V}$ in (2.113) along trajectories of (2.107)is given by

$$\dot{\mathcal{V}} = -\mathbf{s}^T[\lambda\mathbf{H}(\mathbf{q}) - \bar{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}})]\mathbf{s} \qquad (2.116)$$

provided that the unknown parameters are updated according to

$$\dot{\hat{\boldsymbol{\theta}}}_u = -\boldsymbol{\Gamma}\mathbf{Y}^T(\mathbf{q}, \dot{\mathbf{q}}, (\ddot{\mathbf{q}}_r - \lambda\mathbf{s}))\mathbf{s} \qquad (2.117)$$

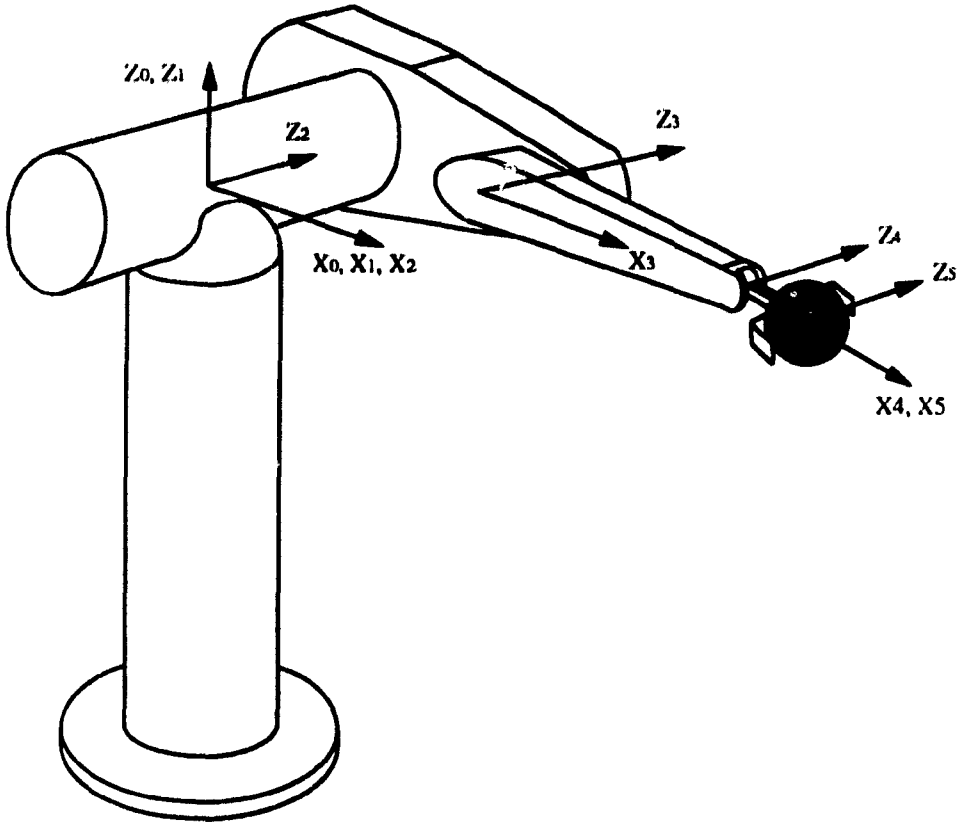


Figure 2.2: A 4 DOF Robot (Identical to First 4 Links of PUMA 560)

Since $\mathbf{H}(\mathbf{q})$ is uniformly positive definite, $\dot{V}$ in (2.116) is negative if λ is sufficiently large. Note that the control torque in this case becomes

$$\boldsymbol{\tau} = \mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}^*) \begin{bmatrix} \boldsymbol{\theta}_k \\ \hat{\boldsymbol{\theta}}_u \end{bmatrix} \tag{2.118}$$

where

$$\ddot{\mathbf{q}}^* = \ddot{\mathbf{q}}_d - 2\lambda\dot{\tilde{\mathbf{q}}} - \lambda^2\tilde{\mathbf{q}} \tag{2.119}$$

Obviously, (2.118) and (2.119) represent a quasi-computed-torque controller which would be identical to the computed-torque algorithm if $\hat{\boldsymbol{\theta}}_u = \boldsymbol{\theta}_u$.

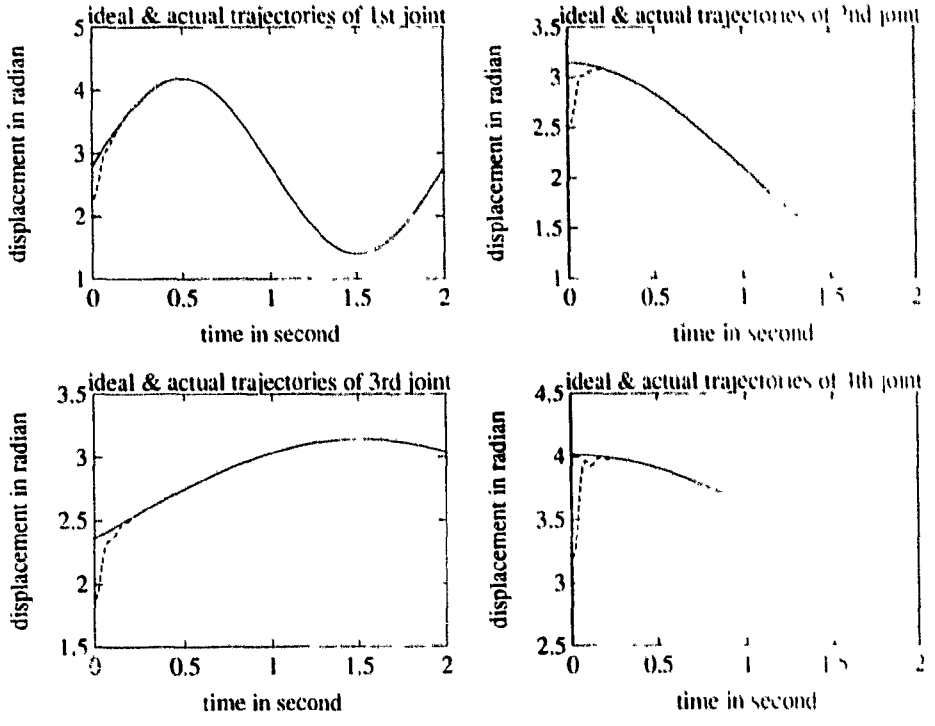


Figure 2.3: Joint trajectory tracking of joint 1, 2, 3, and 4 using modified Slotine-Li adaptive algorithm.

2.3.10 A Case Study

In this section, we present a case study that simulates the two adaptive motion control algorithms where the regressor encountered are evaluated using the recursive regressor dynamics formulation introduced in this Chapter.

Consider a 4 DOF manipulator, shown in Figure 2.2, with a geometry identical to the first 4 links of a PUMA 500 robot. Its dynamic parameters are however simplified as follows: all 4 links are of point-mass with $m_1 = m_2 = m_3 = 2$ Kg and $m_4 = 0.5$ Kg. For the first three links, the mass center of each link is at its geometric center-point, and the mass center of link 4 is at its distal end. The robot

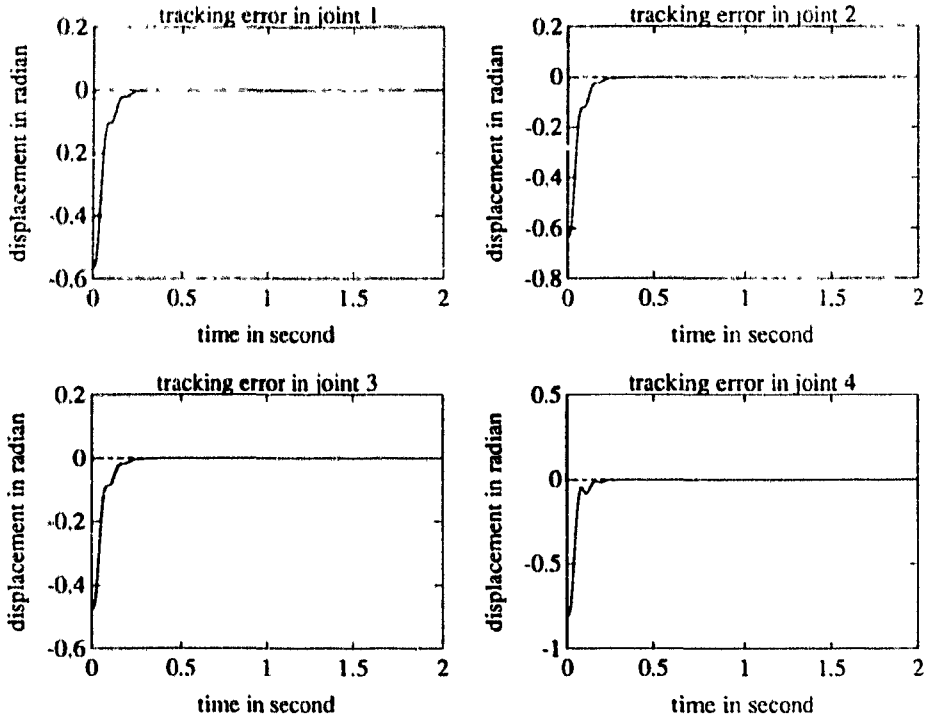


Figure 2.4: Joint trajectory tracking errors of joint 1, 2, 3, and 4 using modified Slotine-Li adaptive algorithm.

carries a 0.5Kg point-mass load, tracking in the joint space a trajectory specified by

$$\begin{bmatrix} q_1(t) \\ q_2(t) \\ q_3(t) \\ q_4(t) \end{bmatrix} = \begin{bmatrix} c_1 + a_1 \sin\omega t \\ c_2 + a_2 \cos(\omega t/2) \\ c_3 + a_3 \sin(\omega t/3) \\ c_4 + a_4 \cos(\omega t/4) \end{bmatrix}$$

with $c_1 = 8\pi/9$, $c_2 = 2\pi/3$, $c_3 = 3\pi/4$, $c_4 = 15\pi/18$, $a_1 = 4\pi/9$, $a_2 = \pi/3$, $a_3 = \pi/4$, and $a_4 = 8\pi/18$ for $0 \leq t \leq 2$ seconds. Under these circumstances, the robot can be treated as if $m_4 = 1$ Kg and carrying no load.

Now *assume* that the user does not know the load mass and makes an initial guess of $m_4 = 0.5$ Kg. The initial robot configuration is set with 20% relative error in joint displacement for each joint. Both the modified Slotine-Li algorithm and

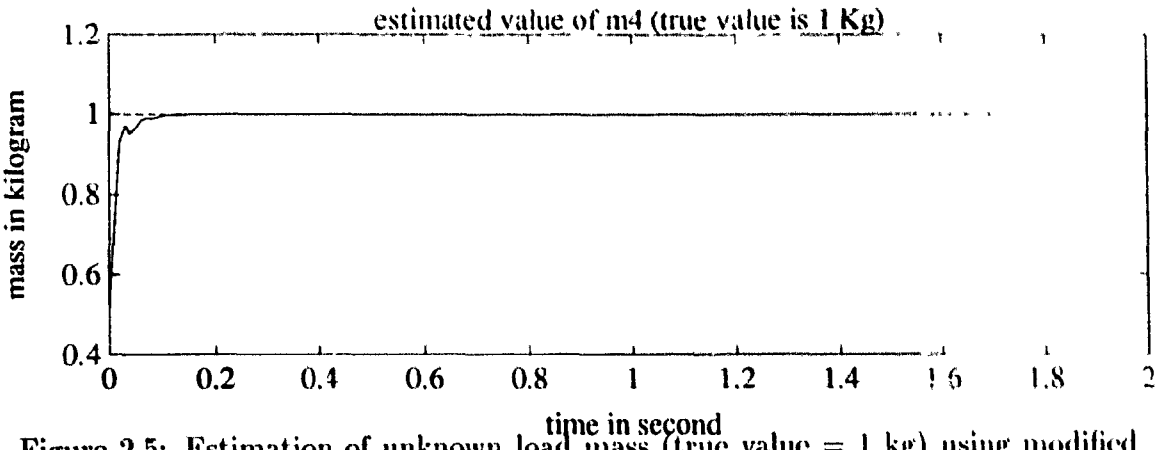


Figure 2.5: Estimation of unknown load mass (true value = 1 kg) using modified Slotine-Li adaptive algorithm

Craig's algorithm are applied to control the robot motion. Figure 2.3 shows the trajectory tracking of joint 1, 2, 3, and 4 with tracking errors depicted in Figure 2.4 and unknown load mass estimation in Figure 2.5 using modified Slotine-Li adaptive control algorithm. Figure 2.6 shows the trajectory tracking of joint 1, 2, 3, and 4 with tracking errors depicted in Figure 2.7 and unknown load mass estimation in Figure 2.8 using Craig's adaptive control algorithm. The regressors $\mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}_r)$ and $\mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, (\ddot{\mathbf{q}}_r - \lambda \mathbf{s}))$ in the modified Slotine-Li algorithm, and the regressor $\mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}^*)$ in Craig's algorithm are evaluated using Algorithm 2.1 proposed in Section 2.3.7. From the figures it is observed that it takes less than 0.3 second for the modified Slotine-Li algorithm and about 0.35 second for Craig's algorithm to see the tracking error converging to an acceptable tolerance. Both algorithms are able to identify the unknown mass of the load but the modified Slotine-Li algorithm can do it quicker. As mentioned earlier, the modified Slotine-Li algorithm is computationally more efficient as compared to Craig's algorithm. In addition, our simulation experience indicates that parameters λ and $\boldsymbol{\Gamma}$ in the modified Slotine-Li algorithm are less sensitive than their counterparts in Craig's algorithm, which is consistent with the observations made in a force control study [39].

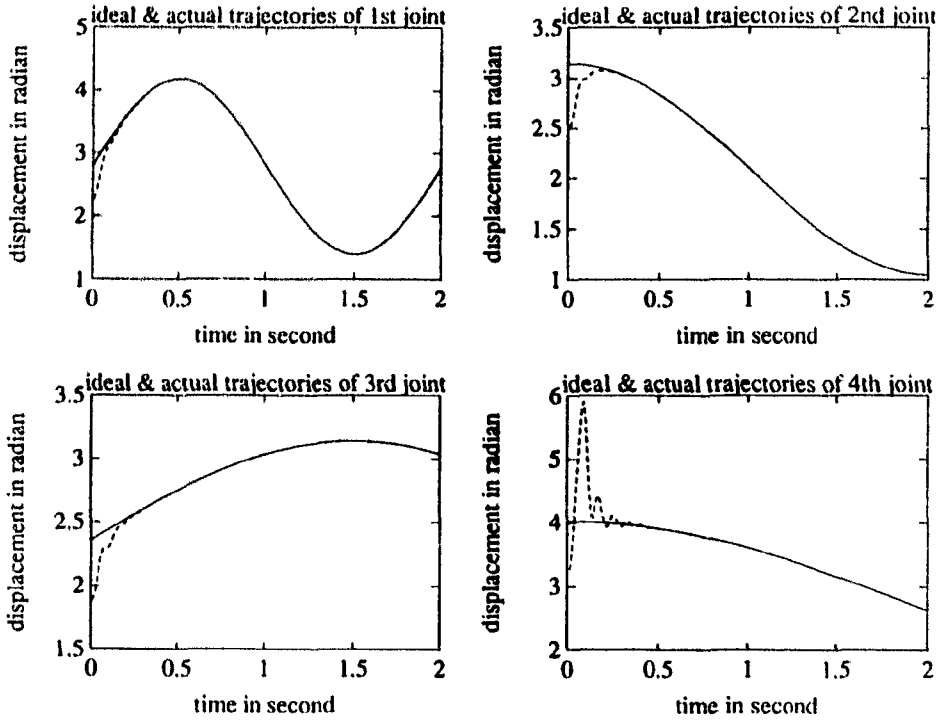


Figure 2.6: Joint trajectory tracking of joint 1, 2, 3, and 4 using Craig's adaptive algorithm

2.4 Real-Time Implementation

The recursive version of regressor dynamics (Algorithm 2.1) has been used to implement model-based adaptive position and force control algorithms in real-time on a PUMA 560 industrial robot. In this section we shall present the experimental results of implementing the modified Slotine-Li adaptive position control on the first three links of a PUMA 560 robot. Issues on computational efficiency, sampling rate selection, and the effect of robot dynamics on the control quality will also be addressed [119].

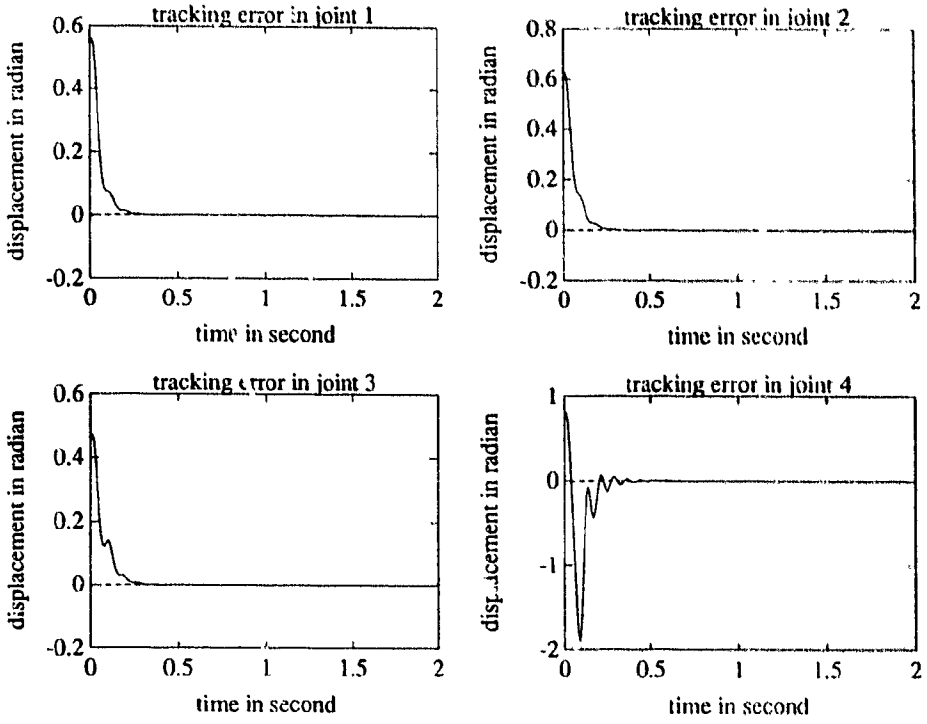


Figure 2.7: Joint trajectory tracking errors of joint 1, 2, 3, and 4 using Craig's adaptive algorithm

2.4.1 The Implementation

As mentioned above, each of the first three joints of a PUMA 560 (similar to Figure 2.2) is controlled to follow a given sinusoidal trajectory. The kinematic and dynamic parameters of the PUMA 560 robot used in the implementation are listed in Table 2.1 (Denavit-Hartenberg parameter table) and Table 2.2 (dynamic parameters) [157, 158, 159].

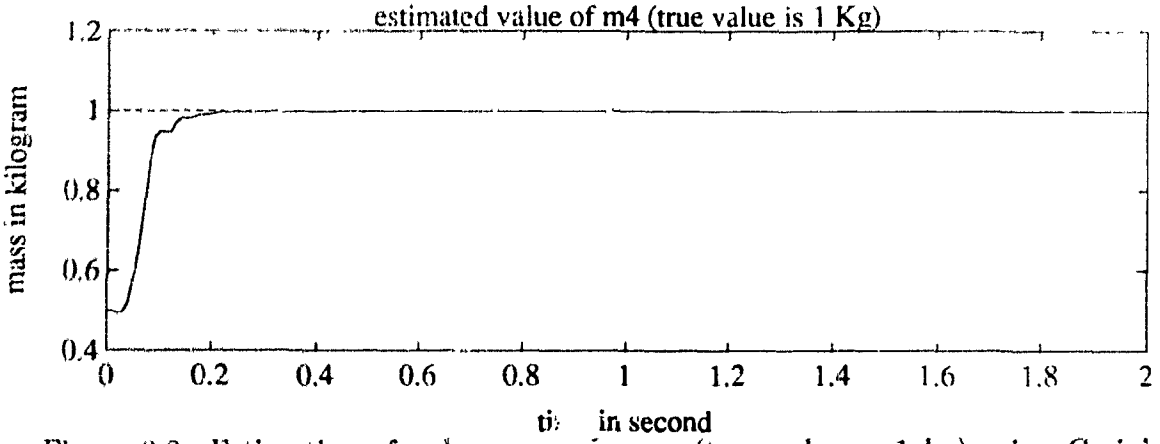


Figure 2.8: Estimation of unknown load mass (true value = 1 kg) using Craig's adaptive algorithm

Table 2.1: Kinematic Parameters of PUMA 560

i	α_{i-1} (degree)	a_{i-1} (meter)	d_i (meter)	θ_i (degree)
1	0	0	0	θ_1
2	-90°	0	0.234	θ_2
3	0	0.432	-0.094	θ_3
4	90°	-0.020	0.433	θ_4
5	-90°	0	0	θ_5
6	90°	0	0	θ_6

and the workspace of the first three joints is described by

$$\begin{aligned} -164^\circ &\leq \theta_1 \leq 162^\circ \\ -216^\circ &\leq \theta_2 \leq 35^\circ \\ -133^\circ &\leq \theta_3 \leq 142^\circ \end{aligned}$$

In the dynamic adaptive position control implementation, the first three joints of the PUMA 560 are controlled to follow the desired trajectories defined by

$$\theta_1(t) = 60\sin(0.628t), \quad \theta_2(t) = 50\sin(0.628t) - 90, \quad \theta_3(t) = 40\sin(0.628t)$$

while the unknown parameter was estimated in real-time.

Table 2.2: Dynamic Parameters of PUMA 560

Link	Mass	Center of Mass			Diagonal Terms of Inertia				Friction
i	m_i	p_x	p_y	p_z	I_{xx}	I_{yy}	I_{zz}	I_{motor}	Coe.
1	-	-	-	-	-	-	0.350	1.140	7.50
2	16.8	0.068	0.006	-0.016	0.130	0.524	0.539	4.710	27.63
3	5.80	0	-0.070	0.014	0.066	0.013	0.086	0.830	4.55
456	1.25	0	0	0.012	0.020	0.020	0.015	0.200	-

The modified Slotine-Li adaptive position controller was implemented on a PUMA 560 robot controlled by a PC 486/33 computer using Borland C^{++} language. Through a set of custom-built parallel interface boards, the computer collects the position data from the joint position sensors. It then computes the control signal and sends the signal through the interface to the power amplifiers that drive the actuators to move the robot. Figure 2.9 and 2.10 show the tracking errors of the 3 joints in degree where the payload changes its weights from 6Kg to 6.42Kg and then to 6.93Kg. The sampling rate used is $f = 200$Hz. It is observed that in all cases the tracking errors for joint 1 and 2 are bounded by 1° after 1 second of motion. The norm of control voltages in these 3 cases are shown in Figure 2.11. Figure 2.12 shows the estimated value of the payload. Note that in the 3 cases the estimated values exhibit fairly large oscillations in a similar manner. This is mainly due to the fact that the last 3 joints are *not* locked during the motion so that the time-varying acceleration of the robot "payload" has contributed to the estimated value of the mass. Nevertheless, it is also noted that in all 3 cases the arithmetical average of the estimates provides a good approximation of the payload weight.

2.4.2 Effect of Dynamics on Control

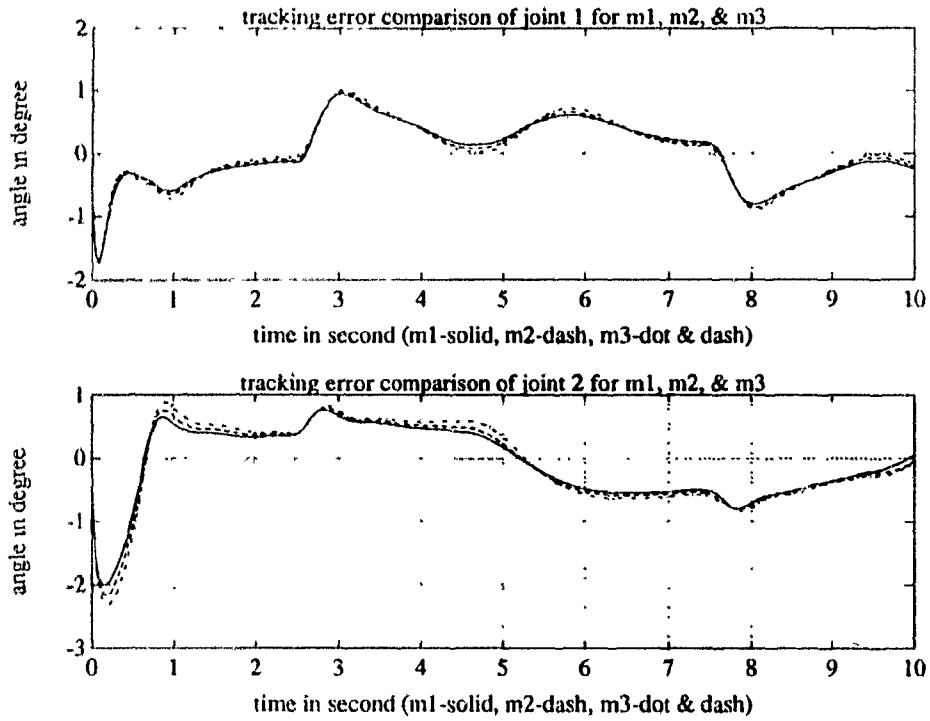


Figure 2.9: Tracking error comparisons of joints 1 and 2. Payload weight is changing from 6 kg to 6.42 kg to 6.93 kg. Sampling rate is 200 Hz.

Next, a payload with mass m =6Kg was used in the test but an incorrect value of m =10Kg was set in the control algorithm as the initial guess of m. The control task was the same as before, and the tracking history using the modified Slotine-Li adaptive control is shown in Figure 2.13. With the same trajectory and same sampling frequency f =200Hz, a fine-tuned PD controller, a computed-torque controller with correct mass value m =6Kg, a computed-torque controller with an incorrect mass value m =10Kg, were separately tested. Together with the results from the adaptive controller, tracking errors of each joint resulted from these 4 controllers are shown in Figure 2.14, 2.15, and 2.16, respectively, where the solid line, the dash line, the dash-dot line, and the "+" line correspond to the PD, the CT with correct m value, the CT with incorrect m value, and the adaptive controller, respectively. It is

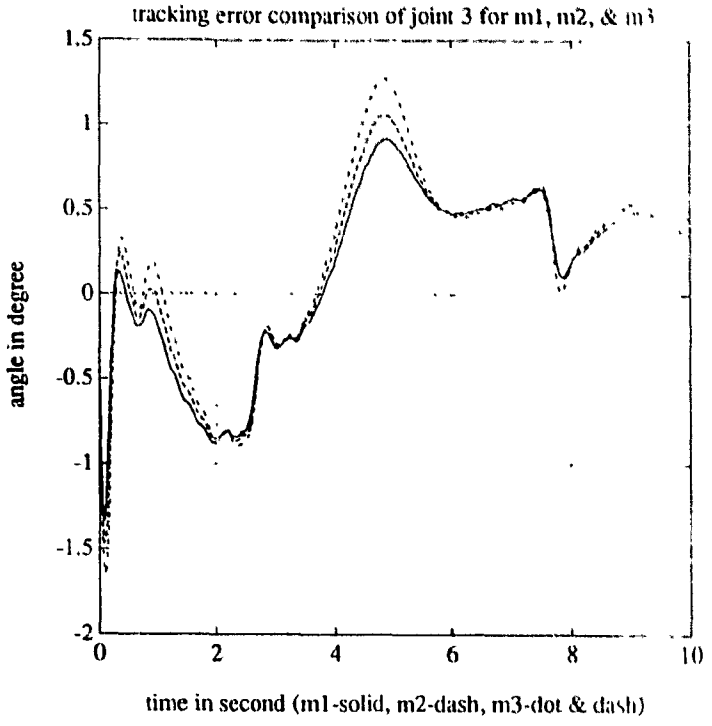


Figure 2.10: Tracking error comparisons of joint 3. Payload weight is changing from 6 kg to 6.42 kg to 6.93 kg. Sampling rate is 200 Hz.

observed that after 2 seconds of motion, the adaptive controller (with an incorrect mass value m) out-performs the best PD and the CT (with the same incorrect mass value m) controllers and is nearly as good as the perfectly tuned CT controller. And all model-based controllers, CT with correct mass value and adaptive controller in this case, out-perform the non-model-based PD controller.

2.4.3 Effect of Sampling Rate on Control

The Northgate PC 486/33 machine was able to compute the control torque within 2.8 ms. In other words, sampling rate can, if necessary, be set as high as 350Hz. This rate was achieved without using any special purpose DSP board. As mentioned earlier, the default sampling rate set for above-reported implementation was

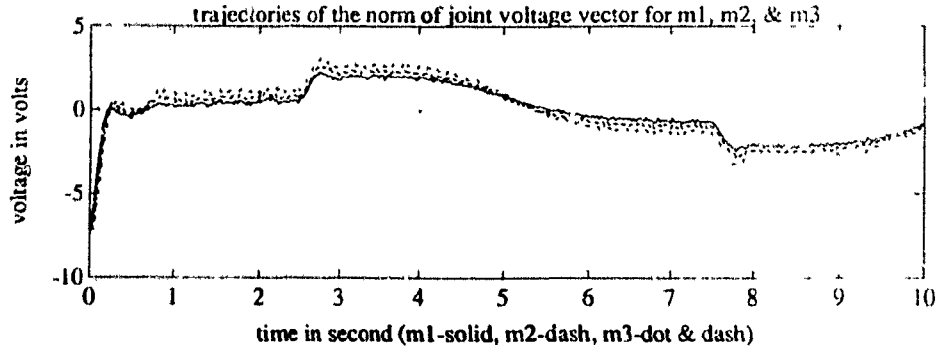


Figure 2.11: Trajectories of the norm of joint control voltage vector for three different payload weights.

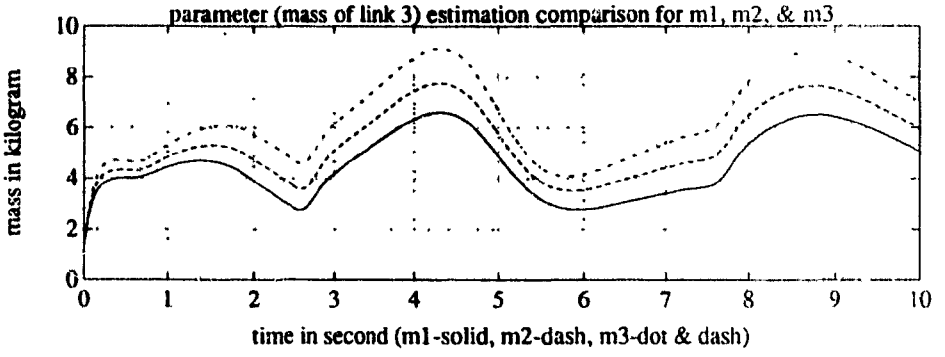


Figure 2.12: Estimation of payload we. for three different payload weights.

f =200Hz, and the system performance appeared to be satisfactory.

Changing the sampling rate from the default value of f_1 =200Hz to f_2 =133Hz and then to f_3 =100Hz. The tracking errors at each joint corresponding to these 3 sampling frequencies are shown in Figure 2.17, 2.18. Trajectories of the norm of joint control voltage vector for the three different frequencies are shown in 2.19. From the figures large tracking errors were observed at joints 2 and 3 when f_3 =100Hz is used. From Figure 2.20, it is noted that the parameter adaptation law did not work well either in the case of sampling frequency=100Hz. This is due to the saturation of the control voltages. By lowering the gain and therefore to reduce saturation, it is found that the controller still works for a sampling frequency as low as 40 Hz.

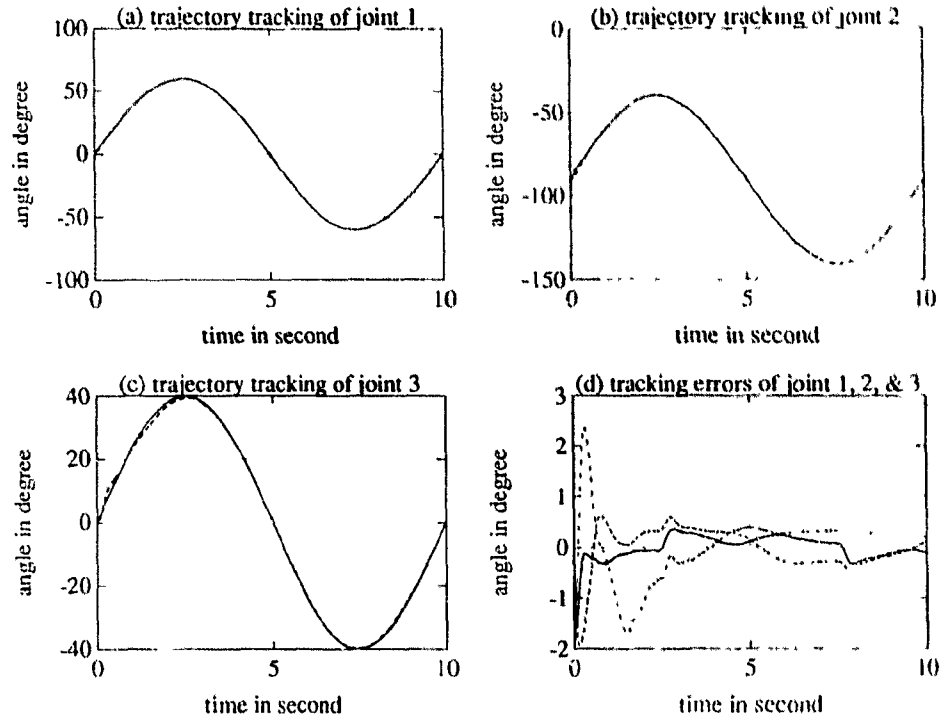


Figure 2.13: Trajectory tracking and tracking errors of the three joints. The payload is 6 kg, but an incorrect initial guess of 10 kg was used.

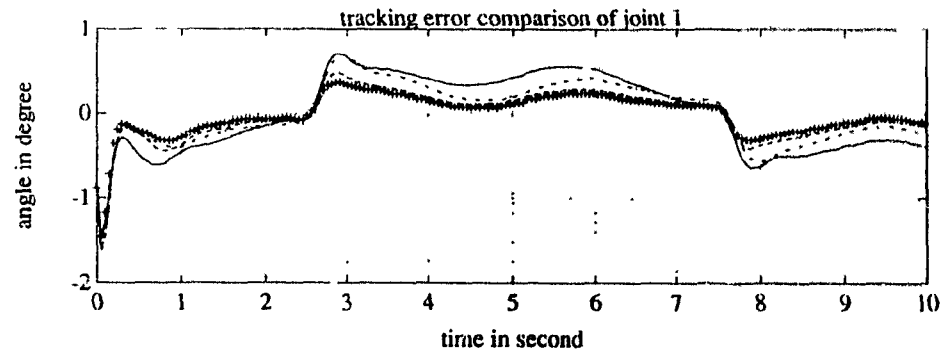


Figure 2.14: Tracking error comparison of joint 1. Solid line - PD control; dashed line - CT control with correct m value; dashed-dot line - CT control with incorrect m value; "+" line - adaptive control.

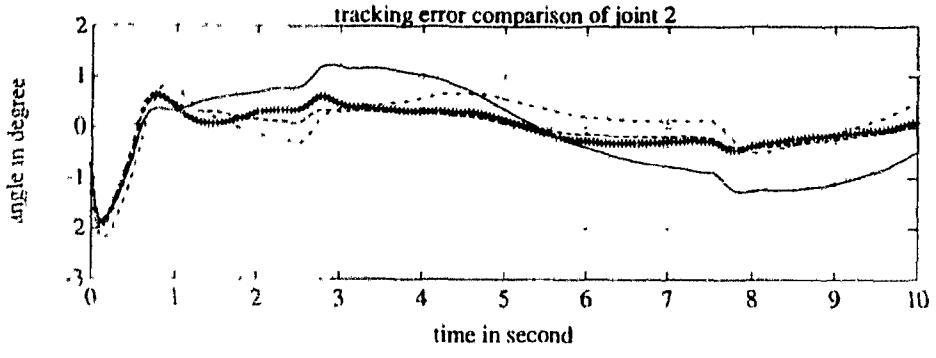


Figure 2.15: Tracking error comparison of joint 2. Solid line - PD control; dashed line - CT control with correct m value; dashed-dot line - CT control with incorrect m value; "+" line - adaptive control.

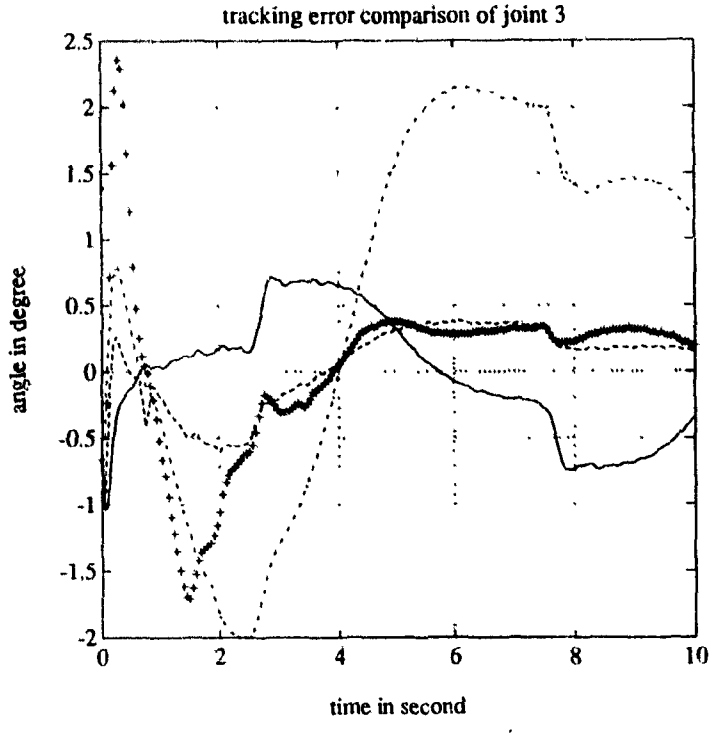


Figure 2.16: Tracking error comparison of joint 3. Solid line - PD control; dashed line - CT control with correct m value; dashed-dot line - CT control with incorrect m value; "+" line - adaptive control.

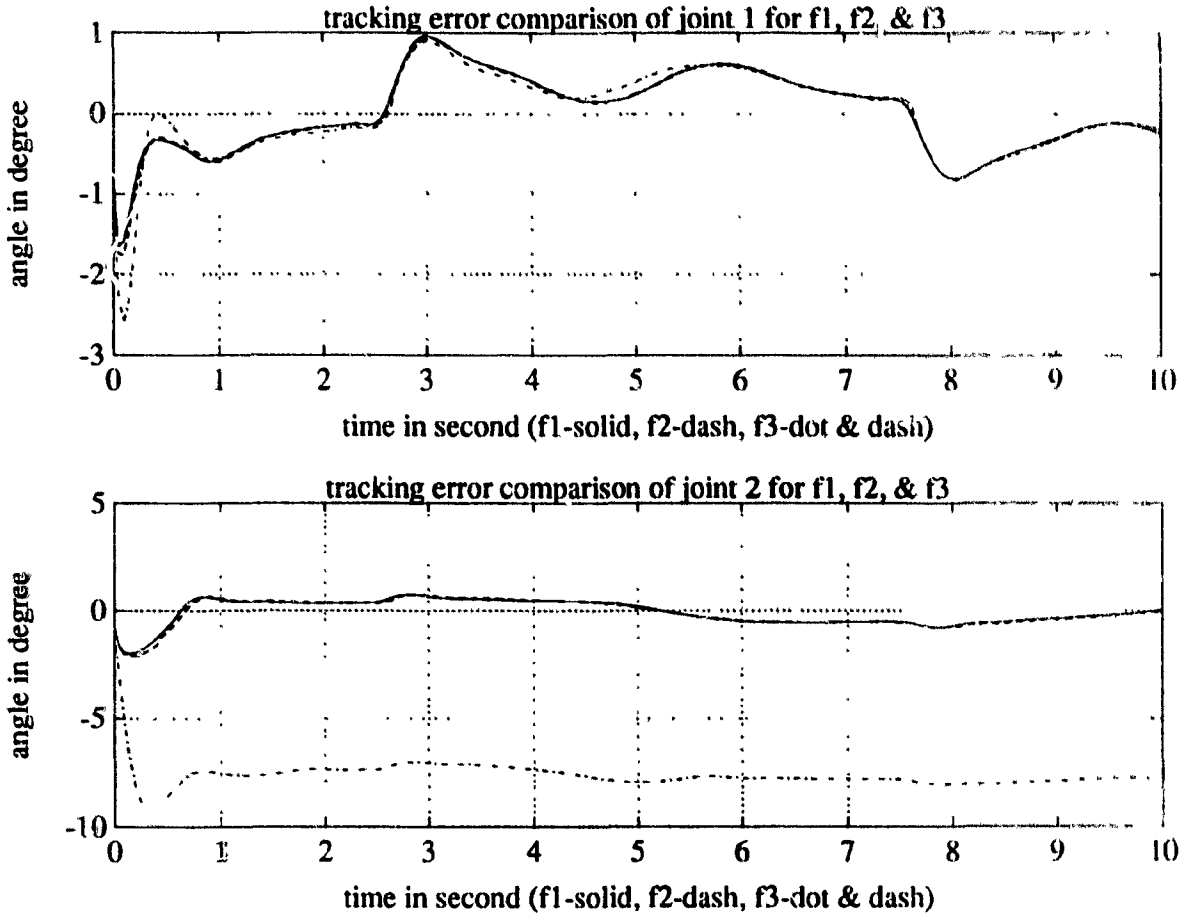


Figure 2.17: Tracking error comparisons of joints 1 and 2. Sampling frequencies are f1=200 Hz (solid line); f2=133 Hz (dashed line); f3=100 Hz (dashed-dot line).

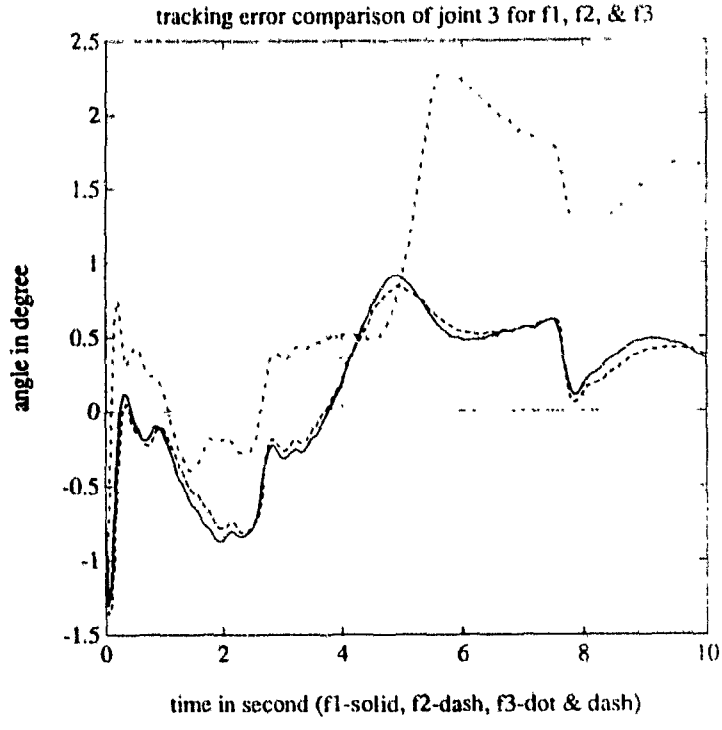


Figure 2.18: Tracking error comparisons of joint 3. Sampling frequencies are f1=200 Hz (solid line); f2=133 Hz (dashed line); f3=100 Hz (dashed-dot line).

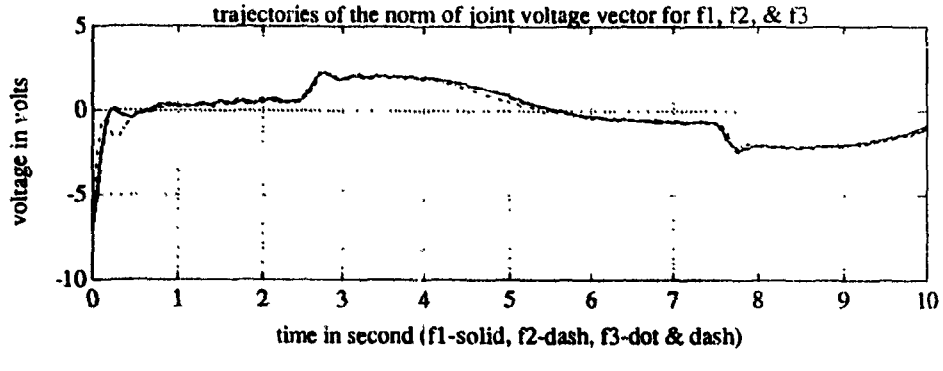


Figure 2.19: Trajectories of the norm of joint control voltage vector for three different frequencies: f1=200 Hz (solid line); f2=133 Hz (dashed line); f3=100 Hz (dashed-dot line).

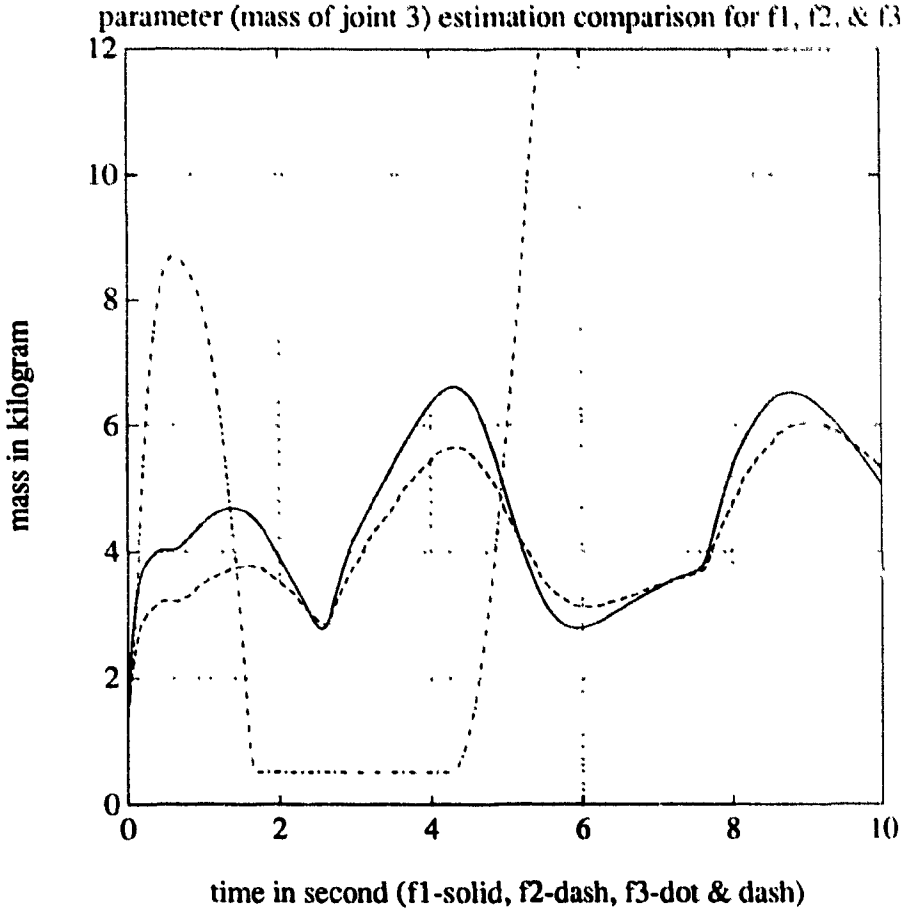


Figure 2.20: Estimation of payload weight for three different frequencies: f1=200 Hz (solid line); f2=133 Hz (dashed line); f3=100 Hz (dashed-dot line).

Chapter 3

Adaptive Impedance Control

3.1 Introduction

Stable execution of contact tasks by mechanical manipulators has been identified as one of the major challenges in robotics [49, 62, 94], and a great deal of researchers' attention has been attracted to the problem during the past decade. Among the control schemes reviewed in Section 1.2, it appears that impedance control of Hogan [60, 63] provides an unified approach to unconstrained motion control, obstacle avoidance, and constrained manipulation. Unlike the hybrid position/force control approach, impedance control does not need to switch back and forth between position control and force control. In addition, impedance control has the flexibility of using [63] or not using [60] robot dynamics in its control algorithm. It has theoretically been shown that the approach may also preserve motion stability during the contact between the end-effector and environment [62]. This is a property especially attractive to the field of contact-force control [49]. Because of these good features, it is worthwhile to seek its extension to those robotic systems whose parameters are not precisely known. This chapter is devoted to the development of two adaptive force control algorithms that are extensions of the conventional impedance control

initiated by N. Hogan [63].

As in the computed torque control for unconstrained motion [4, 6], implementation of Hogan's conventional impedance control [63] requires the use of accurate manipulator dynamics as well as perfect measurement of the external contact forces. On the other hand, however, some parameters of the manipulator such as moments of inertia or the mass center of the end-effector may be uncertain due e.g. to unpredictable payload changes [37]. In addition, measurements from a wrist force sensor are often found noisy. As will be demonstrated in Section 3.7, model uncertainties and imperfection of the force measurements could severely degrade the performance of a model/sensor-based robot controller such as impedance control [63] and, therefore, feasible compensation methods should be sought so as to maintain performance quality as well as motion stability in the presence of parameter uncertainties and measurement noise.

Motivated by the adaptive algorithms of Craig [34] and of Slotine and Li [36] for unconstrained motion (position) control, two adaptive versions of impedance control shall be presented in this chapter. In our treatment it is assumed that some parameters in manipulator dynamics may be uncertain, and the measurements from the wrist force sensor utilized are imprecise. The difference of an adaptive impedance control from an adaptive position control is twofold. First, unlike position control, effort must be made to prevent performance degradation due to imperfection of the force sensor used. Second, here one deals with the dynamic interaction between the robot end-point and its environment that could be very rigid in many applications and, as a reflection of this fact, regulation error must be re-defined in order to accommodate the interactions which do not exist in the free motion case. By introducing the concept of target-impedance reference trajectory (TIRT) which characterizes the desired dynamic relations of the end-point with the environment, it will be shown that the adaptation mechanisms of Craig and Slotine-Li can be

injected into Hogan's conventional impedance control. It will also be shown that the Lyapunov approach utilized in [34, 36] can be refined to properly compensate the measurement noise as long as the magnitude of the noise is assumed to have a known bound. Simulation as well as experimental results of the proposed control algorithms on a PUMA 560 robot are included to illustrate the proposed algorithms.

3.2 Conventional Impedance Control

3.2.1 Robot Dynamic Model in Cartesian Space

Complete dynamics of an n DOF manipulator which has a contact with its high-stiffness environment can be described by

$$\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) + \mathbf{f}(\dot{\mathbf{q}}) = \boldsymbol{\tau} - \mathbf{J}^T\mathbf{F}_{ext} \tag{3.1}$$

where $\mathbf{f}(\dot{\mathbf{q}})$ represent torques due to friction forces, $\mathbf{J}$ is the $n \times 6$ configuration-dependent Jacobian which relates the joint velocity to the linear and angular velocities of the end-effector, and $\mathbf{F}_{ext}$ denotes the force exerted by the end-effector on the environment and measured by a wrist force sensor.

When controlling the dynamic behavior of the end-effector/environment interaction comes to be a main concern, it is often desirable to describe the manipulator dynamics in its operational space. Assuming $n = 6$, the Cartesian space robot dynamics is given by [3]

$$\mathbf{H}_x(\mathbf{x})\ddot{\mathbf{x}} + \mathbf{C}_x(\mathbf{x},\dot{\mathbf{x}})\dot{\mathbf{x}} + \mathbf{g}_x(\mathbf{x}) + \mathbf{f}_x(\dot{\mathbf{x}}) = \mathbf{J}^{-T}\boldsymbol{\tau} - \mathbf{F}_{ext} \tag{3.2}$$

where $\mathbf{x}$ is a 6-dimensional vector representing the position and orientation of the manipulator gripper, and

$$\begin{aligned} \mathbf{H}_x &= \mathbf{J}^{-T}\mathbf{H}\mathbf{J}^{-1} \\ \mathbf{C}_x &= \mathbf{J}^{-T}(\mathbf{C} - \mathbf{H}\mathbf{J}^{-1}\dot{\mathbf{J}})\mathbf{J}^{-1} \end{aligned}$$

$$\mathbf{g}_x = \mathbf{J}^{-T}\mathbf{g}$$
$$\mathbf{f}_x = \mathbf{J}^{-T}\mathbf{f}$$

3.2.2 Two Remarks on Conventional Impedance Control

In the impedance control, target impedance is usually specified by a higher-level supervisory system as a second-order dynamics [63]

$$\mathbf{M}(\ddot{\mathbf{x}} - \ddot{\mathbf{x}}_v) + \mathbf{B}(\dot{\mathbf{x}} - \dot{\mathbf{x}}_v) + \mathbf{K}(\mathbf{x} - \mathbf{x}_v) = -\mathbf{F}_{ext} \tag{3.3}$$

where $\mathbf{x}_v$ is the virtual trajectory that often coincides with the desired trajectory when no contact occurs. However, it would most likely correspond to positions beyond the robot's workspace during the contact in order to maintain a proper amount of contact force. We shall return to this point in Section 3.7 when presenting the simulation results of the proposed control strategy.

Two remarks on the conventional impedance control are now in order.

Remarks

1. Given manipulator dynamics (3.2) and target impedance (3.3), it is quite natural to use a constrained motion control counterpart of the well-known resolved acceleration algorithm [10] to assign a control torque $\boldsymbol{\tau}$ such that the overall system dynamics coincides with that given in (3.3). Obviously, such a torque is given by $\boldsymbol{\tau} = \mathbf{J}^T\mathbf{F}$ with

$$\mathbf{F} = \mathbf{F}_{ext} + \mathbf{C}_x\dot{\mathbf{x}} + \mathbf{g}_x + \mathbf{f}_x + \mathbf{H}_x\{\ddot{\mathbf{x}}_v - \mathbf{M}^{-1}[\mathbf{B}(\dot{\mathbf{x}} - \dot{\mathbf{x}}_v) + \mathbf{K}(\mathbf{x} - \mathbf{x}_v) + \mathbf{F}_{ext}]\} \tag{3.4}$$

i.e., in the notation of [63],

$$\begin{aligned}\boldsymbol{\tau} = \mathbf{J}^T\{&\mathbf{W}^{-1}[\mathbf{M}^{-1}(\mathbf{K}(\mathbf{x}_v - \mathbf{L}(\mathbf{q})) + \mathbf{B}(\dot{\mathbf{x}}_v - \mathbf{J}\dot{\mathbf{q}})) + \ddot{\mathbf{x}}_v \\ &+ \mathbf{J}\mathbf{H}^{-1}(\mathbf{C}\dot{\mathbf{q}} + \mathbf{g} + \mathbf{f}) - \dot{\mathbf{J}}\dot{\mathbf{q}}] + [\mathbf{I} - \mathbf{W}^{-1}\mathbf{M}^{-1}]\mathbf{F}_{ext}\}\end{aligned} \tag{3.5}$$

where L(q) represents the forward kinematics operator which maps a set of joint displacements into the corresponding end-point position/orientation, and $\mathbf{W} = \mathbf{J}\mathbf{H}^{-1}\mathbf{J}^T$ is the mobility tensor whose inverse is the actual inertia of the robot end effector in operational space [60]. The control law (3.5) has been known as impedance control for constrained motion of robots where the major control task is to regulate the dynamic relation of the end-effector with the environment in contact. It follows from above observation that the impedance control is indeed a duality of the resolved acceleration control in the domain of constrained motion control. The structure of the conventional impedance control is shown in Figure 3.1.

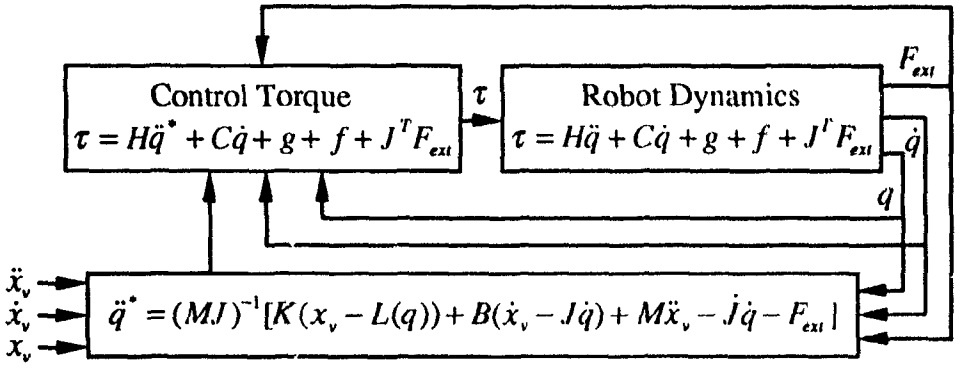


Figure 3.1: Structure of the Conventional Impedance Controller

2. Control law (3.5) can also be used for redundant manipulators where the Jacobian **J** is a full row-rank non-square matrix. Indeed the introduction of mobility tensor **W** in this case avoids performing inverse operation for the Jacobian. On the other hand, however, the control torque given by (3.5) is not computationally efficient when the manipulator involved has six DOF and its Jacobian are nonsingular over the workspace. As a matter of fact, it can readily be shown that the joint torque τ needed can be computed using the

formula

$$\begin{aligned} \boldsymbol{\tau} &= \mathbf{H}\ddot{\mathbf{q}}^* + \mathbf{C}\dot{\mathbf{q}} + \mathbf{g} + \mathbf{f} + \mathbf{J}^T\mathbf{F}_{ext} \\ &= \mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}^*)\boldsymbol{\theta} + \mathbf{J}^T\mathbf{F}_{ext} \end{aligned} \tag{3.6}$$

where $\ddot{\mathbf{q}}^*$ is given by

$$\ddot{\mathbf{q}}^* = (\mathbf{MJ})^{-1}[\mathbf{K}(\mathbf{x}_v - \mathbf{L}(\mathbf{q})) + \mathbf{B}(\dot{\mathbf{x}}_v - \mathbf{J}\dot{\mathbf{q}}) + \mathbf{M}\ddot{\mathbf{x}}_v - \dot{\mathbf{J}}\dot{\mathbf{q}} - \mathbf{F}_{ext}] \tag{3.7}$$

One can therefore use the measurements from joint position/velocity sensors as well as wrist force sensor to compute $\ddot{\mathbf{q}}^*$, and then adopt the recursive Newton-Euler computation scheme [147] or Algorithm 2.1 to calculate the sum of the first four terms on the right-hand side of (3.6).

3.3 Target Impedance Reference Trajectory

Given the desired end-point/environment relation (3.3), a Target Impedance Reference Trajectory (TIRT) is a differentiable vector function $\mathbf{x}_t(t)$ of time, which solves the second-order differential equation (3.3) with a proper set of initial conditions, e.g.

$$\begin{aligned} \mathbf{x}_t(0) &= \mathbf{x}_v(0) \\ \dot{\mathbf{x}}_t(0) &= \dot{\mathbf{x}}_v(0) \end{aligned}$$

and with $\mathbf{F}_{ext}$ replaced by $\hat{\mathbf{F}}_{ext}$, which denotes the measurement of $\mathbf{F}_{ext}$ obtained from the wrist force sensor. Clearly, the TIRT as a function of time coincides with the desired tracking trajectory over the unconstrained motion duration simply because of $\hat{\mathbf{F}}_{ext} = 0$. However, it may differ considerably from the pre-assigned virtual trajectory during the contact. This is particularly true when either the environment involved has a very high stiffness, or the target impedance is assigned such that its dynamics has a pole sufficiently close to the $j\omega$-axis. To be specific,

the TIRT $\mathbf{x}_t(t)$ in the rest of the thesis is defined as the unique solution of the initial-value problem

$$\begin{aligned} \mathbf{M}\ddot{\mathbf{x}}_t + \mathbf{B}\dot{\mathbf{x}}_t + \mathbf{K}\mathbf{x}_t &= -\hat{\mathbf{F}}_{ext} + (\mathbf{M}\ddot{\mathbf{x}}_v + \mathbf{B}\dot{\mathbf{x}}_v + \mathbf{K}\mathbf{x}_v) \\ \mathbf{x}_t(0) &= \mathbf{x}_v(0) \\ \dot{\mathbf{x}}_t(0) &= \dot{\mathbf{x}}_v(0) \end{aligned} \tag{3.8}$$

An example will be given in the simulation study in Section 3.7 to show the difference between the TIRT and virtual trajectory of a robot arm.

A couple of remarks on this concept are now in order.

1. With measurements from the wrist force sensor and the given virtual trajectory, the TIRT can be found very quickly in each sampling duration by numerically integrating equation (3.8). Further notice that such numerical integration is necessary only during the contact since one may otherwise simply take $\mathbf{x}_t(t) = \mathbf{x}_v(t)$.

2. The importance of this concept is due to the fact that the essence of impedance control is to reform the dynamic relation for the end-point/environment interaction such that the end-point could move along the TIRT. The importance of equation (3.8) will become more apparent when an attempt is made to derive an error equation for end-effector's position in the next section.

3.4 Adaptive Impedance Control: Algorithm 3.1

In the adaptive impedance control algorithm proposed below, unknown model parameters are estimated at each control instant in a manner similar to that of [34]. This is made possible by re-defining the regulation error as $\mathbf{x} - \mathbf{x}_t$ where $\mathbf{x}_t$ is the TIRT introduced in Section 3.3.

3.4.1 Impedance control with estimated dynamics

Let $\boldsymbol{\theta}$ and $\hat{\boldsymbol{\theta}} \in R^{r\times 1}$ denote the vector of all unknown parameters in the robot dynamic equation (3.2) and its estimate from an adaptation law to be specified later (see (3.19)), respectively. The impedance control can be implemented if the estimated parameter values are used in (3.4) or (3.6). For the sake of simplicity, it is assumed that the Jacobian is a square and nonsingular matrix over the workspace. The control torque can then be written as

$$\begin{aligned} \boldsymbol{\tau} &= \hat{\mathbf{H}}\ddot{\mathbf{q}}^* + \hat{\mathbf{C}}\dot{\mathbf{q}} + \hat{\mathbf{g}} + \hat{\mathbf{f}} + \mathbf{J}^T\hat{\mathbf{F}}_{ext} + \mathbf{J}^T\mathbf{F}_c \\ &= \mathbf{Y}(\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}}^*)\hat{\boldsymbol{\theta}} + \mathbf{J}^T\hat{\mathbf{F}}_{ext} + \mathbf{J}^T\mathbf{F}_c \end{aligned} \tag{3.9}$$

where $\hat{\mathbf{H}}$, $\hat{\mathbf{C}}$, $\hat{\mathbf{g}}$, and $\hat{\mathbf{f}}$ are evaluated using estimated parameters, $\hat{\mathbf{F}}_{ext}$ is the measurement of $\mathbf{F}_{ext}$, $\mathbf{F}_c$ is a control force to be determined later (see (3.22)) to deal with measurement noise from the force sensor, and

$$\ddot{\mathbf{q}}^* = (\mathbf{MJ})^{-1}[\mathbf{K}(\mathbf{x}_v - \mathbf{L}(\mathbf{q})) + \mathbf{B}(\dot{\mathbf{x}}_v - \mathbf{J}\dot{\mathbf{q}}) + \mathbf{M}\ddot{\mathbf{x}}_v - \dot{\mathbf{J}}\dot{\mathbf{q}} - \hat{\mathbf{F}}_{ext}] \tag{3.10}$$

Similar to the discussion earlier, the recursive Newton-Euler computation scheme [147] or the recursive regressor formula (Algorithm 2.1) applies to (3.9), (3.10), making it a numerically attractive formulation in implementing the proposed algorithm.

In order to perform an error analysis, however, it is more appropriate to write the control torque as $\boldsymbol{\tau} = \mathbf{J}^T\mathbf{F}$ with

$$\begin{aligned} \mathbf{F} = \hat{\mathbf{F}}_{ext} + \hat{\mathbf{C}}_x\dot{\mathbf{x}} + \hat{\mathbf{g}}_x + \hat{\mathbf{f}}_x + \hat{\mathbf{H}}_x\{\ddot{\mathbf{x}}_v - \mathbf{M}^{-1}[\mathbf{B}(\dot{\mathbf{x}} - \dot{\mathbf{x}}_v) \\ + \mathbf{K}(\mathbf{x} - \mathbf{x}_v) + \hat{\mathbf{F}}_{ext}]\} + \mathbf{F}_c \end{aligned} \tag{3.11}$$

3.4.2 The parameter adaptation and determination of $\mathbf{F}_c$

Applying control law (3.11) to dynamic equation (3.2), simple algebra leads to

$$\begin{aligned} &\mathbf{M}(\ddot{\mathbf{x}} - \ddot{\mathbf{x}}_v) + \mathbf{B}(\dot{\mathbf{x}} - \dot{\mathbf{x}}_v) + \mathbf{K}(\mathbf{x} - \mathbf{x}_v) + \hat{\mathbf{F}}_{ext} \\ &\qquad = \mathbf{M}\hat{\mathbf{H}}_x^{-1}[(\tilde{\mathbf{H}}_x\ddot{\mathbf{x}} + \tilde{\mathbf{C}}_x\dot{\mathbf{x}} + \tilde{\mathbf{g}}_x + \tilde{\mathbf{f}}_x) + \mathbf{F}_c + \Delta\mathbf{F}_c] \end{aligned}$$

where

$$\begin{aligned}
\tilde{\mathbf{H}}_x &= \hat{\mathbf{H}}_x - \mathbf{H}_x \\
\tilde{\mathbf{C}}_x &= \hat{\mathbf{C}}_x - \mathbf{C}_x \\
\tilde{\mathbf{g}}_x &= \hat{\mathbf{g}}_x - \mathbf{g}_x \\
\tilde{\mathbf{f}}_x &= \hat{\mathbf{f}}_x - \mathbf{f}_x \\
\Delta\mathbf{F}_e &= \hat{\mathbf{F}}_{ext} - \mathbf{F}_{ext}
\end{aligned}$$

As mentioned before, it is not the virtual trajectory but the TIRT that the end-point tries to follow. Consequently, the regulation error for the constrained motion is defined by $\mathbf{e} = \mathbf{x} - \mathbf{x}_t$, and the error dynamics is obtained by combining the above equation with (3.8) as

$$\ddot{\mathbf{e}} + \mathbf{B}_n\dot{\mathbf{e}} + \mathbf{K}_n\mathbf{e} = \hat{\mathbf{H}}_x^{-1}(\mathbf{Y}_e\tilde{\boldsymbol{\theta}} + \mathbf{F}_e + \Delta\mathbf{F}_e) \triangleq \mathbf{u} \tag{3.12}$$

where $\mathbf{B}_n = \mathbf{M}^{-1}\mathbf{B}$, $\mathbf{K}_n = \mathbf{M}^{-1}\mathbf{K}$, and

$$\mathbf{Y}_e\tilde{\boldsymbol{\theta}} = \tilde{\mathbf{H}}_x\ddot{\mathbf{x}} + \tilde{\mathbf{C}}_x\dot{\mathbf{x}} + \tilde{\mathbf{g}}_x + \tilde{\mathbf{f}}_x \tag{3.13}$$

with

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

and $\mathbf{Y}_e$ the $6 \times r$ regressor matrix depending on $\ddot{\mathbf{x}}$, $\dot{\mathbf{x}}$, $\mathbf{x}$, and the known manipulator parameters. On comparing equation (3.12) with the error equation (11) of [34], the following distinctions can be observed:

- Equation (3.12) is established in operational space rather than in joint space;
- The left-hand side of (3.12) is a reflection of the desired target impedance rather than desired unconstrained motion;

- A friction term has been included in equation (3.13) with the assumption that only viscous friction and dynamic Coulomb friction effects are modeled;
- Sensor noise $\Delta\mathbf{F}_c$ is present and the control force $\mathbf{F}_c$ is included in the right-hand side to suppress $\Delta\mathbf{F}_c$ using a Lyapunov approach as will now be demonstrated.

It follows from [34] and the Appendix B that an augmented error vector can be formed as

$$\mathbf{e}^* = \dot{\mathbf{e}} + \mathbf{\Psi}_c\mathbf{e} \tag{3.14}$$

with $\mathbf{\Psi}_c = \text{diag}(\psi_1, \ldots, \psi_n)$, $\psi_i \in (0, b_{n_i})$, and b_{n_i} the ith diagonal element in $\mathbf{B}_n$ so that the input/output map $\mathbf{u} \longmapsto \mathbf{e}^*$ admits a state-space description given by

$$\begin{aligned} \dot{\mathbf{X}} &= \mathcal{A}\mathbf{X} + \mathcal{B}\mathbf{u} \\ e^* &= \mathcal{C}\mathbf{X} \end{aligned} \tag{3.15}$$

where

$$\begin{aligned} \mathbf{X} &= [e_1\ \dot{e_1}\ \ldots\ e_n\ \dot{e_n}]^T \\ \mathcal{A} &= \text{diag}(\mathbf{A}_1, \ldots, \mathbf{A}_n) \\ \mathcal{B} &= \text{diag}(\mathbf{b}_1, \ldots, \mathbf{b}_n) \\ \mathcal{C} &= \text{diag}(\mathbf{c}_1, \ldots, \mathbf{c}_n) \end{aligned}$$

with

$$\mathbf{A}_i = \begin{bmatrix} 0 & 1 \\ -k_{n_i} & -b_{n_i} \end{bmatrix}$$

$$\mathbf{b}_i = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$\mathbf{c}_i = [\psi_i \quad 1]$$

Note that transfer function matrix of system (3.15) is diagonal with all diagonal elements SPR (see Appendix B). Consequently, there exist

$$\begin{aligned} \mathcal{P} &= \text{diag}(P_1, \ldots, P_n) > 0 \\ \mathcal{Q} &= \text{diag}(Q_1, \ldots, Q_n) > 0 \end{aligned}$$

such that

$$\begin{aligned} \mathcal{A}^T\mathcal{P} + \mathcal{P}\mathcal{A} &= -\mathcal{Q} \\ \mathcal{P}\mathcal{B} &= \mathcal{C}^T \end{aligned} \tag{3.16}$$

A Lyapunov type error analysis begins by defining

$$\mathcal{V}_c(\mathbf{X}, \tilde{\boldsymbol{\theta}}) = \mathbf{X}^T\mathcal{P}\mathbf{X} + \tilde{\boldsymbol{\theta}}^T\boldsymbol{\Gamma}_c^{-1}\tilde{\boldsymbol{\theta}} \tag{3.17}$$

with $\boldsymbol{\Gamma}_c$ positive definite. While the above Lyapunov function has been used in [34] (see Section 1.1.3) for the unconstrained motion control problem, a careful treatment for the imprecise force measurements, which was not the issue in [34], is now needed. By (3.15)-(3.17),

$$\dot{\mathcal{V}}_c = -\mathbf{X}^T\mathbf{Q}\mathbf{X} + 2\tilde{\boldsymbol{\theta}}^T(\mathbf{Y}_c^T\hat{\mathbf{H}}_x^{-1}\mathbf{e}^* + \boldsymbol{\Gamma}_c^{-1}\dot{\hat{\boldsymbol{\theta}}}) + 2\mathbf{e}^{*T}\hat{\mathbf{H}}_x^{-1}(\mathbf{F}_c + \Delta\mathbf{F}_c) \tag{3.18}$$

Thus the parameter adaptation law

$$\dot{\hat{\boldsymbol{\theta}}} = -\boldsymbol{\Gamma}_c\mathbf{Y}_c^T\hat{\mathbf{H}}_x^{-1}\mathbf{e}^* \tag{3.19}$$

leads (3.18) to

$$\dot{\mathcal{V}}_c = -\mathbf{X}^T\mathbf{Q}\mathbf{X} + 2\mathbf{e}^{*T}\hat{\mathbf{H}}_x^{-1}(\mathbf{F}_c + \Delta\mathbf{F}_c) \tag{3.20}$$

If the measurement noise $\Delta\mathbf{F}$ has a known bound, i.e.

$$\|\Delta\mathbf{F}_c\| \leq \delta \tag{3.21}$$

then by choosing

$$\mathbf{F}_c = -\frac{\gamma}{2}\hat{\mathbf{H}}_x\mathbf{e}^* \tag{3.22}$$

where $\gamma \geq 0$ is the gain factor to be determined later, we obtain

$$\dot{\mathcal{V}}_c \leq -\alpha - \gamma\beta^2 + 2\mu\delta \tag{3.23}$$

where

$$\begin{aligned} \alpha &= \mathbf{X}^T\mathbf{Q}\mathbf{X} \\ \beta &= \|\mathbf{e}^*\| \\ \mu &= \|\hat{\mathbf{H}}_x^{-1}\mathbf{e}^*\| \end{aligned}$$

Consequently, the time derivative of $\mathcal{V}_c$ will be strictly negative if

(a) $\mathbf{X} \neq 0$ and $\mathbf{e}^* = 0$;

or

(b) $\mathbf{X} \neq 0$, $\mathbf{e}^* \neq 0$, and

$$\alpha > 2\mu\delta$$

or

(c) $\mathbf{X} \neq 0$, $\mathbf{e}^* \neq 0$, $\alpha \leq 2\mu\delta$, and

$$\gamma > \frac{2\mu\delta - \alpha}{\beta^2} \tag{3.24}$$

If condition (a) or (b) is satisfied, (3.20)-(3.23) indicate that $\mathbf{F}_c$ is not needed i.e. one may assume $\mathbf{F}_c = 0$. Otherwise the term $\mathbf{F}_c$ given by (3.22) with γ satisfying (3.24) must be included in the control torque specified by (3.9) to guarantee a negative $\dot{\mathcal{V}}_c$. In other words, the term $\mathbf{F}_c$ in (3.9) can be specified by (3.22) with γ satisfying

$$\gamma > \max(0, (2\mu\delta - \alpha)/\beta^2) \tag{3.25}$$

Furthermore, since in case (c) μ satisfies $\mu \geq \alpha/2\delta$ and $\mu \leq \|\hat{\mathbf{H}}_x^{-1}\|\beta$, we have $\beta \geq \alpha/(2\delta\|\hat{\mathbf{H}}_x^{-1}\|)$ which implies that the right-hand side of (3.24) has an upper bound

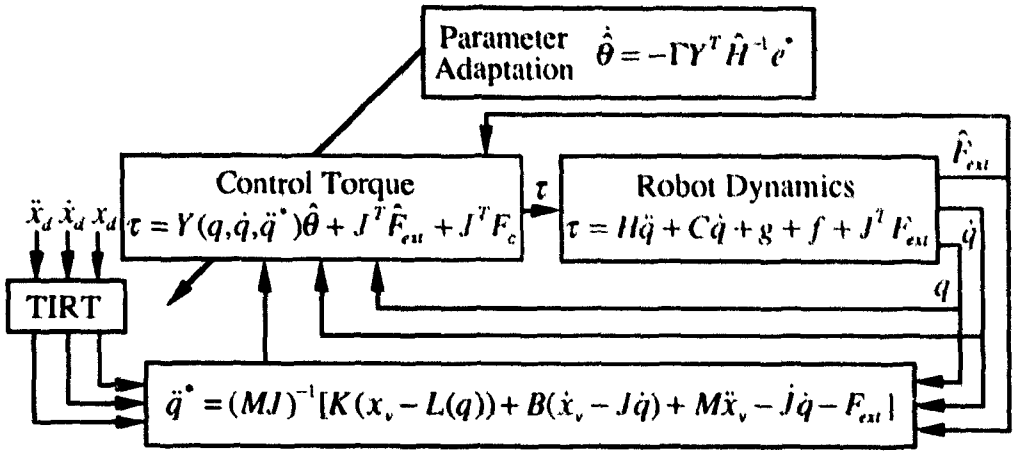


Figure 3.2: Structure of Adaptive Impedance Control Algorithm 3.1

$$\frac{2\mu\delta - \alpha}{\beta^2} \leq \frac{4\delta^2 \| \hat{H}_x^{-1} \|^2 (2\beta\delta \| \hat{H}_x^{-1} \| - \alpha)}{\alpha^2} \equiv \gamma^* \tag{3.26}$$

Obviously, γ^* defined in (3.26) gives an indication of how large the gain factor γ satisfying (3.24) (and hence (3.25)) might be for a given noise bound δ.

In summary, the Algorithm 3.1 can be listed as follows:

Adaptive Impedance Control: Algorithm 3.1

Step 1 Using equation (3.25) to determine γ;

Step 2 Using equation (3.14) to form augmented error vector $\mathbf{e}^*$;

Step 3 Using equation (3.22) to compute $\mathbf{F}_c$;

Step 4 Using Algorithm 2.1 to evaluate the regressor $\mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}^*)$ with $\ddot{\mathbf{q}}^*$ given by (3.10);

Step 5 Using equation (3.9) to compute the control torque $\boldsymbol{\tau}$;

Step 6 Using equation (3.19) to update unknown parameter vector.

The structure of the adaptive control Algorithm 3.1 is shown in Figure 3.2.

The above Algorithm 3.1 has provided an adaptive version of Hogan's impedance control for a robot manipulator with uncertain parameters and an imperfect force sensor.

Remarks

1. In equations (3.2), (3.9) and (3.11) etc., it has been implicitly assumed that no singularities will occur along the planned motion trajectory. Effective approaches to avoiding singular configurations can be found in the literature, see e.g. [160, Chapter 7] and the references therein.

2. As in the unconstrained motion case [34], it follows from (3.10) and (3.19) that implementation of the proposed algorithm requires the invertibility of $\hat{\mathbf{H}}_x$ and the use of joint acceleration. Moreover, (3.19) does not guarantee *a priori* that $\hat{\mathbf{H}}_x^{-1}$ remains bounded [161]. A remedy for this difficulty is to replace (3.19) by a modified adaptive law suggested in [161, Sec. IIIc] in which only a single $\hat{\mathbf{H}}_x^{-1}$ with *fixed* parameters is needed.

3.4.3 A Robustness Property of the Algorithm

There are a number of cases in which some or all components of the control torque $\boldsymbol{\tau}$ calculated from (3.9) exceed the maximum torques that the actuators can provide and, consequently, saturation occurs. In what follows we consider one of such cases where saturation is due to an unusually large force measurement $\hat{\mathbf{F}}_{ext}$. Assume that $\hat{\mathbf{H}}_x^{-1}$ is uniformly bounded, i.e.

$$\|\hat{\mathbf{H}}_x^{-1}\| \leq h^* \tag{3.27}$$

for some constant $h^* > 0$, and that the measurement noise from the force sensor is controlled by a known bound δ as in (3.21). Denoting by $\boldsymbol{\tau}_{sat}$, the saturated

controller output and $\mathbf{F}_{sat} = \mathbf{J}^{-T}\boldsymbol{\tau}_{sat}$, the end-point force during the saturation can be written as

$$\mathbf{F}_{sat} = \mathbf{F} + \Delta\mathbf{F}_s$$

where $\Delta\mathbf{F}_s$ is defined as

$$\Delta\mathbf{F}_s = \mathbf{F}_{sat} - \mathbf{F}$$

with $\mathbf{F}$ obtained from (3.11). Obviously, with this $\mathbf{F}_{sat}$ as the control force, the error dynamics equation (3.12) should be modified as

$$\ddot{\mathbf{e}} + \mathbf{B}_v\dot{\mathbf{e}} + \mathbf{K}_v\mathbf{e} = \hat{\mathbf{H}}_x^{-1}(Y_c\tilde{\boldsymbol{\theta}} + \mathbf{F}_c + \Delta\mathbf{F}_c + \Delta\mathbf{F}_s) \triangleq \mathbf{u}_s$$

which leads to the following modification of (3.23)

$$\dot{\mathcal{V}}_c \leq -\alpha - \gamma\beta^2 + 2\mu(\delta + \|\Delta F_s\|)$$

By (3.27) we have

$$\dot{\mathcal{V}}_c \leq -\alpha - \gamma\beta^2 + 2h^*(\delta + \|\Delta F_s\|)\beta \tag{3.28}$$

It follows that the saturation may lead to a positive $\dot{\mathcal{V}}_c$ and hence an increasing error signal $\mathbf{X}$ (which is equivalent to $\mathbf{e}$ and $\dot{\mathbf{e}}$). On the other hand, since both α and β^2 are proportional to *square* of $\|\mathbf{e}\| + \|\dot{\mathbf{e}}\|$, (3.28) indicates that $\dot{\mathcal{V}}_c$ will become negative again before the error gets too large. In other words, as may be expected, saturation of this type certainly causes a relatively large error as compared to that in an unsaturated control but robust stability of the error dynamics is preserved for tasks in which the contact force may occasionally go up to a level causing controller saturation.

3.5 Adaptive Impedance Control: Algorithm 3.2

Again the TIRT is employed to form the regulation error

$$\mathbf{e} = \mathbf{x} - \mathbf{x}_t \tag{3.29}$$

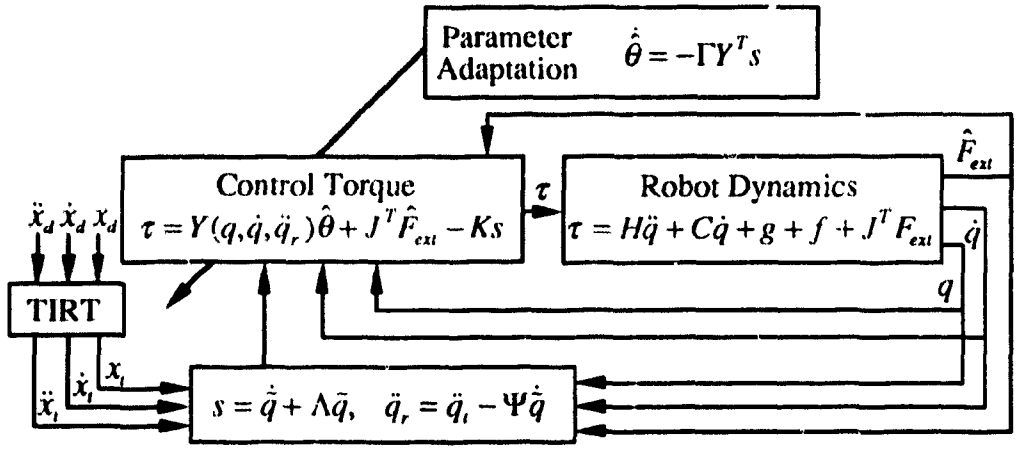


Figure 3.3: Structure of Adaptive Impedance Control Algorithm 3.2

The augmented error vector is defined as

$$\mathbf{s} = \dot{\mathbf{e}} + \boldsymbol{\Psi}_s \mathbf{e} \tag{3.30}$$

where $\boldsymbol{\Psi}_s > 0$, but other constraints that were imposed on $\boldsymbol{\Psi}_c$, its counterpart in Algorithm 3.1 (see (3.14)), are now not needed. Further let $\mathbf{x}_r(t)$ be such that

$$\dot{\mathbf{x}}_r = \dot{\mathbf{x}} - \mathbf{s} \tag{3.31}$$

The second proposed adaptive impedance control is then given by

$$\begin{aligned} \mathbf{F} &= \hat{\mathbf{H}}_x \ddot{\mathbf{x}}_r + \hat{\mathbf{C}}_x \dot{\mathbf{x}} + \hat{\mathbf{g}}_x + \hat{\mathbf{f}}_x + \hat{\mathbf{F}}_{ext} - \mathbf{K}_d \mathbf{s} \\ &= \mathbf{Y}_s(\mathbf{x}, \dot{\mathbf{x}}, \ddot{\mathbf{x}}_r)\hat{\boldsymbol{\theta}} + \hat{\mathbf{F}}_{ext} - \mathbf{K}_d \mathbf{s} \end{aligned} \tag{3.32}$$

where $\mathbf{K}_d > 0$, and $\hat{\mathbf{H}}_x$, $\hat{\mathbf{C}}_x$, $\hat{\mathbf{g}}_x$, and $\hat{\mathbf{f}}_x$ are obtained by substituting the estimated parameters $\hat{\boldsymbol{\theta}}$ into $\mathbf{H}_x$, $\mathbf{C}_x$, $\mathbf{g}_x$, $\mathbf{f}_x$ respectively.

Defining

$$\mathcal{V}_s(\mathbf{s}, \tilde{\boldsymbol{\theta}}) = \frac{1}{2}(\mathbf{s}^T \mathbf{H}_x \mathbf{s} + \tilde{\boldsymbol{\theta}}^T \boldsymbol{\Gamma}_s^{-1} \tilde{\boldsymbol{\theta}})$$

which is an operational space counterpart of the Lyapunov function utilized in [36], (3.2), (3.31), (3.32), and the fact that $\dot{\mathbf{H}}_x - 2\mathbf{C}_x$ is skew-symmetric (see Section

2.2.4) lead to

$$\dot{\mathcal{V}}_s = \tilde{\boldsymbol{\theta}}^T(\mathbf{Y}_s^T\mathbf{s} + \boldsymbol{\Gamma}_s^{-1}\dot{\tilde{\boldsymbol{\theta}}}) - \mathbf{s}^T(\mathbf{K}_d - \mathbf{C}_x)\mathbf{s} + \mathbf{s}^T\Delta\mathbf{F}_e$$

Consequently, if the adaptive law

$$\dot{\tilde{\boldsymbol{\theta}}} = -\boldsymbol{\Gamma}_s\mathbf{Y}_s^T\mathbf{s} \tag{3.33}$$

is used where $\mathbf{Y}_s$ is determined by

$$\mathbf{Y}_s\tilde{\boldsymbol{\theta}} = \tilde{\mathbf{H}}_x\ddot{\mathbf{x}}_r + \tilde{\mathbf{C}}_x\dot{\mathbf{x}}_r + \tilde{\mathbf{g}}_x + \tilde{\mathbf{f}}_x \tag{3.34}$$

then

$$\dot{\mathcal{V}}_s = -\mathbf{s}^T(\mathbf{K}_d - \bar{\mathbf{C}}_x)\mathbf{s} + \mathbf{s}^T\Delta\mathbf{F}_e \le -(\underline{k}_d - \bar{c})\|\mathbf{s}\|^2 + \delta\|\mathbf{s}\| \tag{3.35}$$

where $\underline{k}_d$ denotes the smallest eigenvalue of $\mathbf{K}_d$, $\bar{c}$ denotes the largest eigenvalue of $\bar{\mathbf{C}}_x = [\mathbf{C}_x + \mathbf{C}_x^T]/2$, and (3.21) has been assumed. Unlike the unconstrained motion case [36], (3.35) indicates that an *arbitrarily* chosen $\mathbf{K}_d > 0$ could yield a positive $\dot{\mathcal{V}}_s$ when $\|\mathbf{s}\|$ is sufficiently small, $\|\Delta\mathbf{F}_e\|$ is large, and the inner product $\mathbf{s}^T\Delta\mathbf{F}_e$ is positive. Nevertheless, the following analysis shows that the error signal $\mathbf{s}$ (hence $\mathbf{e}$ and $\dot{\mathbf{e}}$) due to the sensor noise is bounded and the bound can be reduced if more powerful actuators are available.

The boundedness of $\|\mathbf{s}\|$ is an immediate consequence of (3.35): For a fixed $\underline{k}_d$, $\dot{\mathcal{V}}_s < 0$ whenever $\|\mathbf{s}\| > \delta/\underline{k}_d$, and a negative $\dot{\mathcal{V}}_s$ means a reducing $\mathcal{V}_s$ (versus time t) and hence $\mathbf{s}$. The above argument also indicates that a larger $\underline{k}_d$ implies that a smaller bound for $\|\mathbf{s}\|$. By (3.32), it is observed that a large $\underline{k}_d$ (therefore a "large" $\mathbf{K}_d$) may saturate the controller unless the actuators used are sufficiently powerful. Furthermore, (3.30) can be written in an integral form as

$$\mathbf{e}(t) = e^{-\boldsymbol{\Psi}_s(t-t_0)}\mathbf{e}(t_0) + \int_{t_0}^{t} e^{-\boldsymbol{\Psi}_s(t-\nu)}\mathbf{s}(\nu)d\nu$$

which implies that

$$\|\mathbf{e}(t)\| \le \|e^{-\boldsymbol{\Psi}_s(t-t_0)}\mathbf{e}(t_0)\| + \|\mathbf{s}\|\int_{t_0}^{t}\|e^{-\boldsymbol{\Psi}_s(t-\nu)}\|d\nu \le \|\mathbf{e}(t_0)\| + \|\mathbf{s}\|/\underline{\psi}_s$$

where $\underline{\psi}_s$ is the smallest eigenvalue of $\boldsymbol{\Psi}_s$. Therefore, the boundedness of $\|\mathbf{s}\|$ implies the boundedness of $\|\mathbf{e}\|$. Finally, (3.30) gives

$$\|\dot{\mathbf{e}}\| \leq \|\mathbf{s}\| + \|\boldsymbol{\Psi}_s\|\|\mathbf{e}\|$$

i.e. the boundedness of $\|\dot{\mathbf{e}}\|$.

In summary, Algorithm 3.2 can be listed as follows.

Adaptive Impedance Control: Algorithm 3.2

Step 1 Using equation (3.29) to form error $\mathbf{e}$;

Step 2 Using equation (3.30) to form augmented error $\mathbf{s}$;

Step 3 Using equation (3.31) to compute $\mathbf{x}_r$;

Step 4 Using Algorithm 2.1 to evaluate the regressor $\mathbf{Y}$ or $\mathbf{Y}_s$;

Step 5 Using equation (3.36) to compute control torque $\boldsymbol{\tau}$ or using equation (3.32) to compute the control force $\mathbf{F}$;

Step 6 Using equation (3.33) to update unknown parameter vector.

The above Algorithm 3.2 provides a natural combination of Hogan's impedance control with Slotine-Li's parameter adaptation. Figure 3.3 shows the structure of the adaptive control Algorithm 3.2. As is seen from the above argument, robust stability of the error dynamics has been preserved, and the error $\mathbf{s}$ (as well as $\mathbf{e}$ and $\dot{\mathbf{e}}$) may be further reduced by increasing gain $\mathbf{K}_d$ as long as the actuators are sufficiently powerful. Finally, a Lyapunov analysis similar to that used in Section 3.4.3 may be applied to conclude that when the controller is saturated due to a large external force/torque, or a sensor failure, the control error will likely increase but remain bounded.

3.6 Comparison of the Two Algorithms

As implementation feasibility of the algorithms is concerned, note that by (3.29)-(3.31)

$$\ddot{\mathbf{x}}_r = \ddot{\mathbf{x}}_t - \Psi_s \dot{\mathbf{e}}$$

namely Algorithm 3.2 does not require the acceleration of end-effector, $\ddot{\mathbf{x}}$, for computing $\mathbf{F}$ in (3.32) and $\mathbf{Y}_s$ in (3.34). Moreover, Algorithm 3.2 does not need the inversion of $\hat{\mathbf{H}}_x$ in updating parameter vector $\hat{\boldsymbol{\theta}}$ as is seen from (3.33). On the contrary, $\ddot{\mathbf{x}}$ and $\hat{\mathbf{H}}_x^{-1}$ are required in implementing Algorithm 3.1 as was mentioned in Remark 2 (Section 3.4.2). For those cases where the parameters to be identified are time-invariant or slowly time-varying, however, the rate for updating $\hat{\boldsymbol{\theta}}$ may be chosen much lower than the control rate so that the implementation complexity of Algorithm 3.1 due to the evaluation of $\ddot{\mathbf{x}}$ and $\hat{\mathbf{H}}_x^{-1}$ may be substantially reduced.

To compare computation efficiency of the algorithms, we first note that unlike Algorithm 3.1, the recursive Newton-Euler computation scheme does not directly apply to equation (3.32) of Algorithm 3.2. It is, however, possible to reform equation (3.32) as

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

where

$$\boldsymbol{\tau} = \mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}_r)\hat{\boldsymbol{\theta}} - \mathbf{J}^T\mathbf{K}_d\mathbf{s} + \mathbf{J}^T\hat{\mathbf{F}}_{ext} \tag{3.36}$$

where the first term can be evaluated using the regressor dynamics similar to (3.9) and what we have done in Section 2.3.9. This is also the approach adopted in simulations as well as implementations of the algorithms on a PUMA 560 robot.

3.7 A Simulation Study

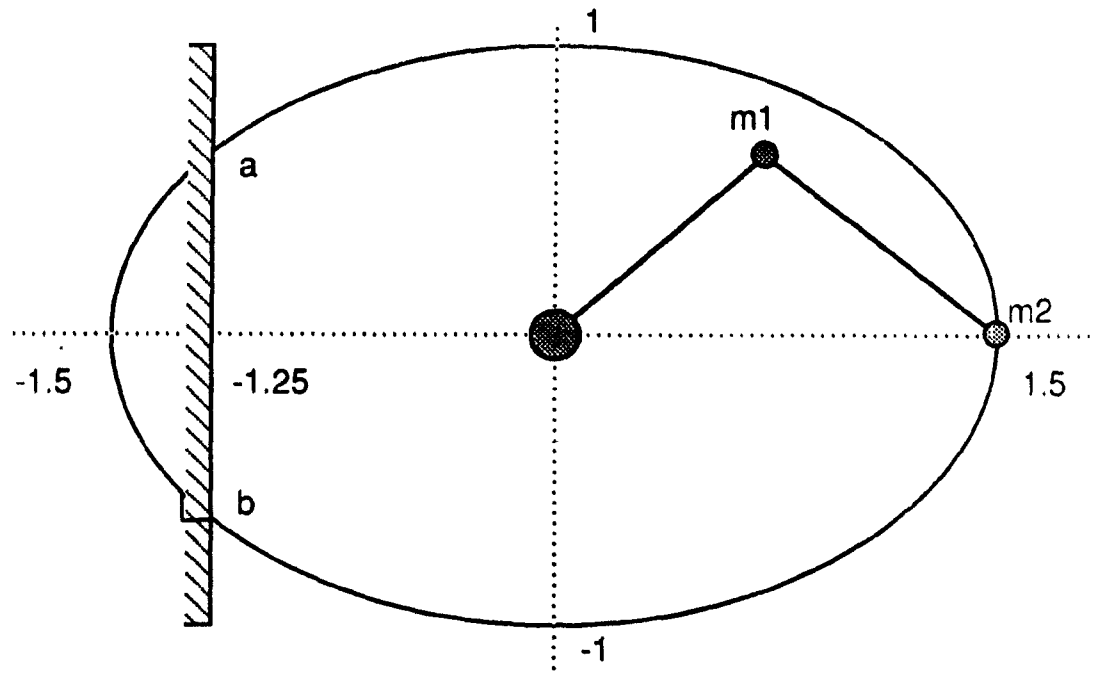


Figure 3.4: The robot manipulator and assigned virtual trajectory

A 2 DOF vertical planar robot was used in the simulation to verify the proposed algorithms. As shown in Figure 3.4, the robot motion begins at point (1.5, 0) moving along an elliptic trajectory until the end-point of the robot hits a wall at point **a**. The end-point is then required to be in proper contact with the wall while sliding down to point **b** along the surface of the wall. The end-point then leaves the wall moving back to point (1.5, 0) along the elliptic trajectory. It is assumed that the mass of each link is a point mass located at the distal end of the link. The true mass values are $m_1 = 2$ kg, $m_2 = 1$ kg, where m_2 represents 0.5 kg link mass plus a 0.5 kg payload, and the link lengths are $l_1 = l_2 = 1m$. The major portion of the virtual trajectory $x_v(t) = [x_{1v}(t) \;\; x_{2v}(t)]^T$ is an ellipse described by

$$\begin{aligned} x_{1v}(t) &= 1.5\cos\pi t \\ x_{2v}(t) &= \sin\pi t \end{aligned}$$

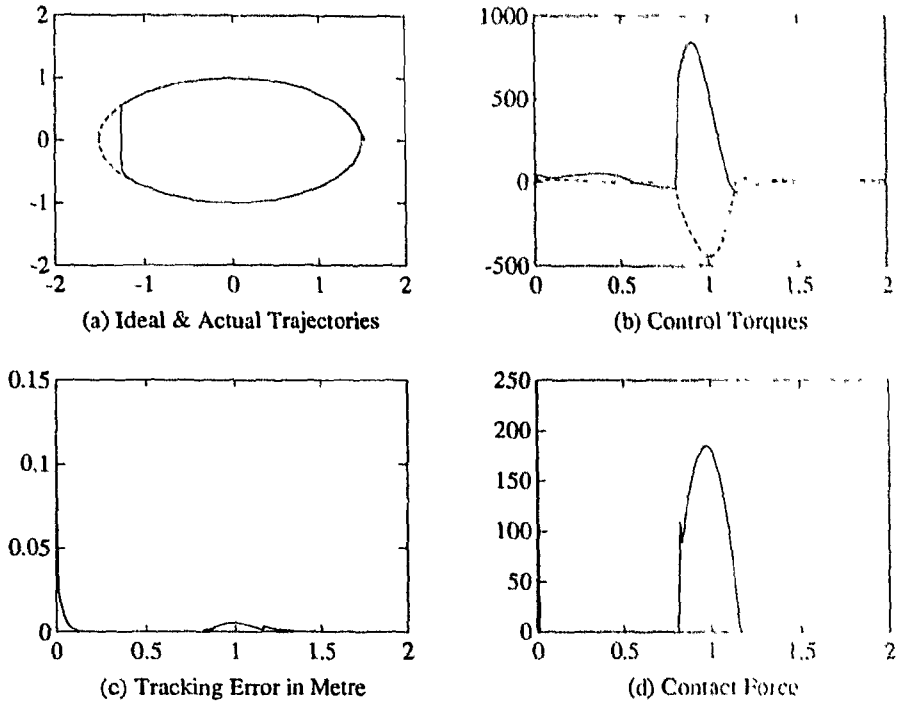


Figure 3.5: Conventional impedance control with true value of m_2 ($m_2 = 1$ kg). (a) Ideal and actual trajectories. (b) Control torques. (c) Tracking error in meters. (d) Contact forces.

for $0 \leq t \leq 2$ second, but as is shown in Figure 3.4, there is a small straight segment replacing the elliptic arc in order to maintain proper contact force.

Assuming that all parameters of the robot manipulator are precisely known and a perfect force sensor is used, Figure 3.5 shows the simulation results when the conventional impedance control (3.5) is employed, where the parameters in target impedance characterization (3.3) are chosen as $B = 2\sqrt{MK}$ with $M = 2$, $K = 4000$, and the environment is assumed to have a high stiffness $K_f = 10^5$. Figure 3.6 shows the TIRT i.e. $x_t(t)$ (dash line) and virtual trajectory $x_v(t)$ (solid line) of the robot. It is seen that $x_t(t)$ and $x_v(t)$ are nearly identical until a large external force

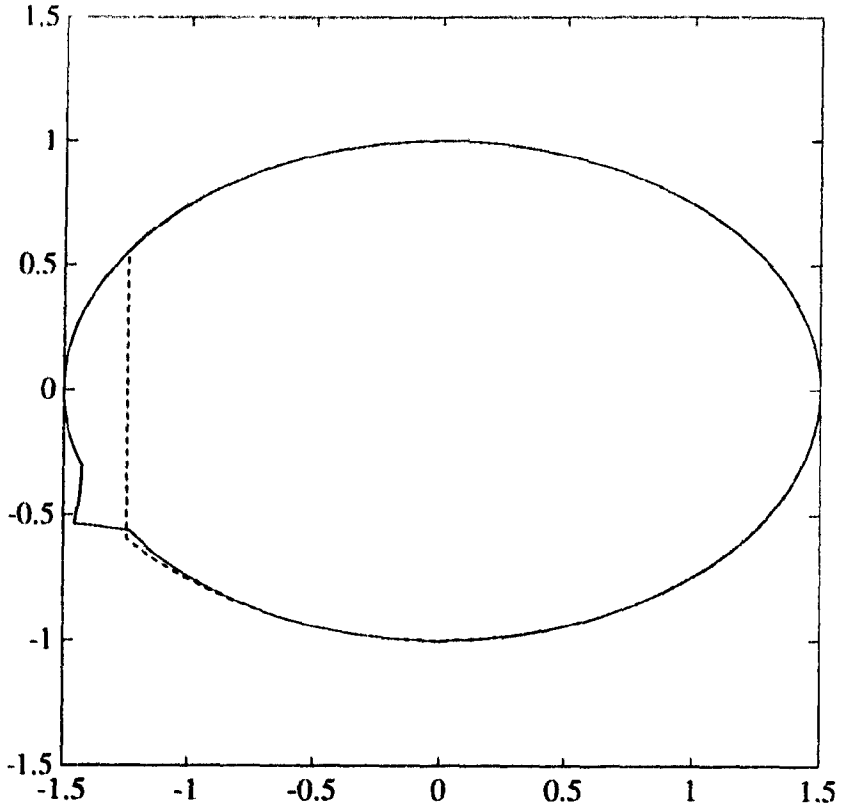


Figure 3.6: The TIRT versus virtual trajectory of the robot.

applies to the gripper when it hits the wall. Next the simulation considers using the conventional impedance control algorithm (3.5) with parameter m_2 replaced by the incorrect value 2.5 kg. As is shown in Figure 3.7, the robot motion then exhibits an unacceptable performance degradation.

The target impedance in the adaptive control case is set as $B = 340$, $M = 2$, and $K = 4000$. For the comparison purpose, m_2 is treated as the only unknown parameter in the manipulator's model and an initial guess of $m_2 = 2.5$ kg is made for the two proposed algorithms. In addition, $\hat{\mathbf{F}}_{ext}$ in the simulation is assumed to be $\mathbf{F}_{ext}$ plus a normally distributed random noise whose maximum magnitude is about 15% of the peak value of the contact force. By applying Algorithm 3.1 and assuming no controller saturation, the actual (solid line) and the pre-assigned virtual trajectory (dash line), the estimated values of m_2, the tracking error, and the

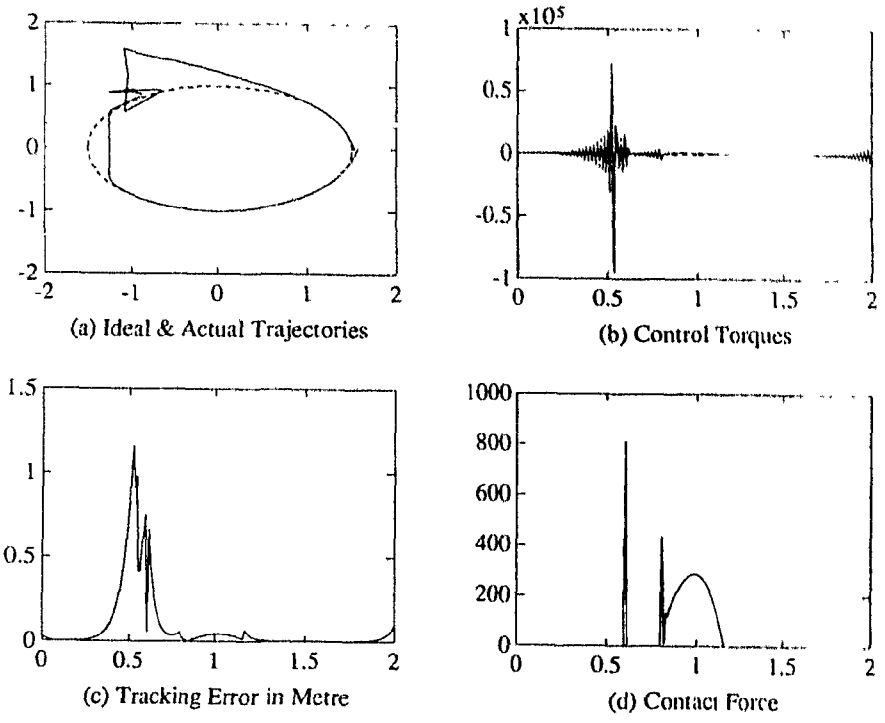


Figure 3.7: Conventional impedance control with $m_2 = 2.5$ kg). (a) Ideal and actual trajectories. (b) Control torques. (c) Tracking error in meters. (d) Contact forces.

contact force are shown in Figure 3.8. Next it is assumed that the actuators can only provide 75% of the maximum torque required in the preceding simulation. As shown in Figure 3.9, in such a case the same algorithm leads to larger tracking errors and larger contact force errors, but the overall performance remains acceptable. Finally, Algorithm 3.2 is applied and the results are shown in Figure 3.10. It is found that the control torques required by Algorithm 3.2 to perform the same task is considerably less than that needed in Algorithm 3.1. Consequently, the controller will not be saturated even if only 75% of the maximum torques required by Algorithm 3.1 is available. It is noted that the tracking errors remain very small during the free

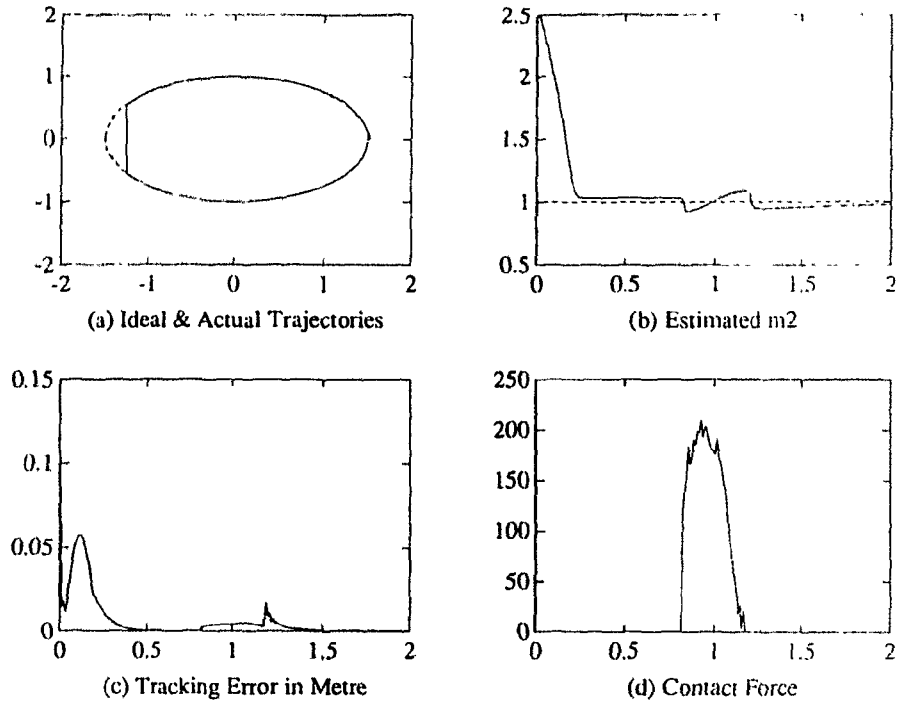


Figure 3.8: Adaptive impedance control using Algorithm 3.1 without saturation. (a) Ideal and actual trajectories. (b) Estimated m_2. (c) Tracking error in meters. (d) Contact forces.

motion period but both tracking errors and contact force errors are slightly larger than that achieved by Algorithm 3.1 in the non-saturation case. Also note that the estimated value of m_2 in all cases quickly approaches to its true value 1 kg during the first free-motion duration. It is then followed by some fluctuation with a less than 20% maximum deviation during the contact and a recovery as soon as the end-effector leaves from the wall at point **b**.

As discussed in Section 3.5, Algorithm 3.2 proposed there appears to be computationally more efficient as compared with Algorithm 3.1. In addition, from a

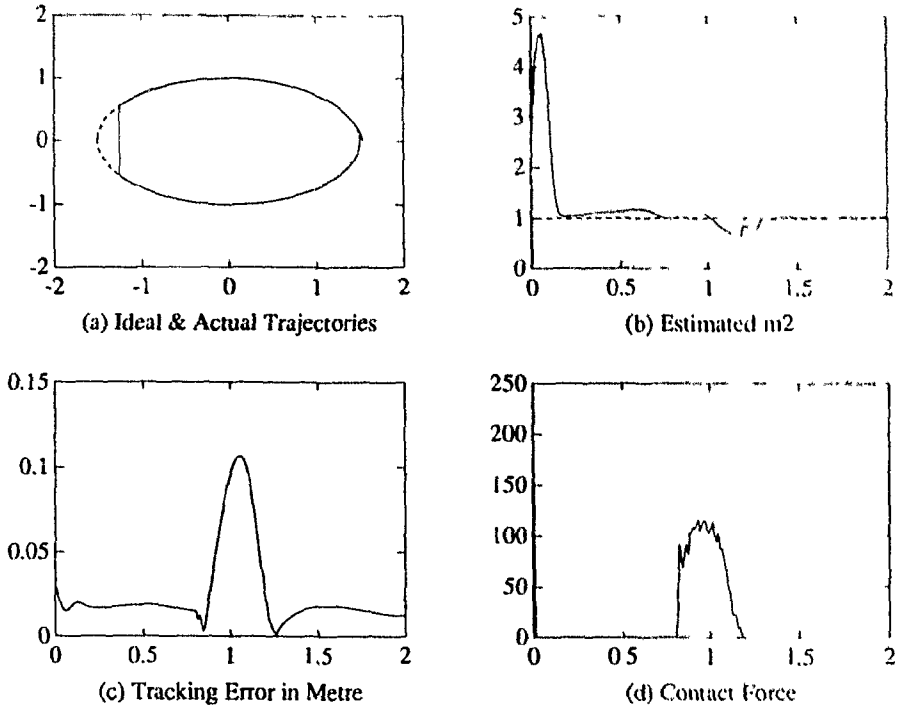


Figure 3.9: Adaptive impedance control using Algorithm 3.1 with saturation. (a) Ideal and actual trajectories. (b) Estimated m_2. (c) Tracking error in meters. (d) Contact forces.

number of computer simulations carried out so far, it is found that parameters $\boldsymbol{\Psi}_s$ and $\boldsymbol{\Gamma}_s$ in Algorithm 3.2 are less sensitive than their counterparts $\boldsymbol{\Psi}_v$ and $\boldsymbol{\Gamma}_v$ in Algorithm 3.1 in order to complete a stable execution of a contact task.

3.8 Real-Time Implementation

The proposed adaptive force control Algorithm 3.1 has been implemented on a PUMA 560 robot. By fixing joints 1, 4, and 6, the PUMA robot is treated as a three-link vertical planar robot using joints 2, 3, and 5, moving in x-y plane,

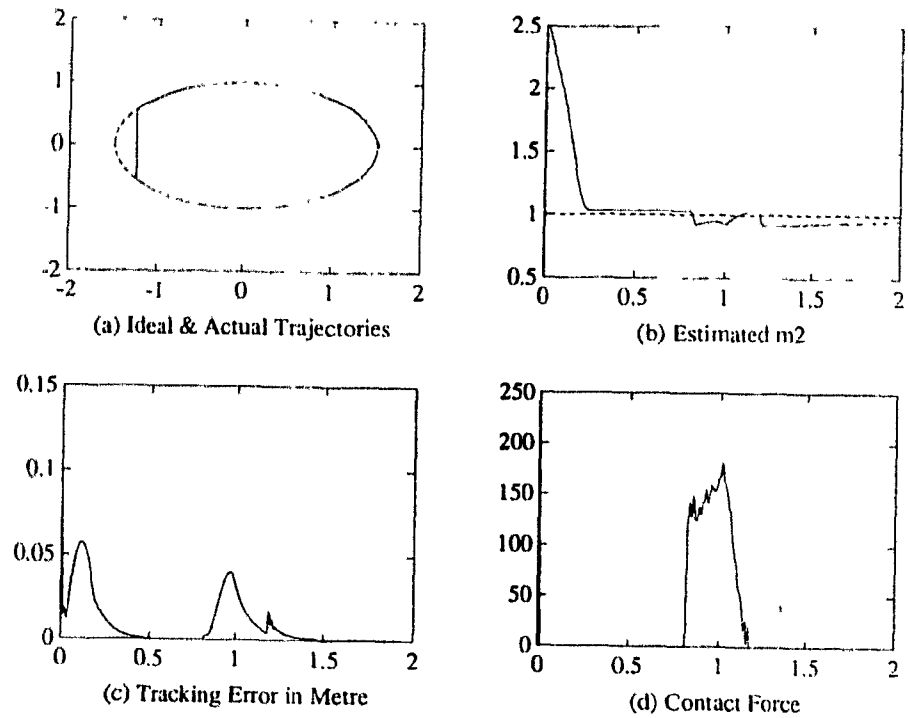


Figure 3.10: Adaptive impedance control using Algorithm 3.2. (a) Ideal and actual trajectories. (b) Estimated m_2. (c) Tracking error in meters. (d) Contact forces.

as shown in Figure 3.11. The kinematic and dynamic parameters of the PUMA 560 robot used in the implementation are listed in Table 3.1 (Denavit-Hartenberg parameter table) and Table 3.2 (dynamic parameters).

A wrist force sensor F/T 30/100 from Assurance Technologies was mounted on the tool flange of the PUMA robot as illustrated in Figure 3.11 to detect the contact force/torque between the robot tip and a hard surface. The computer used in this implementation is a Northgate Elegance 433i 486 computer running at 33Hz. The control and communication source codes necessary for the implementation are written in Borland C++. The position signals from the joint sensors are transferred

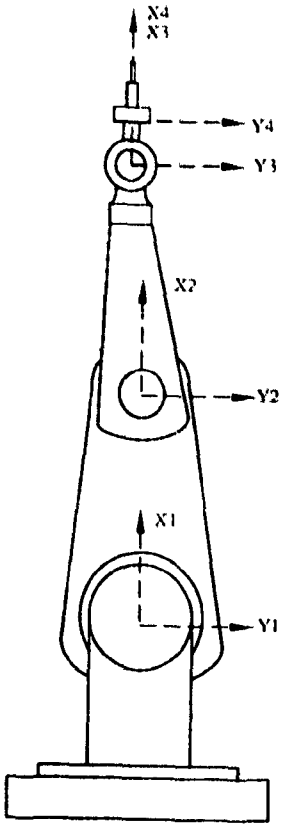


Figure 3.11: PUMA 560 robot (using joints 2, 3, and 5 as a vertical planar robot).

to the computer through a custom-built parallel interface [162, 163] which bypasses the original PUMA controller (Mark II) and the control signals computed by the proposed algorithm are transferred through the same interface to the original PUMA power amplifier unit to drive the robot joint actuators. Signals from the wrist force sensor are acquired through a serial port at a frequency of nearly 200Hz. Accordingly, the control signals are updated at a frequency of 200Hz. By using the recursive regressor dynamics formula (Algorithm 2.1), it takes less than 3ms for the computer to compute the control signal that includes the evaluation of full dynamics for the chosen three joints of the PUMA. It is the force sensor connected to a serial port that slows down the overall updating frequency from about 400Hz to 200Hz.

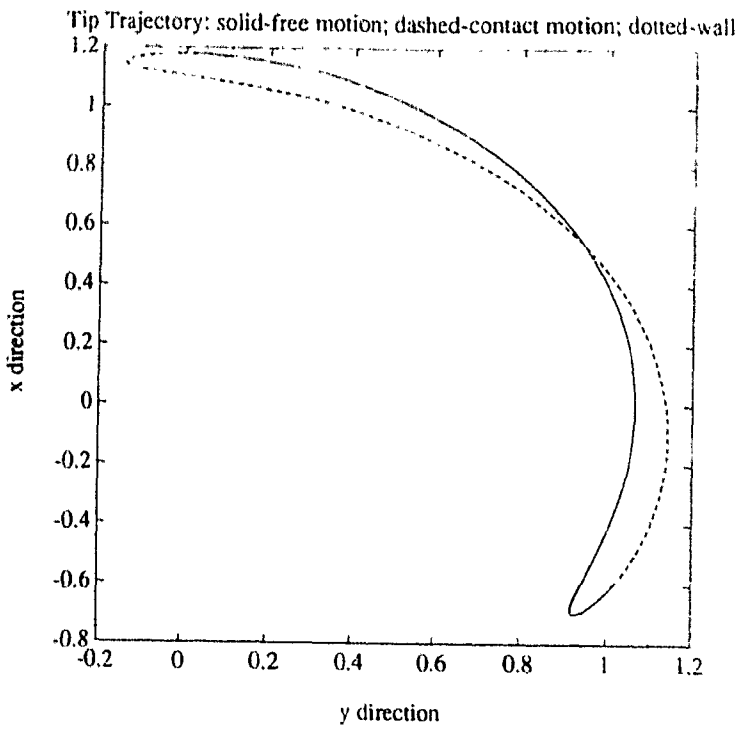


Figure 3.12: Desired tip trajectory in meters.

In this regard, it is expected that the use of a parallel interface between the force sensor and the computer will allow the user to considerably increase the updating frequency, therefore improve overall control quality and even to control more robot joints.

The desired tip trajectory in Cartesian space is shown in Figure 3.12. The motion starts from point (1.2, 0) and goes along the trajectory represented by the solid line to approach point (-0.6, 1). Then the robot tip moves from that point back to (1.2, 0) along the part of the trajectory represented by the dashed line. During this part of the motion, a small portion of the trajectory is blocked by a hard wall represented by the dotted line where the robot tip comes into contact with the wall, slides along

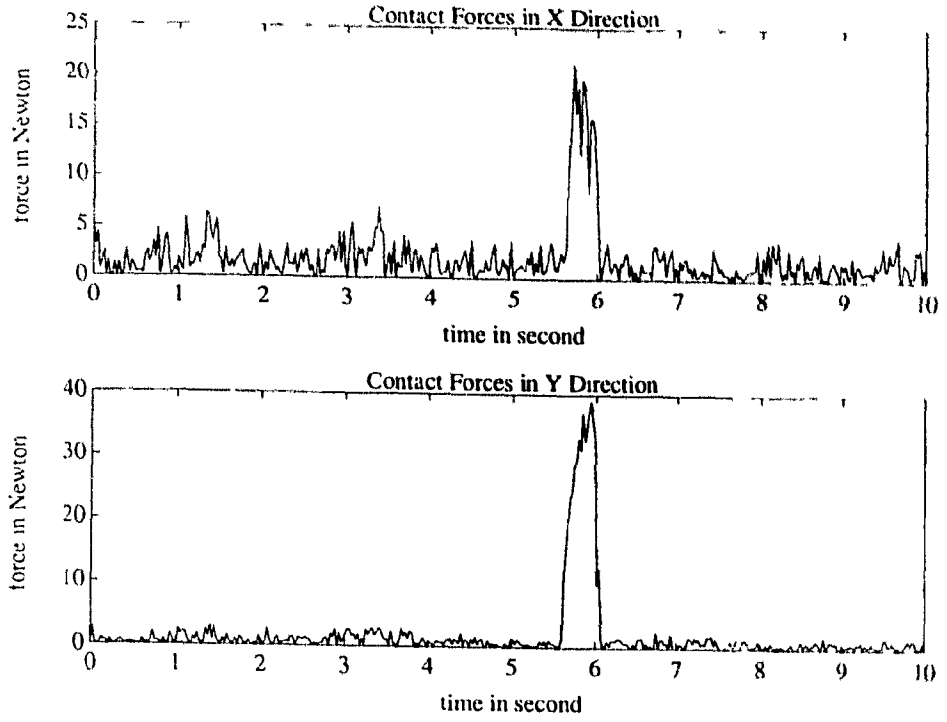


Figure 3.13: Contact forces in (a) Y-direction; (b) X-direction.

the surface to maintain an appropriate amount of contact force, and then leaves the wall going back to the initial point. Both free motion (position) and constrained motion (force) controls were applied. In the implementation, the mass of link 3, which includes the force sensor, was assumed to be unknown and was required to be estimated on line. The experimental results are shown in Figure 3.13 to Figure 3.16.

Table 3.1: Kinematic Parameters of PUMA 560 (joints 2,3,5)

i	α_{i-1} (degree)	a_{i-1} (meter)	d_i (meter)	θ_i (degree)
2	0	0	0	θ_2
3	0	0.432	-0.094	θ_3
5	0	0.433	0	θ_5

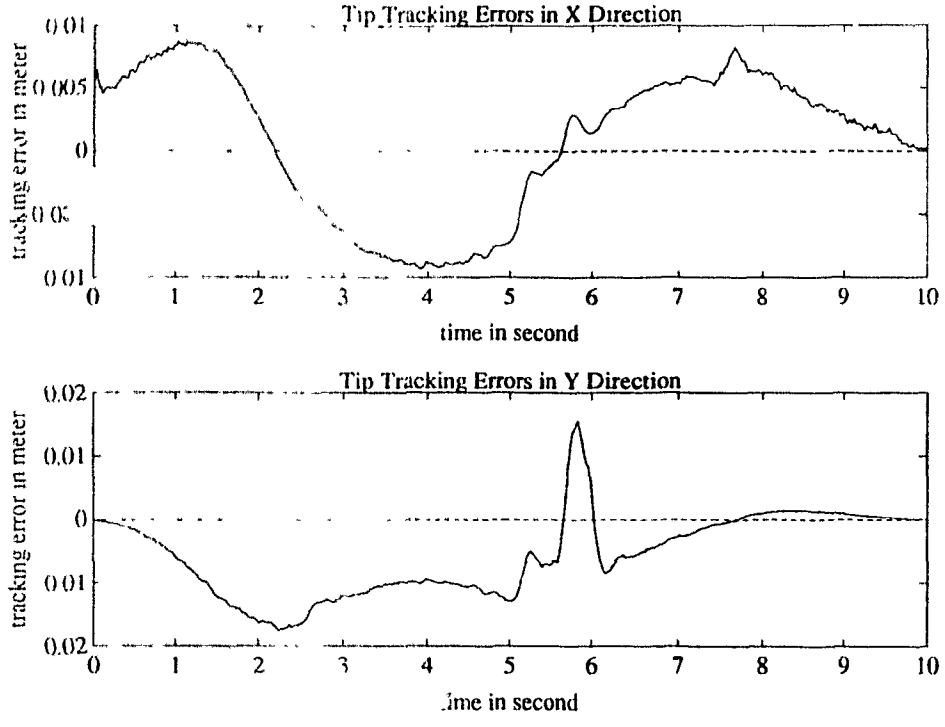


Figure 3.14: Tracking errors of robot tip in (a) Y-direction; (b) X-direction.

In Figure 3.13 the contact forces between the robot gripper and the wall detected by the wrist force sensor are plotted. Figure 3.13(a) shows the contact force in Y-direction which is orthogonal to the contact surface and is in general larger than the contact force in X-direction shown in Figure 3.13(b) which is the force along the tangent of the contact surface. Forces less than 10 Newton are noises picked by the force sensor and the forces in the time interval of 5.5s to 6.1s are contact forces. In Figure 3.14, tracking errors at the robot tip are plotted. As is shown in Figure 3.14(b) and 3.14(a) respectively, constrained motion does not affect the tracking accuracy in X-direction but does affect the tracking accuracy in Y-direction, the orthogonal direction of the contact surface, especially during the contact in the time interval of 5.5s to 6.1s when the tip motion in Y-direction is blocked by the contact surface. Figure 3.15 shows the estimated parameter values, i.e. the mass

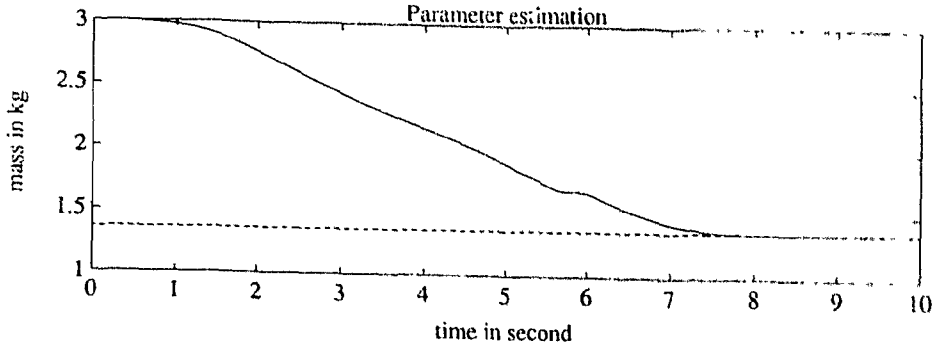


Figure 3.15: Estimated parameter value m_3.

Table 3.2: Dynamic Parameters of PUMA 560 (joints 2,3,5)

Link	Mass	Center of Mass			Diagonal Terms of Inertia				Friction
i	m_i	p_x	p_y	p_z	I_{xx}	I_{yy}	I_{zz}	I_{motor}	Coe.
2	16.8	0.432	0	-0.094	0.130	0.524	0.539	4.710	27.63
3	5.80	0.433	0	0	0.066	0.013	0.086	0.830	4.53
456	1.25	0.062	0	0	0.020	0.020	0.015	0.200	-

of link 3 with the force sensor. An initial guess of 3 kg was used to start the estimation. By using parameter update law (3.33), the estimated parameter values finally converge to the true value 1.36 kg. As is shown in Figure 3.15, the contact has very little influence on the parameter estimation. Control voltages supplied to the PUMA power amplifiers to drive the actuators at joints 2, 3, and 5 are plotted in Figure 3.16(a)–(c) respectively. As the maximum allowed control voltage is 10 volts in absolute value, Figure 3.16 shows that the control voltages in the implementation are well within the range.

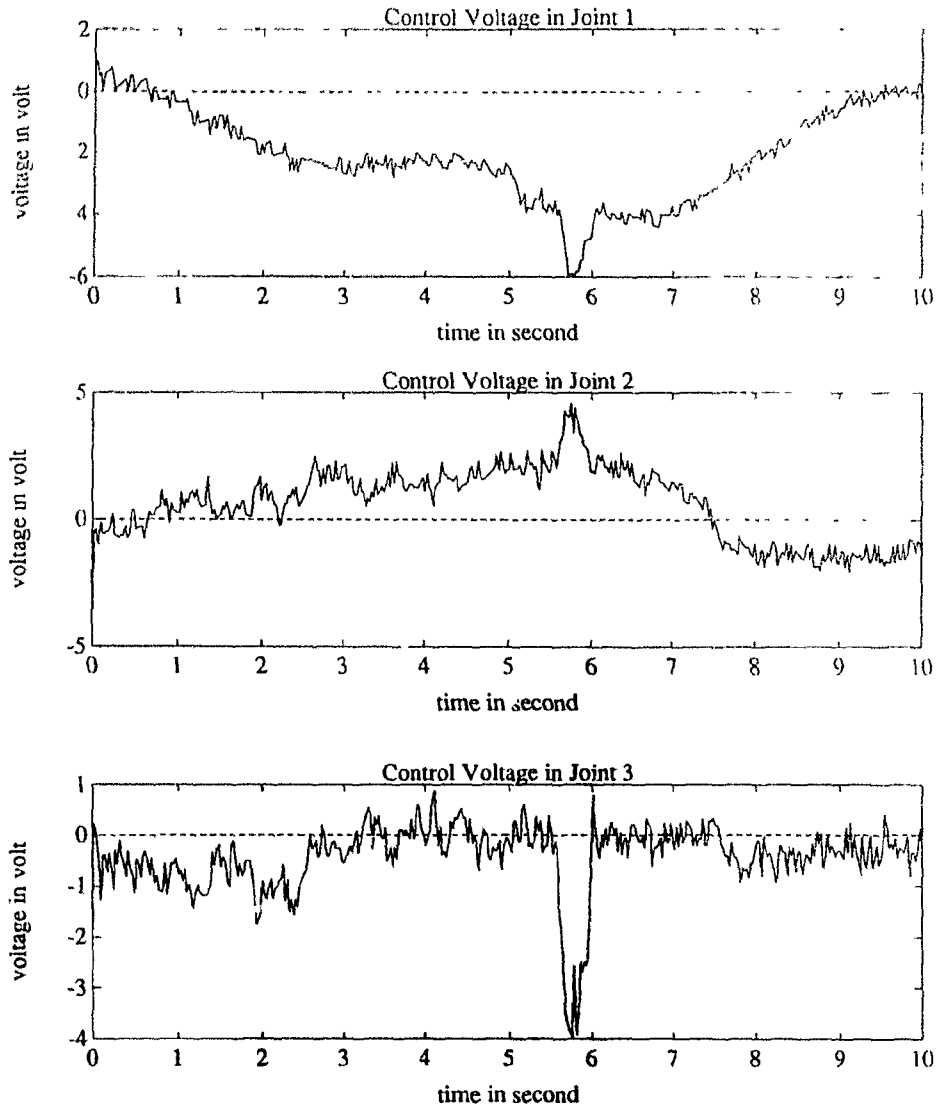


Figure 3.16: Control voltages applied to joint actuator amplifiers, (a) first joint; (b) second joint; (c) third joint.

Chapter 4

Coordination of Multiple Robots

4.1 Introduction

Due to the advantages of multi-arm robotic systems in handling large objects and in implementing automated assembly tasks, research of feasible manipulator coordination approaches has been intensive for years [133, 164]. The major issues of the present research in this area include the following [133, 164]:

1. Dynamics formulations for coordination scheme development;
2. Coordination strategies for performing stable and fine tasks;
3. Load distribution algorithms that can be used in real-time for optimal load sharing among the participating robots;
4. Regulation of internal forces which do not contribute to the motion of the object among the coordinating manipulators;
5. Adaptive laws for dealing with coordination problems for robots with unknown dynamics and possibly unknown payload.

As was reviewed in Chapter 1, there are three basic approaches to solve the coordination problem of a multi-robot system, namely the master-slave method, the parallel control method, and the object control method. These methods have their own attractive features and certain drawbacks as well. Feasible coordination structures are yet to be constructed for satisfactory solution to the main issues associated with a multi-robot system.

The main difference between controlling a single robot and multiple coordinating robots is that in the latter case all the participating robots have to work in a coordinative manner so as to achieve acceptable system efficiency in a given task. Among other things, it appears that an optimal dynamic load distribution strategy is critical for efficient coordination of the participating robots. It should distribute the load at the object level to each of the participating robots in an optimal manner so that the load shared by each of the participating robots is equal (after a torque normalization, of course). Under this structure, strategies for the control of individual robot, internal force regulation, dynamic parameter estimation, and dynamic uncertainty compensation, can be developed with certain flexibility.

In this chapter, we shall set up a dynamics model for multiple robot systems, and develop algorithms for dynamic load sharing [140, 141]. As will be shown, the load sharing algorithms obtained are computationally feasible and, therefore, have potentials in real-time applications. By properly distributing the instant load to each of the robots as a starting point, we shall construct a coordination scheme to deal with several coordination issues mentioned above [165]. A simulation study is included to compare the proposed method with several existing methods.

4.2 Dynamics Model of A Multi-Robot System

4.2.1 Basic Formulations

Position and Orientation of a Robot Manipulator

The position and orientation of a robot gripper in a reference coordinate frame $\{x, y, z\}$ can be described by the position and orientation of a coordinate frame attached to the gripper. The gripper frame can be described by a homogeneous transformation [3]

$$\mathbf{T}_{gripper} = \mathbf{T}_1 \cdot \mathbf{T}_2 \cdot \mathbf{T}_3 \cdot \mathbf{T}_4 \cdot \mathbf{T}_5 \cdot \mathbf{T}_6 \tag{4.1}$$

$$= \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.2}$$

where transformation $\mathbf{T}_i$ is associated with the frame attached to the i-th link and vector $\mathbf{p} = [p_x\ p_y\ p_z]^T$ specifies the position of the origin of the gripper frame in the reference frame, and (n_x, n_y, n_z), (o_x, o_y, o_z), and (a_x, a_y, a_z) are the principal unit vectors of the gripper frame with respect to the reference frame.

Velocity of the Manipulator

The velocity of the origin of the gripper frame in the reference frame is

$$\dot{\mathbf{x}} = \begin{bmatrix} \mathbf{v} \\ \boldsymbol{\omega} \end{bmatrix}$$

which is related to joint velocity $\dot{\mathbf{q}}$ by equation:

$$\dot{\mathbf{x}} = \mathbf{J}\dot{\mathbf{q}} \tag{4.3}$$

where $\mathbf{v} = [v_x\ v_y\ v_z]^T$ are the linear velocities of the gripper frame with respect to the reference frame, $\boldsymbol{\omega} = [\omega_x\ \omega_y\ \omega_z]^T$ represent the angular velocities of the origin of the gripper frame about the x, y, and z axes of the reference frame, $\dot{\mathbf{q}} = [\dot{q}_1\ \dot{q}_2\ \ldots\ \dot{q}_n]^T$ are joint velocities of the manipulator, and $\mathbf{J}$ is the configuration dependent Jacobian matrix.

Trajectory Specification

A position trajectory can be defined as a function of $\mathbf{T}_{gripper}$, i.e., $\mathbf{T}_{gripper}(t)$. Its first-order derivative is specified by a linear velocity $\mathbf{v}(t)$ and an angular velocity $\boldsymbol{\omega}(t)$ such that [166]

$$\mathbf{T}_n(t+\Delta t) = \mathbf{T}_n(t) \begin{bmatrix} 1 & -\omega_z(t) & \omega_y(t) & v_x(t) \\ \omega_z(t) & 1 & -\omega_x(t) & v_y(t) \\ -\omega_y(t) & \omega_x(t) & 1 & v_z(t) \\ 0 & 0 & 0 & 1 \end{bmatrix} \Delta t \qquad (4.4)$$

It is assumed that $\mathbf{T}_{gripper}(t)$ as well as $\mathbf{v}(t)$ and $\boldsymbol{\omega}(t)$ are differentiable, so that $\dot{\mathbf{v}}(t)$ and $\dot{\boldsymbol{\omega}}(t)$ exist.

Position and Orientation Errors

The desired Cartesian position and orientation in the reference frame $\mathbf{T}_d$ can be represented in the form of (4.2) and the actual position and orientation $\mathbf{T}_a$ can be obtained from (4.1). Thus the position and orientation error

$$\mathbf{e}_x = \mathbf{x}_d - \mathbf{x}_a = [dx\ \ dy\ \ dz\ \ \delta_x\ \ \delta_y\ \ \delta_z]^T \qquad (4.5)$$

can be expressed in the following as

$$\mathbf{T}_d = \mathbf{T}_a \begin{bmatrix} 1 & -\delta_z & \delta_y & dx \\ \delta_z & 1 & -\delta_x & dy \\ -\delta_y & \delta_x & 1 & dz \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.6}$$

It can readily be shown that

$$\mathbf{e}_x = \begin{bmatrix} dx \\ dy \\ dz \\ \delta_x \\ \delta_y \\ \delta_z \end{bmatrix} = \begin{bmatrix} \mathbf{n}_a \cdot (\mathbf{p}_d - \mathbf{p}_a) \\ \mathbf{o}_a \cdot (\mathbf{p}_d - \mathbf{p}_a) \\ \mathbf{a}_a \cdot (\mathbf{p}_d - \mathbf{p}_a) \\ (\mathbf{a}_a \cdot \mathbf{o}_d - \mathbf{a}_d \cdot \mathbf{o}_a)/2 \\ (\mathbf{n}_a \cdot \mathbf{a}_d - \mathbf{n}_d \cdot \mathbf{a}_a)/2 \\ (\mathbf{o}_a \cdot \mathbf{n}_d - \mathbf{o}_d \cdot \mathbf{n}_a)/2 \end{bmatrix} \tag{4.7}$$

where $\mathbf{n}_a, \mathbf{o}_a, \mathbf{a}_a, \mathbf{p}_a$ and $\mathbf{n}_d, \mathbf{o}_d, \mathbf{a}_d, \mathbf{p}_d$ are the column vectors of $\mathbf{T}_a$ and $\mathbf{T}_d$ respectively.

Velocity Errors

The actual joint velocity $\dot{\mathbf{q}}_a$ can be measured by joint tachometers so that the corresponding gripper velocity

$$\dot{\mathbf{x}}_a = [v_{xa} \;\; v_{ya} \;\; v_{za} \;\; \omega_{xa} \;\; \omega_{ya} \;\; \omega_{za}]^T$$

can be calculated by (4.3) as

$$\dot{\mathbf{x}}_a = \mathbf{J}\dot{\mathbf{q}}_a \tag{4.8}$$

The desired Cartesian velocity

$$\dot{\mathbf{x}}_d = [v_{xd} \;\; v_{yd} \;\; v_{zd} \;\; \omega_{xd} \;\; \omega_{yd} \;\; \omega_{zd}]^T$$

can be obtained from (4.4) as

$$\begin{bmatrix} 1 & -\omega_{zd} & \omega_{yd} & v_{xd} \\ \omega_{zd} & 1 & -\omega_{xd} & v_{yd} \\ -\omega_{yd} & \omega_{xd} & 1 & v_{zd} \\ 0 & 0 & 0 & 1 \end{bmatrix} = \frac{1}{\Delta t}(\mathbf{T}_d^{-1}(t) \cdot \mathbf{T}_d(t + \Delta t)) \tag{4.9}$$

where Δt is the sampling interval. It follows that

$$\dot{\mathbf{x}}_d(t) = \begin{bmatrix} v_{xd}(t) \\ v_{yd}(t) \\ v_{zd}(t) \\ \omega_{xd}(t) \\ \omega_{yd}(t) \\ \omega_{zd}(t) \end{bmatrix} = \begin{bmatrix} (\mathbf{n}_d(t) \cdot (\mathbf{p}_d(t + \Delta t) - \mathbf{p}_d(t)))/\Delta t \\ (\mathbf{o}_d(t) \cdot (\mathbf{p}_d(t + \Delta t) - \mathbf{p}_d(t)))/\Delta t \\ (\mathbf{a}_d(t) \cdot (\mathbf{p}_d(t + \Delta t) - \mathbf{p}_d(t)))/\Delta t \\ (\mathbf{a}_d(t) \cdot \mathbf{o}_d(t + \Delta t) - \mathbf{a}_d(t + \Delta t) \cdot \mathbf{o}_d(t))/(2 \cdot \Delta t) \\ (\mathbf{n}_d(t) \cdot \mathbf{a}_d(t + \Delta t) - \mathbf{n}_d(t + \Delta t) \cdot \mathbf{a}_d(t))/(2 \cdot \Delta t) \\ (\mathbf{o}_d(t) \cdot \mathbf{n}_d(t + \Delta t) - \mathbf{o}_d(t + \Delta t) \cdot \mathbf{n}_d(t))/(2 \cdot \Delta t) \end{bmatrix} \tag{4.10}$$

Hence the Cartesian velocity error

$$\dot{\mathbf{e}}_x = \dot{\mathbf{x}}_d - \dot{\mathbf{x}}_a \tag{4.11}$$

can now be calculated.

4.2.2 Dynamics of the Manipulated Object

The dynamic equation of the manipulated object can be described using Newton-Euler formulation in Cartesian space as

$$\mathbf{M}_o\ddot{\mathbf{x}}_o + \mathbf{B}_o\dot{\mathbf{x}}_o + \mathbf{M}_o\mathbf{g}_o + \mathbf{f}_e = \mathbf{f}_o \tag{4.12}$$

with

$$\mathbf{M}_o = \begin{bmatrix} \mathbf{m}_o & 0 \\ 0 & \mathbf{I}_o \end{bmatrix}, \qquad \mathbf{B}_o = \begin{bmatrix} 0 & 0 \\ 0 & \mathbf{b}_o \end{bmatrix}$$

$$\mathbf{b}_o = \begin{bmatrix} 0 & I_z\omega_z & -I_y\omega_y \\ -I_z\omega_z & 0 & I_x\omega_x \\ I_y\omega_y & -I_x\omega_x & 0 \end{bmatrix}, \qquad \ddot{\mathbf{x}}_o = \begin{bmatrix} \ddot{\mathbf{p}}_o \\ \dot{\boldsymbol{\omega}}_o \end{bmatrix}$$

where $\mathbf{m}_o = m\mathbf{I}_3$, a 3×3 diagonal matrix whose diagonal elements denote the mass of the object m, $\mathbf{I}_o = \text{diag}\{I_x\ I_y\ I_z\}$ is a 3×3 diagonal matrix whose diagonal elements denote the principal moments of the inertia of the object, $\mathbf{g}_o = [0\,0\,g\,0\,0\,0]^T$ with g being the gravitational acceleration, $\mathbf{p}_o = [p_x\ p_y\ p_z]^T$ is the position of a frame attached to the mass center of the object and $\boldsymbol{\omega}_o = [\omega_x\ \omega_y\ \omega_z]^T$ is the angular velocities of the object about the three principal axes of the frame at the mass center of the object, $\mathbf{f}_e$ denotes possible external force exerted on the mass center of the object from environment and it is zero during free motion, and

$$\mathbf{f}_o = \sum_{k=1}^{K} \mathbf{f}_k \tag{4.13}$$

is the geometric sum of $\mathbf{f}_k$ which is the force contributed by the k-th participating manipulator.

Let $\boldsymbol{\Lambda}_k$ $(k = 1, \ldots, K)$ be matrices satisfying

$$\sum_{k=1}^{K} \boldsymbol{\Lambda}_k = \mathbf{I} \tag{4.14}$$

$\boldsymbol{\Lambda}_k$ is called the kth *distribution matrix* that quantitatively characterizes the involvement of the ith robot in a load sharing process. Using $\boldsymbol{\Lambda}_k$, we can write

$$\mathbf{f}_k = \boldsymbol{\Lambda}_k \mathbf{f}_o \tag{4.15}$$

4.2.3 Dynamics of Participating Robotic Manipulator

Assume that there are totally K coordinating manipulators that handle a rigid object through rigid grasp. The dynamic equation of the k-th manipulator with n DOF can be described in the Cartesian space as

$$\mathbf{H}_{xk}(\mathbf{q}_k)\ddot{\mathbf{x}}_k + \mathbf{C}_{xk}(\mathbf{q}_k, \dot{\mathbf{q}}_k)\dot{\mathbf{x}}_k + \mathbf{g}_{xk}(\mathbf{q}_k) + \mathbf{f}_k = \mathbf{f}_{rk} \quad k = 1, \ldots, K \tag{4.16}$$

which in conjunction with (4.16) leads to

$$\mathbf{H}_{xk}(\mathbf{q}_k)\ddot{\mathbf{x}}_k + \mathbf{C}_{xk}(\mathbf{q}_k, \dot{\mathbf{q}}_k)\dot{\mathbf{x}}_k + \mathbf{g}_{xk}(\mathbf{q}_k) + \mathbf{\Lambda}_k\mathbf{f}_o = \mathbf{f}_{rk} \quad k = 1, \ldots, K \tag{4.17}$$

where $\mathbf{f}_{rk}$ is the generalized force generated at the end-effector of the kth manipulator by its actuators, and $\dot{\mathbf{x}}_k$ is the Cartesian velocity of the kth manipulator gripper.

4.3 Optimal Dynamic Load Distribution

In a certain sense each of the algorithms to be developed provides an optimal load distribution to multiple coordinating manipulators in which the force amplitude of each actuator is assumed to be bounded. To be specific, the objective function to be adopted is the p-norm of the joint forces as the p-norm is differentiable and approaches to the infinity norm as $p \longrightarrow \infty$ meaning that, with a sufficiently large p, torque constraints

$$|\tau_j| \leq b_j \tag{4.18}$$

can *implicitly* be incorporated into the objective function so that one actually deals with an *unconstrained* optimization problem. Moreover, it will be shown that this p-norm type objective function possesses a set of closed-form formulas for its gradient and Hessian matrix which turns out to be positive definite in the whole parameter space if the optimal distribution algorithm is applied only to the load components

greater than a given threshold. As a consequence, the strict convexity of the objective function enables one to use the classic Newton's method to find the solution with an order-two convergence rate.

4.3.1 Basic Equations and Problem Formulation

In this section we first derive several basic equations for force distribution of K coordinating robots. Joint torque normalization is then used to eliminate a set of inequality constraints, which makes it possible to formulate a force distribution task as an unconstrained optimization problem.

Basic Equations

Consider a K-robot system handling a single object. It is assumed throughout that the object is held rigidly by K grippers so that there is no relative motion between the object and the grippers.

The dynamics of the object is given by (4.12) with the assumption that $\mathbf{f}_v \equiv 0$

$$\mathbf{M}_o\ddot{\mathbf{x}}_o + \mathbf{B}_o\dot{\mathbf{x}}_o + \mathbf{M}_o\mathbf{g}_o = \mathbf{f} \tag{4.19}$$

Given a desired motion trajectory of the object, $\mathbf{x}_d(t)$, the control force $\mathbf{f}$ may be assigned by using an inverse dynamics feedback law as

$$\mathbf{f} = \mathbf{B}_o\dot{\mathbf{x}}_o + \mathbf{M}_o\mathbf{g}_o + \mathbf{M}_o[\ddot{\mathbf{x}}_d + \mathbf{K}_v(\dot{\mathbf{x}}_d - \dot{\mathbf{x}}_o) + \mathbf{K}_p(\mathbf{x}_d - \mathbf{x}_0)] \tag{4.20}$$

where $\mathbf{K}_v$ and $\mathbf{K}_p$ are the differential and proportional gains, respectively. It is the responsibility of the participating robots to provide the object with the control force $\mathbf{f}$ given in (4.20), and such a force is often referred to as the *load* for the robots.

With the distributed load $\mathbf{\Lambda}_k\mathbf{f}$ (see (4.15), the dynamic equation of the k-th robot in its joint space becomes

$$\mathbf{H}_k\ddot{\mathbf{q}}_k + \mathbf{C}_k(\mathbf{q}_k, \dot{\mathbf{q}}_k)\dot{\mathbf{q}}_k + \mathbf{g}_k(\mathbf{q}_k) + \mathbf{J}_k^T(\mathbf{q}_k)\mathbf{\Lambda}_k\mathbf{f} = \hat{\boldsymbol{\tau}}_k \tag{4.21}$$

If the inverse dynamic control is chosen to drive the robot so as to follow a desired trajectory $\mathbf{q}_{kd}(t)$, then the joint force $\hat{\boldsymbol{\tau}}_k$ that the robot actuators must provide is given by

$$\hat{\boldsymbol{\tau}}_k = \hat{\mathbf{p}}_k + \mathbf{J}_k^T(\mathbf{q}_k)\boldsymbol{\Lambda}_k\mathbf{f} \tag{4.22}$$

where

$$\hat{\mathbf{p}}_k = \mathbf{H}_k(\mathbf{q}_k)[\ddot{\mathbf{q}}_{kd} + \mathbf{K}_{kv}(\dot{\mathbf{q}}_{kd} - \dot{\mathbf{q}}_k) + \mathbf{K}_{kp}(\mathbf{q}_{kd} - \mathbf{q}_k)] + \mathbf{C}_k(\mathbf{q}_k, \dot{\mathbf{q}}_k)\dot{\mathbf{q}}_k + \mathbf{g}_k(\mathbf{q}_k)$$

Together equations (4.19)-(4.22) represent an analytic description of the dynamic behavior of the overal' ystem.

Joint Torque Normalization and Problem Formulation

Writing $\boldsymbol{\Lambda}_k\mathbf{f}$ as

$$\boldsymbol{\Lambda}_k\mathbf{f} = \mathbf{F}\lambda_k$$

where $\mathbf{F} = \text{diag}(\mathbf{f})$, i.e. a diagonal matrix with $\mathbf{f}$ on its main diagonal, and λ_k is a vector formed by the main diagonal of $\boldsymbol{\Lambda}_k$, (4.22) becomes

$$\hat{\mathbf{p}}_k + \hat{\mathbf{Q}}_k\lambda_k = \hat{\boldsymbol{\tau}}_k \tag{4.23}$$

with

$$\hat{\mathbf{Q}}_k = \mathbf{J}_k^T\mathbf{F}$$

and (4.14) becomes

$$\sum_{k=1}^{K} \lambda_k = [1 \ \ldots \ 1]^T \triangleq \mathbf{e}^T \tag{4.24}$$

Denoting by $\hat{\tau}_{k_i}$ the i-th component of $\hat{\boldsymbol{\tau}}_k$, a bound for each $|\hat{\tau}_{k_i}|$ may be set to reflect the fact that the force that can be provided by each actuator is limited in practice, i.e.

$$|\hat{\tau}_{k_i}| \leq b_{k_i} \qquad \text{for } 1 \leq i \leq n, \ \ 1 \leq k \leq K$$

Let

$$\mathbf{b}_k = [b_{k_1} \ \ldots \ b_{k_n}]^T \quad \text{and} \quad \mathbf{B}_k = \text{diag}(\mathbf{b}_k)$$

(4.23) can be normalized by multiplying $\mathbf{B}_k^{-1}$ on the both sides of (4.23):

$$\mathbf{p}_k + \mathbf{Q}_k \lambda_k = \boldsymbol{\tau}_k, \quad 1 \leq k \leq K \tag{4.25}$$

where

$$\begin{aligned} \mathbf{p}_k &= \mathbf{B}_k^{-1} \hat{\mathbf{p}}_k \\ \mathbf{Q}_k &= \mathbf{B}_k^{-1} \hat{\mathbf{Q}}_k \\ \boldsymbol{\tau}_k &= \mathbf{B}_k^{-1} \hat{\boldsymbol{\tau}}_k \end{aligned}$$

The constraints on magnitude of the joint forces now become

$$\|\boldsymbol{\tau}_k\|_\infty \leq 1, \quad 1 \leq k \leq K \tag{4.26}$$

where $\|\boldsymbol{\tau}_k\|_\infty$ denotes the infinity norm of $\boldsymbol{\tau}_k$ defined as the largest component of $\boldsymbol{\tau}_k$ in absolute value.

Let

$$\boldsymbol{\tau} = [\boldsymbol{\tau}_1^T \ \ldots \ \boldsymbol{\tau}_k^T]^T$$

the optimal load distribution problem with bounded joint force constraints can now be formulated as

The Optimization Problem (OP): Given a load vector $\mathbf{f} \in R'^n$, find a set of distribution-factors λ_k ($k = 1, \ldots, K$) that minimizes $\|\boldsymbol{\tau}\|_\infty$ with constraints (4.24).

While a solution to the OP does not necessarily satisfy (4.26), having a solution to the OP with $\|\boldsymbol{\tau}\|_\infty > 1$ simply means that there at least is one actuator in one of the K participating robots, for which the load distributed is more than it can undertake. Consequently, such a solution is considered unacceptable.

4.3.2 A Modified Optimization Problem and Its Solution: The General Case

By (4.24), equation (4.25) with $k = K$ can be written as

$$\bar{\mathbf{p}}_K - \mathbf{Q}_K \sum_{l=1}^{K-1} \lambda_l \;=\; \boldsymbol{\tau}_K \tag{4.27}$$

where

$$\bar{\mathbf{p}}_K = \mathbf{p}_K + \mathbf{Q}_K \mathbf{e}$$

A compact vector equation can be obtained by putting (4.25) with $k = 1, \ldots, K-1$ and (4.27) together as

$$\mathbf{r} + \mathbf{Q}\mathbf{x} \;=\; \boldsymbol{\tau} \tag{4.28}$$

where

$$\mathbf{r} = \begin{bmatrix} \mathbf{p}_1 \\ \vdots \\ \mathbf{p}_{K-1} \\ \bar{\mathbf{p}}_K \end{bmatrix}, \quad \mathbf{Q} = \begin{bmatrix} \mathbf{Q}_1 & & & & \\ & \mathbf{Q}_2 & & \mathbf{0} & \\ & & \ddots & & \\ & \mathbf{0} & & \ddots & \\ & & & & \mathbf{Q}_{K-1} \\ -\mathbf{Q}_K & -\mathbf{Q}_K & \cdots & \cdots & -\mathbf{Q}_K \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_{K-1} \end{bmatrix}$$

With constraints (4.24) incorporated into (4.28), we are now dealing with an unconstrained optimization problem and the goal is to find vector $\mathbf{x}$ that minimizes $\|\boldsymbol{\tau}\|_\infty$. The difficulty with this optimization problem is a classic one: its objective function $\|\mathbf{r} + \mathbf{Q}\mathbf{x}\|_\infty$ is *not* differentiable with respect to variable $\mathbf{x}$. An alternative approach is to optimize $\|\mathbf{r} + \mathbf{Q}\mathbf{x}\|_{2p}$ with a sufficiently large positive integer p, as $\|\mathbf{r} + \mathbf{Q}\mathbf{x}\|_{2p}$ with any fixed p is differentiable w.r.t. $\mathbf{x}$ and approximates $\|\mathbf{r} + \mathbf{Q}\mathbf{x}\|_\infty$ as $p \longrightarrow \infty$. More precisely, for an nK-dimensional $\boldsymbol{\tau}$, $\|\boldsymbol{\tau}\|_{2p}$ can be estimated in terms of $\|\boldsymbol{\tau}\|_\infty$ as

$$\|\boldsymbol{\tau}\|_\infty \leq \|\boldsymbol{\tau}\|_{2p} \leq 1.17\|\boldsymbol{\tau}\|_\infty \tag{4.29}$$

whenever $p \geq 8$. From (4.29) it also follows that a vector $\boldsymbol{\tau}$ with $\|\boldsymbol{\tau}\|_{2p} \leq 1$ automatically satisfies condition (4.26). Therefore, solutions to the following modified optimization problem may be found useful in practice provided that p is chosen to be sufficiently large.

The Modified Optimization Problem (MOP): Given a load vector $\mathbf{f} \in R^m$, find vector $\mathbf{x} \in R^{(K-1)m}$ such that

$$v(\mathbf{x}) = \|\mathbf{r} + \mathbf{Q}\mathbf{x}\|_{2p}^{2p} \qquad \text{is minimized} \tag{4.30}$$

Note that in order to ease the numerical computation involved in the subsequent optimization algorithms, in (4.30) we have used $\|\boldsymbol{\tau}\|_{2p}^{2p}$ instead of $\|\boldsymbol{\tau}\|_{2p}$ as the object function. Clearly the solution to minimizing $\|\boldsymbol{\tau}\|_{2p}^{2p}$ is identical to the solution to minimizing $\|\boldsymbol{\tau}\|_{2p}$. A solution to the MOP is said to be *admissible* if it gives

$$\|\boldsymbol{\tau}\|_\infty = \|\mathbf{r} + \mathbf{Q}\mathbf{x}\|_\infty \leq 1$$

As discussed above, an admissible solution provides such a load distribution for the participating robots that $\|\boldsymbol{\tau}\|_{2p}$ is minimized and the required joint torque for each individual joint does not exceed its limit.

To solve the MOP, let us denote

$$\begin{aligned} N &= Kn \\ M &= (K-1)m \\ \mathbf{r} &= [r_1 \ \ldots \ r_N]^T \\ \mathbf{Q} &= (q_{ij}), \quad 1 \leq i \leq N, \quad 1 \leq j \leq M \\ \mathbf{x} &= [x_1 \ \ldots \ x_M]^T \\ s_i &= r_i + \sum_{j=1}^{M} q_{ij} x_j, \quad 1 \leq i \leq N \end{aligned}$$

and write the objective function as

$$v(\mathbf{x}) = \sum_{i=1}^{N} s_i^{2p} \tag{4.31}$$

The k-th component of the gradient ∇v and the (k,l)-entry of the Hessian matrix $\mathbf{H}$ of $v(\mathbf{x})$ are given by

$$\frac{\partial v}{\partial x_k} = 2p\sum_{i=1}^{N} s_i^{2p-1} q_{ik}, \qquad 1 \le k \le M$$

and

$$h_{kl} \triangleq \frac{\partial^2 v}{\partial x_k \partial x_l} = 2p(2p-1)\sum_{i=1}^{N} s_i^{2(p-1)} q_{ik} q_{il}, \qquad 1 \le k, l \le M$$

respectively. It follows that

$$\nabla v = 2p\, \mathbf{Q}^T [s_1^{2p-1} \ \ldots \ s_N^{2p-1}]^T$$

and

$$\mathbf{H} = \mathbf{W}^T \mathbf{W} \tag{4.32}$$

where

$$\mathbf{W} = \sqrt{2p(2p-1)}\, \mathbf{S}^{p-1} \mathbf{Q} \tag{4.33}$$

$$\mathbf{S} = \mathrm{diag}(s_1, \ \ldots, \ s_N) \tag{4.34}$$

Therefore, if $\mathrm{rank}(\mathbf{W}) = M$, then $\mathbf{H}$ is positive definite and the objective function $v(\mathbf{x})$ is globally strictly *convex* over the entire parameter space and the MOP has a unique solution [167]. Even if matrix $\mathbf{H}$ fails to be of full rank, $\mathbf{H}$ is then positive semi-definite and $v(\mathbf{x})$ remains globally convex and solution(s) to the MOP corresponding to the global minimum of $v(\mathbf{x})$ exist [168, Chapters 2 and 3].

There are a number of optimization methods suitable for finding a global minimum point of a globally convex objective function. Reliable methods such as the Davidon-Fletcher-Powell method, the BFGS method, the Fletcher-Reeves method, and the Polak-Ribiere method often lead to fast, global convergence but they are computationally expensive owing to the requirement of a line search procedure during each iteration [167]. On the other hand, the nonquadratic extension of the

well-known conjugate gradient algorithm, often called the quadratic approximation (Q-A) method, is an inexpensive alternative as it does not need line search and its global convergence is guaranteed if the objective function is convex [167]. A problem with the Q-A method is its slow convergence. Perhaps the fastest yet simple method suitable for the present case is the classic Newton's method. If the number of robots in use is $K = 2$, then $M = m \leq 6$ and the order two convergence rate of the Newton's method for such a small size optimization problem becomes especially attractive [167, Chapter 7]. The only problem with the Newton's method is that the convexity of $v(\mathbf{x})$ must be *strict* in order to guarantee the nonsingularity of the Hessian matrix. In the next section, we present an easy-to-apply approach for the case $K = 2$, which will assure the nonsingularity of $\mathbf{H}$.

4.3.3 Two Special Cases

The K=2 Case

A common situation in practice is when the system uses two coordinating robots. The parameter vector $\mathbf{x}$ then becomes $\mathbf{x} = \lambda_1 \in R^m$, and matrix $\mathbf{Q}$ in the basic equation (4.28) can be written as

$$\mathbf{Q} = \begin{bmatrix} \mathbf{Q}_1 \\ -\mathbf{Q}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{B}_1^{-1}\mathbf{J}_1^T \\ -\mathbf{B}_2^{-1}\mathbf{J}_2^T \end{bmatrix} \mathbf{F}$$

which implies that

$$\mathbf{W}^T = \sqrt{2p(2p-1)}\,\mathbf{F}\,[\mathbf{J}_1 \quad \mathbf{J}_2] \begin{bmatrix} \mathbf{B}_1 & \mathbf{0} \\ \mathbf{0} & -\mathbf{B}_2 \end{bmatrix}^{-1} \mathbf{S}^{p-1} \tag{4.35}$$

It follows that matrix $\mathbf{H}$ is nonsingular if and only if

- matrices $\mathbf{S}$ and $\mathbf{F}$ are nonsingular and

- matrix $[\mathbf{J}_1 \quad \mathbf{J}_2] \in R^{m\times 2n}$ has full row rank, i.e.

$$\text{rank}[\mathbf{J}_1 \quad \mathbf{J}_2] = m \tag{4.36}$$

Condition (4.36) usually holds unless there exists a configuration during the motion which is singular for both manipulators. By (4.31), the objective function $v(\mathbf{x})$ depends only on those *nonzero* s_i's so that

$$\det\mathbf{S} = \prod_{i=1}^{N} s_i \neq 0$$

may be assumed without loss generality. Now recall that $\mathbf{F}$ is the diagonal matrix generated by the load vector $\mathbf{f}$, i.e.

$$\mathbf{F} = \text{diag}(\mathbf{f})$$

Thus $\mathbf{F}$ will become singular (or nearly singular) if zero (or very small) component(s) are found in $\mathbf{f}$. A simple and reasonable approach to dealing with this singularity problem is that a fixed $\varepsilon > 0$ is set as a threshold and the proposed load distribution algorithm is applied only to the components of $\mathbf{f}$ with magnitude greater than ε while the rest of the components of $\mathbf{f}$ are distributed *equally* to the two manipulators. Note that the uniform distribution policy will not seriously degrade the overall load sharing efficiency since it only applies to those negligible components in $\mathbf{f}$. As a result, this approach leads to a nonsingular diagonal matrix $\mathbf{F}$ with reduced dimension (denoted again by m) and hence a full row-rank matrix $\mathbf{W}^T$ provided that condition (4.36) holds. With necessary modifications of the dimension of the relevant matrices and vectors, all equations derived so far remain valid. Moreover, from the discussion given in the last paragraph of Section 4.3.2, it follows that $v(\mathbf{x})$ is strictly convex and the classic Newton's method is applicable provided that no configurations exist which would lead to

$$\text{rank}[\mathbf{J}_1 \quad \mathbf{J}_2] < m$$

We now summarize the load sharing scheme developed as follows.

Algorithm 4.1 (The vector case with $K = 2$)

Step 0 Set a threshold $\varepsilon > 0$, $i = 0$ and $K = 2$. Choose $\mathbf{x}(i) = \lambda_1(i) = 0.5\mathbf{e}$.

Step 1 Input load vector $\mathbf{f}$. Use the threshold to form, if necessary, a new $\mathbf{f}$ with all components greater than ε in absolute value.

Step 2 If the dimension of $\mathbf{f}$ has been reduced, use 0.5 for all components in λ_1 corresponding to those neglected components in the original $\mathbf{f}$, otherwise skip to the next step.

Step 3 Compute $\mathbf{r}$ and $\mathbf{Q}$.

Step 4 Compute $\{s_j,\ 1 \leq j \leq N\}$. Reduce this set so that all s_i's are nonzero.

Step 5 Delete those rows in $\mathbf{Q}$ corresponding to zero s_i's.

Step 6 Compute $\nabla v(\mathbf{x}(i))$, $\mathbf{H}(\mathbf{x}(i))$, and set

$$\mathbf{x}_{i+1} = \mathbf{x}_i - \mathbf{H}^{-1}(\mathbf{x}(i))\nabla v(\mathbf{x}(i))$$

Step 7 If $\|\nabla v(\mathbf{x}(i+1))\|_2 <$ a prescribed tolerance, output the solution $\lambda_1^* = \mathbf{x}(i+1)$ and $\lambda_2^* = \mathbf{e} - \mathbf{x}(i+1)$, and save λ_1^* as the initial value $\mathbf{x}(0)$ for the next run. (i.e. it will skip step 0 ever since), otherwise replace i by $i+1$ and return to Step 4.

In the above algorithm, the solution λ_1^* obtained at present will serve as an initial value of parameter $\mathbf{x}$ for finding the optimal load distribution for the next run. Such a choice of an initial point will in general lead to a quick convergence, especially if the robot motion is of path tracking type with no sudden contact between the robotic

system and its environment so that an optimal solution $\mathbf{x}(t)$ as a time function is *continuous*.

An alternative approach to dealing with the singularity of the Hessian matrix in the classic Newton's method is to add a matrix $\mu\mathbf{I}$ with small $\mu > 0$ to matrix $\mathbf{H}$ so that the modified Hessian matrix $\mathbf{H} + \mu\mathbf{I}$ is always positive definite. Obviously a larger μ is of help for $\mathbf{H}$ to get rid of ill-conditioned area while a smaller μ leads to a faster convergence as it is closer to the classic Newton's method. As will be demonstrated in Section 4.3.4, the threshold approach as well as the $\mu\mathbf{I}$ modification work well in our simulation studies on force distribution of planar and non-planar 2-robot systems.

The Scalar Case

So far each distribution factor λ_k has been treated as an m-dimensional vector. In this subsection we shall consider the case where each λ_k is assumed to be a scalar and $K = 2$. As may be expected, this assumption considerably reduces the computation required in finding a solution to the MOP.

Denoting $\mathbf{q} = \mathbf{Q}\mathbf{e} = [q_1 \ \ldots \ q_{2n}]^T$, the objective function can be expressed as

$$\begin{aligned} v(\lambda_1) &= \|\mathbf{r} + \mathbf{Q}\mathbf{e}\lambda_1\|_{2p}^{2p} \\ &= \|\mathbf{r} + \mathbf{q}\lambda_1\|_{2p}^{2p} \\ &= \sum_{i=1}^{2n} (r_i + \lambda_1 q_i)^{2p} \end{aligned} \tag{4.37}$$

where $\mathbf{q} = \mathbf{Q}\mathbf{e}$ and λ_1 is at the present case the scalar parameter to be determined. From (4.37) we compute

$$\frac{dv}{d\lambda_1} = 2p \sum_{i=1}^{2n} (r_i + \lambda_1 q_i)^{2p-1} q_i \tag{4.38}$$

and

$$\frac{d^2v}{d\lambda_1^2} = 2p(2p-1) \sum_{i=1}^{2n} (r_i + \lambda_1 q_i)^{2(p-1)} q_i^2 \tag{4.39}$$

By (4.39) $v''(\lambda_1) > 0$ (unless $v(\lambda_1)$ itself is zero), so $v(\lambda_1)$ is strictly convex. The uniqueness of the global minimum of $v(\lambda_1)$ in turn implies that the odd-order polynomial (4.38) has only *one* real zero in the entire parameter space $-\infty < \lambda_1 < \infty$. Moreover, since $v'(+\infty)$ and $v'(-\infty)$ always have opposite signs, a finite interval $[\lambda_{1l} \quad \lambda_{1r}]$ can be found such that

$$v'(\lambda_{1l}) \times v'(\lambda_{1r}) < 0$$

and the golden section or the bisection method [167] can be used to effectively find the unique real solution to equation $v'(\lambda_1) = 0$ i.e.

$$\sum_{i=1}^{2n} (r_i + \lambda_1 q_i)^{2p-1} q_i \; = \; 0 \tag{4.40}$$

Concerning the determination of the interval $[\lambda_{1l} \quad \lambda_{1r}]$, the "continuity argument" given in the $k = 2$ case applies and one may conclude that after the very first load distribution task is accomplished, the real solution to equation (4.40) can be taken as the midpoint of a (small) interval $[\lambda_{1l} \quad \lambda_{1r}]$ suitable for the next run.

The algorithm listed below summarizes the main points discussed above and can be used for the scalar case with $K = 2$.

Algorithm 4.2 (The scalar case with $K = 2$)

Step 0 Input load vector $\mathbf{f}$ and compute $\mathbf{r}$, $\mathbf{q}$.

Step 1 Find λ_{1l} and λ_{1r} such that $\sum_{i=1}^{2n} (r_i + \lambda_1 q_i)^{2p-1}$ has opposite signs at $\lambda_1 = \lambda_{1l}$ and $\lambda_1 = \lambda_{1r}$.

Step 2 Apply the golden section or bisection method to find λ_1^* such that

$$\sum_{i=1}^{2n} (r_i + \lambda_1^* q_i)^{2p-1} q_i = 0.$$

Step 3 Output the optimal distribution factors λ_1^* and $\lambda_2^* = 1 - \lambda_1^*$. Save λ_1^* as the midpoint of an interval $[\lambda_{1L} \quad \lambda_{1r}]$ for the next run.

4.3.4 A Simulation Study

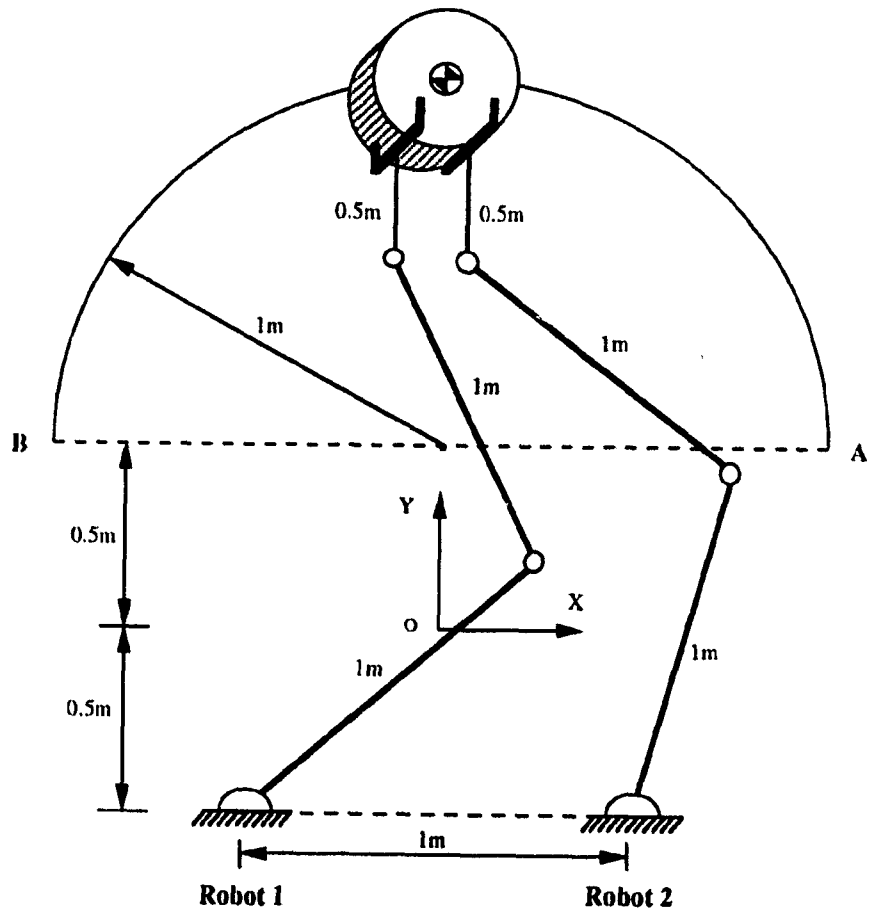


Figure 4.1: Two 3 DOF vertical planar robots.

To illustrate the proposed load sharing algorithms, simulations are carried out using a two-robot system.

Simulation Using Two 3 DOF Planar Robots

Consider a robotic system shown in Figure 4.1, where two 3 DOF vertical planar robots are carrying a 5 kg plate from point A to point B along the upper half of a circle in 1 second. As the robots are redundant, the third link of both robots can be kept vertical to the ground during the entire motion.

As is shown in Figure 4.3(a), while the y component of the load is always away from zero, its x component has been relatively small and becomes zero at time $t = 0.5$. This necessitates the use of a proper threshold ε in order to implement Algorithm 1. With $p = 6$ and $\varepsilon = 15$, the sub-optimal load distribution pattern obtained from Algorithm 4.1 is shown in Figure 4.3(b) and the normalized joint forces of the robots are shown in Figure 4.3(c) and 4.3(d), respectively. It appears that condition $\|\boldsymbol{\tau}\|_\infty \leq 1$ holds during the entire motion and the solution obtained is therefore acceptable. It is observed that the Newton's method provides a fast convergence. As a matter of fact, with a sampling rate of 200 Hz, Algorithm 4.1 converged usually after 3 iterations to the optimal λ_1 as compared to about 30 iterations to achieve the same solution if the Q-A method is employed. To demonstrate the importance of using a sufficiently large p, another load sharing pattern is obtained using Algorithm 4.1 with $p = 1$ (i.e. the least square case). The resulting λ_1 is depicted in Figure 4.4(b) and the corresponding normalized joint forces are shown in Figure 4.4(c) and (d), respectively. It is observed that $\|\boldsymbol{\tau}\|_\infty$ exceeds 1 when $t \in [0,\ 0.09]$ and $t \in [0.69,\ 1.0]$ and the solution is therefore not acceptable.

Next the distribution factor λ_1 is treated as a scalar and Algorithm 4.2 is implemented. With $p = 6$, (4.40) becomes a 11th-order algebraic equation whose unique real root is found by applying the golden section method to an interval centered around the previous solution λ_1. The length of such an interval can be kept as small as 0.02 and, within 3 iterations, the solution can be obtained with an error less than 10^{-6}. The resulting λ_1 and $\lambda_2 = 1 - \lambda_1$ are plotted in Figure 4.5(a), and the normalized joint forces of the two robots are shown in Figure 4.5(c) and 4.5(d), respectively. The figures clearly show that the load sharing pattern is acceptable since $\|\boldsymbol{\tau}\|_\infty < 1$ holds during the entire motion, although the joint forces required are slightly greater than that obtained from Algorithm 4.1. For the comparison purpose the norm trajectories $\|\boldsymbol{\tau}\|_\infty$ for the three cases discussed above are shown

in Figure 4.6.

Finally we increase the mass of the plate from 5 kg to 5.5 kg and apply Algorithm 4.1 with $p = 6$ and $\epsilon = 15$ to distribute the load. The results are shown in Figure 4.7. It is noted that for $t \in [.78 \;\; 1]$ the system becomes overloaded as *both* $\|\boldsymbol{\tau}_1\|_\infty$ and $\|\boldsymbol{\tau}_2\|_\infty$ exceeds 1 (i.e. at least one joint in each robot is required to provide a joint force larger than its upper bound). This in a certain sense demonstrates a desired feature of the proposed algorithm, that is, with a large p the algorithm nearly uniformly distributes the entire load over the participating robots – if one robot gets overloaded, little help can be found from the others as they are very likely also overloaded.

Simulation Using Two 3 DOF Non-Planar Robots

We now consider a system shown in Figure 4.8. It consists of two non-planar robots, each of which has 3 DOF with a structure similar to the first three joints of a PUMA 560 robot. The two robot arms rigidly hold a 5 kg mass and move it along a spatial curve characterized by the following set of parametric equations

$$\begin{aligned} x(t) &= 0.25\sin 2t \\ y(t) &= \cos t \\ z(t) &= 1 + 0.75\sin t \end{aligned}$$

where parameter t corresponds to the time instant varying from 0 to π (seconds). With link lengths $l_1 = 1$, $l_2 = 1$, and $l_3 = 0.5$, the above curve is located inside the system's workspace.

Instead of using a threshold to overcome the singularity of $\mathbf{H}$, a modified Hessian matrix, i.e. $\mathbf{H} + \mu\mathbf{I}$ with $\mu = 0.02$ is employed in Algorithm 4.1 with $p = 6$. It is found that except the very first force distribution task at point $[x(0) \;\; y(0) \;\; z(0)]^T = [0 \;\; 1 \;\; 1]^T$ which requires 6 iterations to achieve the optimal distribution, the algorithm converges usually after 3 iterations. Trajectories of the force to be distributed,

the distribution factor λ_1, and the normalized joint forces for robot 1 and 2 are shown in Figure 4.9(a)-(d), respectively. Figure 4.10 shows the trajectories of $\|\tau_1\|_\infty$ and $\|\tau_2\|_\infty$, from which it is observed that the load has been fairly uniformly distributed to the two participating robots.

4.4 A Coordination Scheme

In this section, a framework for motion coordination of a multi-robot system will be introduced. It utilizes the optimal load distribution described in the preceding section as a starting point that dynamically links the object motion to the motion of each participating robot. This framework consists of an object-level command controller and several task-level manipulator controllers. Issues of internal forces, adaptation, and flexibility of accommodating different control algorithms of this framework are also discussed. A simulation study is included to compare the proposed method with several existing methods.

4.4.1 The Coordination Structure

A two-level structure is chosen as the coordination framework. It consists of a object-level command level object controller and several task-level manipulator controllers. The former is used to guide the object to follow its desired trajectory and to perform other tasks such as maintaining a stable contact with the environment etc. The latter is utilized to control the manipulators to provide required force and, at the same time, to regulate the manipulators to track its own desired trajectory. These two subsystems are dynamically connected through a load distribution algorithm, which distributes in real-time the control force required by the object-level controller to each participating manipulator in certain optimal sense. Figure 4.2 shows the structure of the proposed coordination framework.

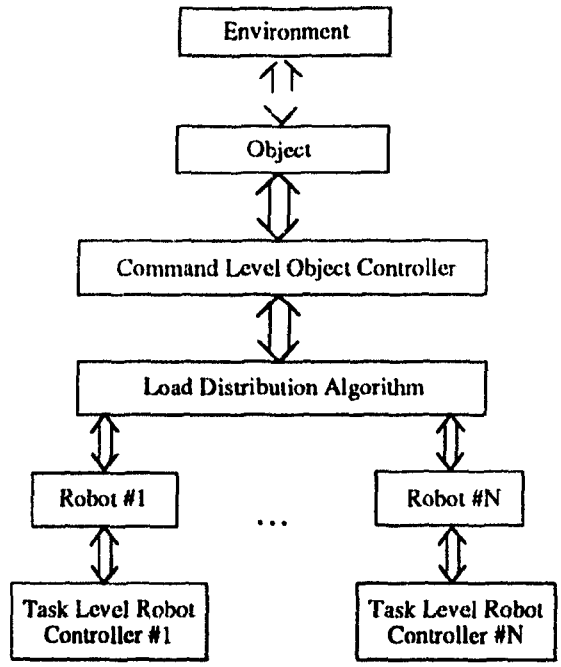


Figure 4.2: Proposed structure for coordination of a multi-robot system.

Command Level Object Controller

At the object-level, the choice of controller is quite flexible. Free motion controllers and force controllers can be applied for free motion and constrained motion of the object respectively. For example, a computed torque controller can be used for free motion control of the object. In this case $\mathbf{f}_e = 0$ and the control force is given by

$$\mathbf{f}_o = \mathbf{M}_o(\ddot{\mathbf{x}}_{od} + \mathbf{K}_{ov}\dot{\mathbf{e}}_o + \mathbf{K}_{op}\mathbf{e}_o) + \mathbf{B}_o\dot{\mathbf{x}}_o + \mathbf{M}_o\mathbf{g}_o \tag{4.41}$$

where $\mathbf{x}_{od}$ is the desired position and orientation of the frame at the mass center of the object (see (4.2)), $\mathbf{K}_p$ and $\mathbf{K}_v$ are the PD gains, and

$$\mathbf{e}_o = \mathbf{x}_{od} - \mathbf{x}_o$$

which is defined in a way similar to (4.7).

The desired force $\mathbf{f}_o$ is to be distributed to the coordinating robot as in (4.15) where $\mathbf{A}_k$, the distribution matrix can be obtained using one of the optimal load

distribution algorithms proposed in Section 4.3.

Task Level Robot Controller

Similarly, by using the computed torque method the task-level control for the i-th manipulator can be assigned as

$$\mathbf{f}_{ri} = \mathbf{H}_i(\ddot{\mathbf{x}}_{di} + \mathbf{K}_{vi}\dot{\mathbf{e}}_i + \mathbf{K}_{pi}\mathbf{e}_i) + \mathbf{C}_i\dot{\mathbf{x}}_i + \mathbf{g}_i + \Lambda_i\mathbf{f}_o \tag{4.42}$$

where $\mathbf{x}_{di}$ is the desired position and orientation of the gripper frame of the i-th robot which can be determined through $\mathbf{x}_{od}$, the object geometry, and present robot configuration, $\mathbf{K}_{pi}$ and $\mathbf{K}_{vi}$ are the PD gains of the computed torque controller for the i-th robot, and

$$\dot{\mathbf{e}} = \dot{\mathbf{x}}_{di} - \dot{\mathbf{x}}_i$$

$$\mathbf{e} = \mathbf{x}_{di} - \mathbf{x}_i$$

which are defined in a way similar to (4.7) and (4.11).

Applying object-level control (4.12) to (4.41), the error dynamics of the object is

$$\ddot{\mathbf{e}}_o + \mathbf{K}_{vo}\dot{\mathbf{e}}_o + \mathbf{K}_{po}\mathbf{e}_o = 0 \tag{4.43}$$

and, applying task-level control (4.42) to (4.17), the error dynamics of the ith manipulator can be obtained as

$$\ddot{\mathbf{e}}_i + \mathbf{K}_{vi}\dot{\mathbf{e}} + \mathbf{K}_{pi}\mathbf{e} = 0 \tag{4.44}$$

Equations (4.43) and (4.44) imply that $\mathbf{e}_0 \to 0$ and $\mathbf{e}_i \to 0$ as $t \to \infty$ for properly chosen $\mathbf{K}_{vo}$, $\mathbf{K}_{po}$, $\mathbf{K}_{vi}$, $\mathbf{K}_{pi}$.

Remarks

1. The proposed coordination structure is quite flexible in adopting different kinds of controllers. Force controllers can be employed to replace the computed

torque controller used in previous section. Force control algorithms can also be applied to the object-level command controller if there is a contact between the object and its environment. Obviously, a wrist force sensor must then be used for each manipulator to detect the contact force.

2. If the dynamic and kinematic parameters of the object are unknown, parameters of the object must be estimated before the proposed model-based control algorithm can be applied. The estimation methods similar to those proposed in [169, 150] can be used. Wrist force sensors are in this case needed. Object-level or task-level adaptive control schemes [130, 132] can also be applied to solve this problem.

3. *Internal forces* are defined as the forces generated by the participating manipulators which do not contribute to the dynamic motion of the object. Internal forces among the grippers through the commonly grasped object may cause deformation of the object due to pushing or pulling. Thus internal forces must be minimized or controlled. Under the proposed coordination structure, forces contributed by the manipulators to the dynamic motion of the object are determined through a load distribution algorithm. Therefore, the internal forces do not accumulate since the load distribution algorithm assigns *new* partitioned load to the manipulators at *every* control instant. In case a certain amount of internal forces is needed, the desired internal forces $\mathbf{f}_d^{int}$ should be assigned to each of the manipulators with different signs and magnitudes based on the number of manipulators, their current configurations, and their torque leftovers etc. Then the force assigned to the i-th manipulator, $\mathbf{f}_i^{int}$, can be directly added to the control equation (4.42) provided that the following condition is satisfied

$$\sum_{i=1}^{N} \mathbf{f}_i^{int} = 0.$$

4. In this thesis, it is assumed that the manipulators grasp the object rigidly. This assumption is impractical in many applications. Assume that the minimum force needed to overcome the static friction force between the object and the i-th robot gripper is f_{sf}, the stiffness coefficient of the object is c_o, and the stiffness of the robot gripper is much larger than c_o so that the gripper is absolutely rigid. The relative motion between the object and the gripper will not start until the distance between any two grippers i and j, d_{ij}, satisfies

$$d_{ij} - d_{normal} > \frac{f_{sf}}{c_o}, \quad \text{or} \quad d_{ij} - d_{normal} < -\frac{f_{sf}}{c_o} \tag{4.45}$$

where d_{ij} is the actual distance between the i-th and j-th gripper and d_{normal} is the normal distance when the two grippers will cause zero pulling or pushing forces. The first condition in (4.45) corresponds to the effect of pulling the object and the second to that of pushing the object. This relative motion will not stop until

$$|d_{ij} - d_{normal}| < \frac{f_{df}}{c_o} \tag{4.46}$$

where f_{df} is the minimum force needed to overcome the dynamic friction force between the object and the gripper. Condition (4.46) will be achieved very quickly once there is relative motion because the result of this relative motion is to make the grippers return to normal (zero-force) distance. So as far as the relative motion will not cause the gripper lose contact with the object, the relative motion will stop eventually and very quickly. It appears that the best cure for this problem is a precise position control algorithm to guarantee that none of the conditions in (4.45) will be met. On the other hand, if the rigid grip condition is removed, there is another question, that is how to estimate the position, orientation and velocity of the frame at the mass center of the object. Possible solutions include the use of a vision system. Further efforts are needed for the resolution of this problem and other problems caused by the removal of the assumption.

4.4.2 A Simulation Study

To illustrate the proposed coordination structure and to compare it with other existing methods, let us consider the robotic system shown in Figure 4.11, where two 3 DOF vertical planar manipulators are carrying a 10 kg point-mass plate from point A to point B along the upper half of a circle in 1 second. As the robots are redundant, the third links of both robots can be kept vertical to the ground during the entire motion. The method for redundant robot control reported in [170] is combined with the proposed methods to carry out the simulation.

Figure 4.12 shows the simulation results by the proposed method. Figure 4.12(a) shows the object tracking, while 4.12(b) shows the object tracking error, 4.12(c) the load sharing of robot 1, 4.12(d) load sharing of robot 2, 4.12(e) tracking of robot 1, 4.12(f) tracking error of robot 1, 4.12(g) tracking of robot 2, and 4.12(h) tracking error of robot 2. The simulation results indicate that the proposed method gives satisfactory results. The load distribution scheme described in Section 4.3 is used in the proposed method to partition the object dynamic load to the two robots. Figure 4.13 shows the simulation results on the comparison of internal forces by using different methods. Results obtained by simulation will be examined through implementation in our future study.

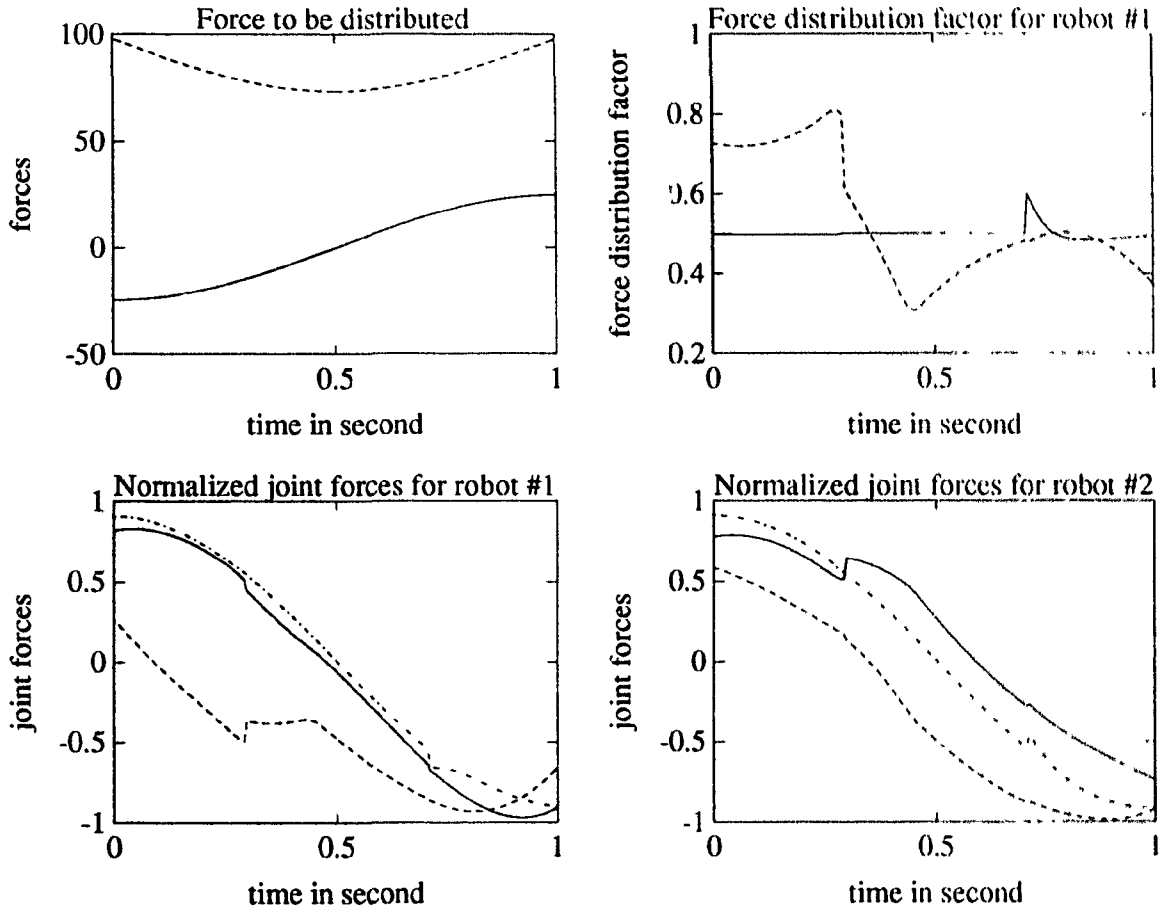


Figure 4.3: Load distribution using Algorithm 4.1. p=6, ε=15.

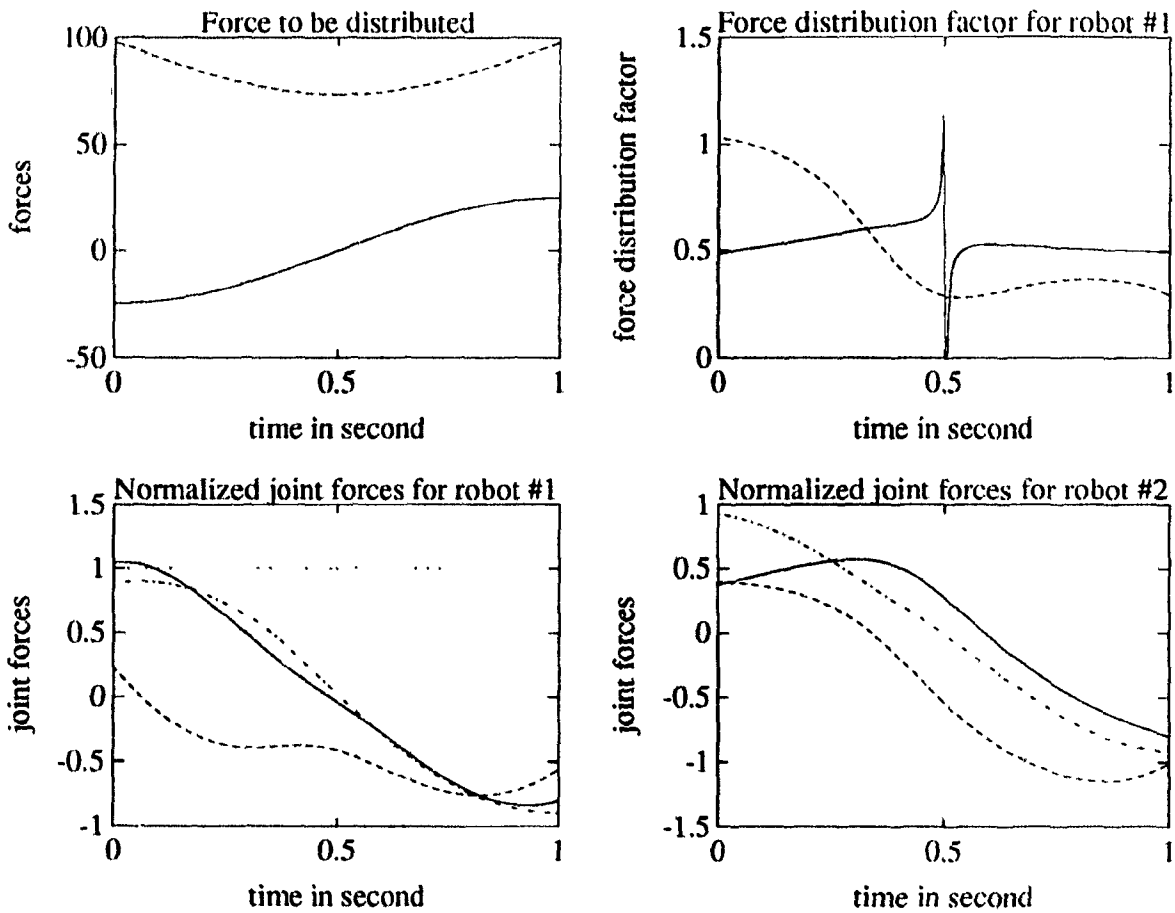


Figure 4.4: Load distribution using Algorithm 4.1. p=1, ε=15.

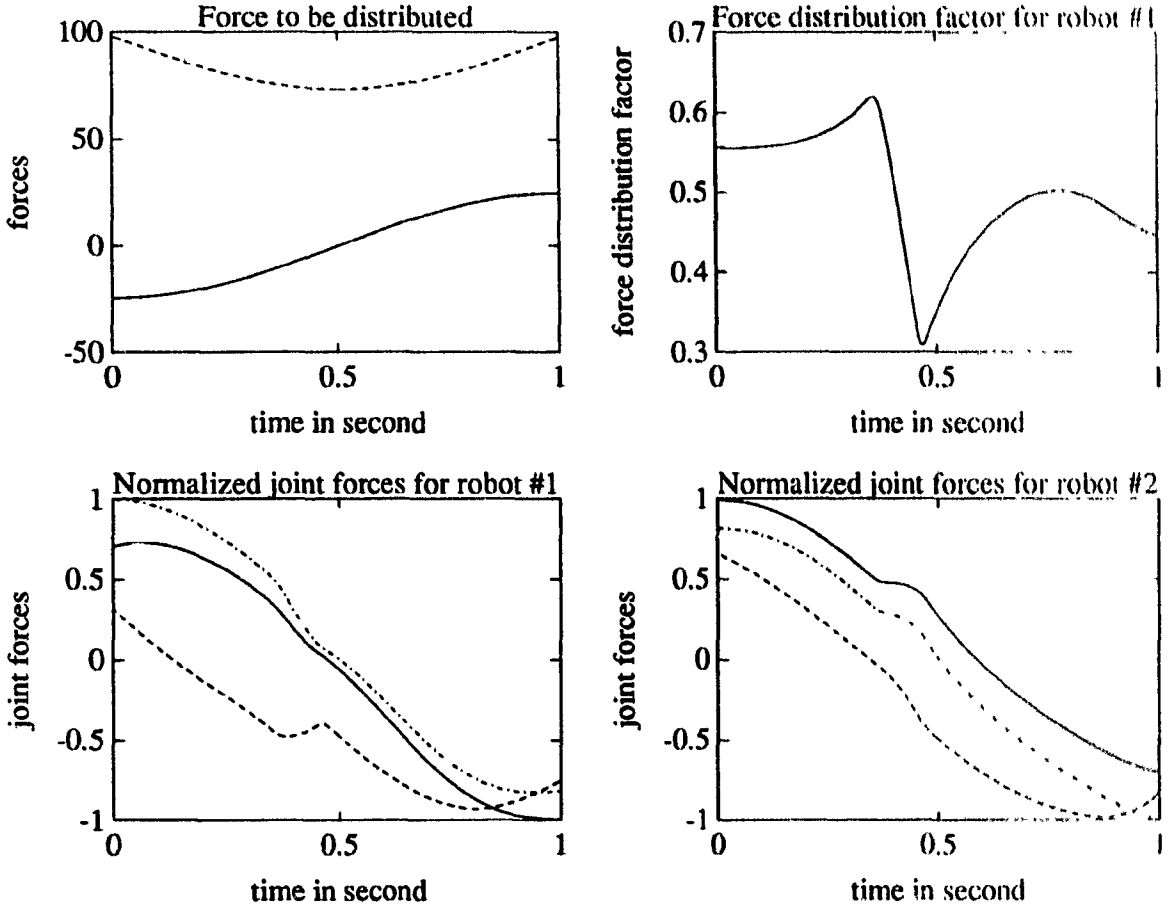


Figure 4.5: Load distribution using Algorithm 4.2. p=6.

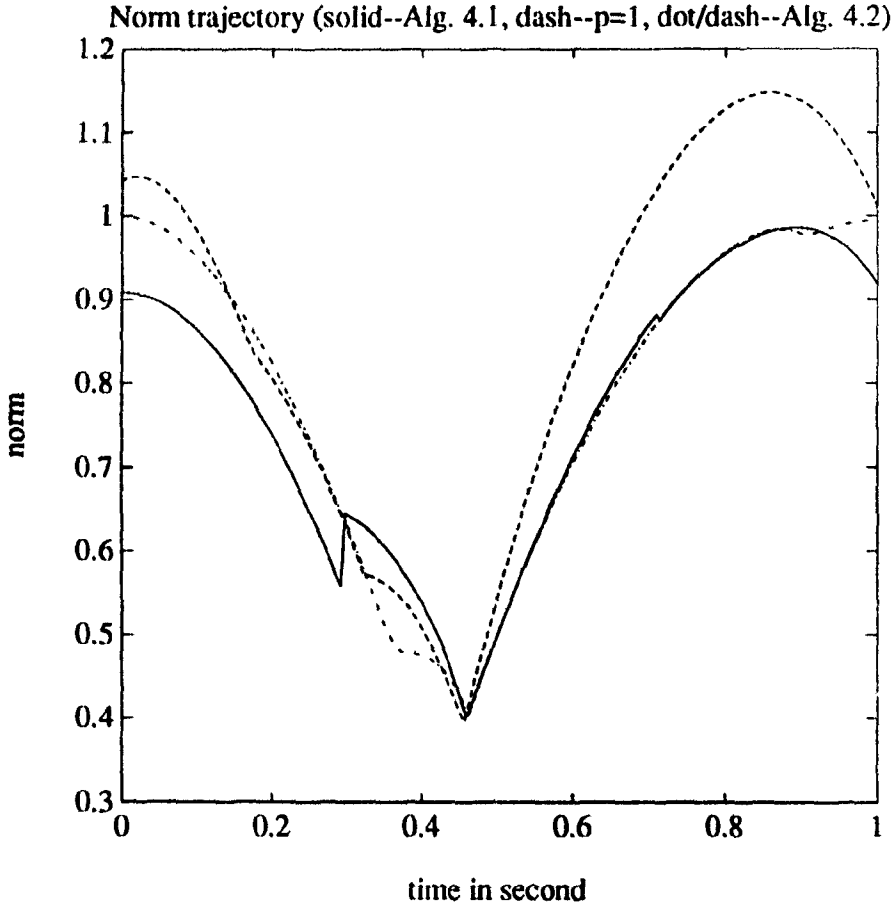


Figure 4.6: Norm trajectories of $\|\tau\|_\infty$ for the first three cases.

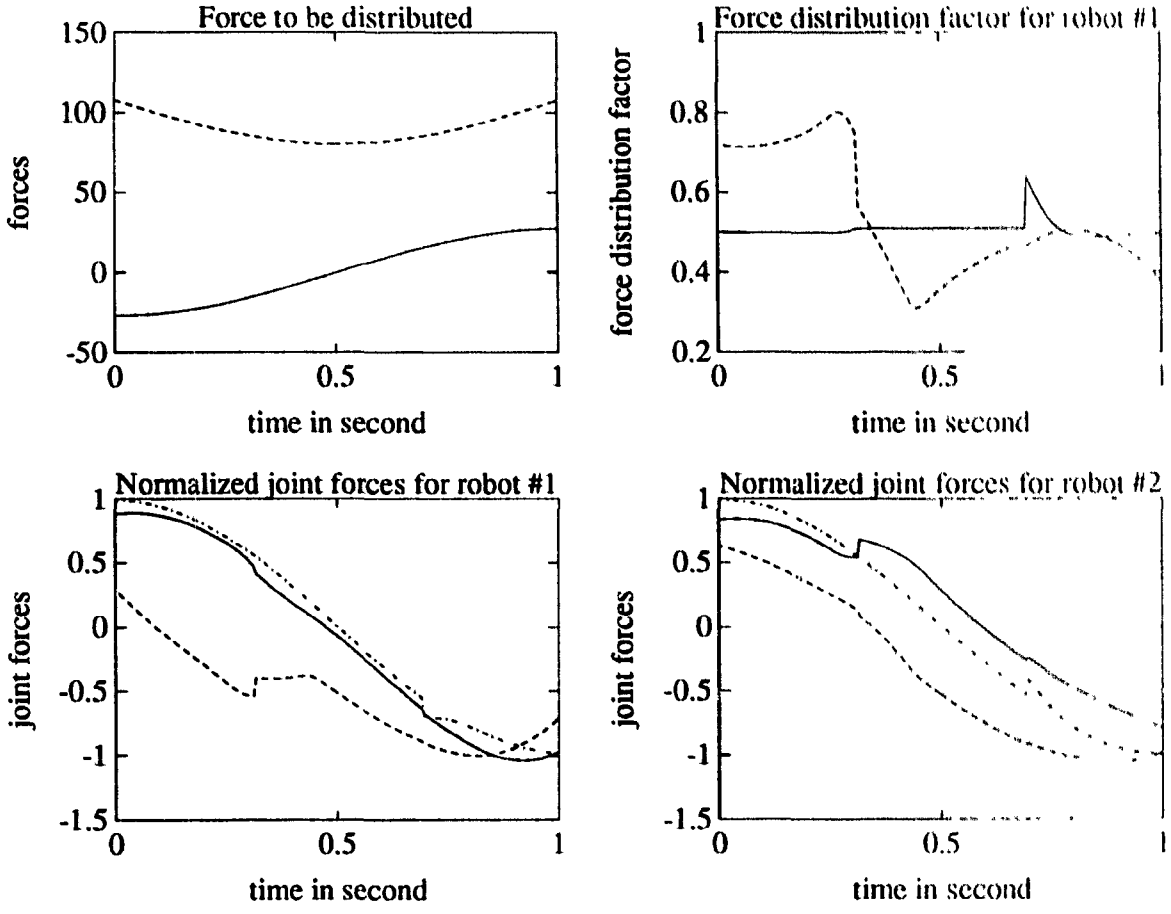


Figure 4.7: Load distribution using Algorithm 4.1. p=6, ε=15, m=5.5 kg.

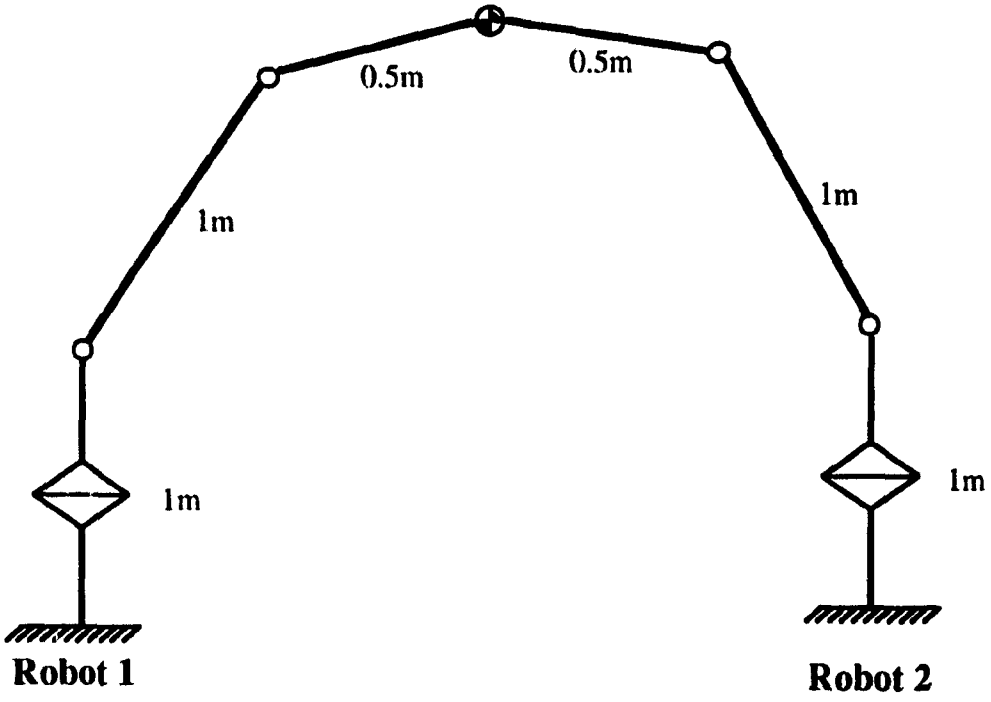


Figure 4.8: Two 3 DOF spatial coordinating robots.

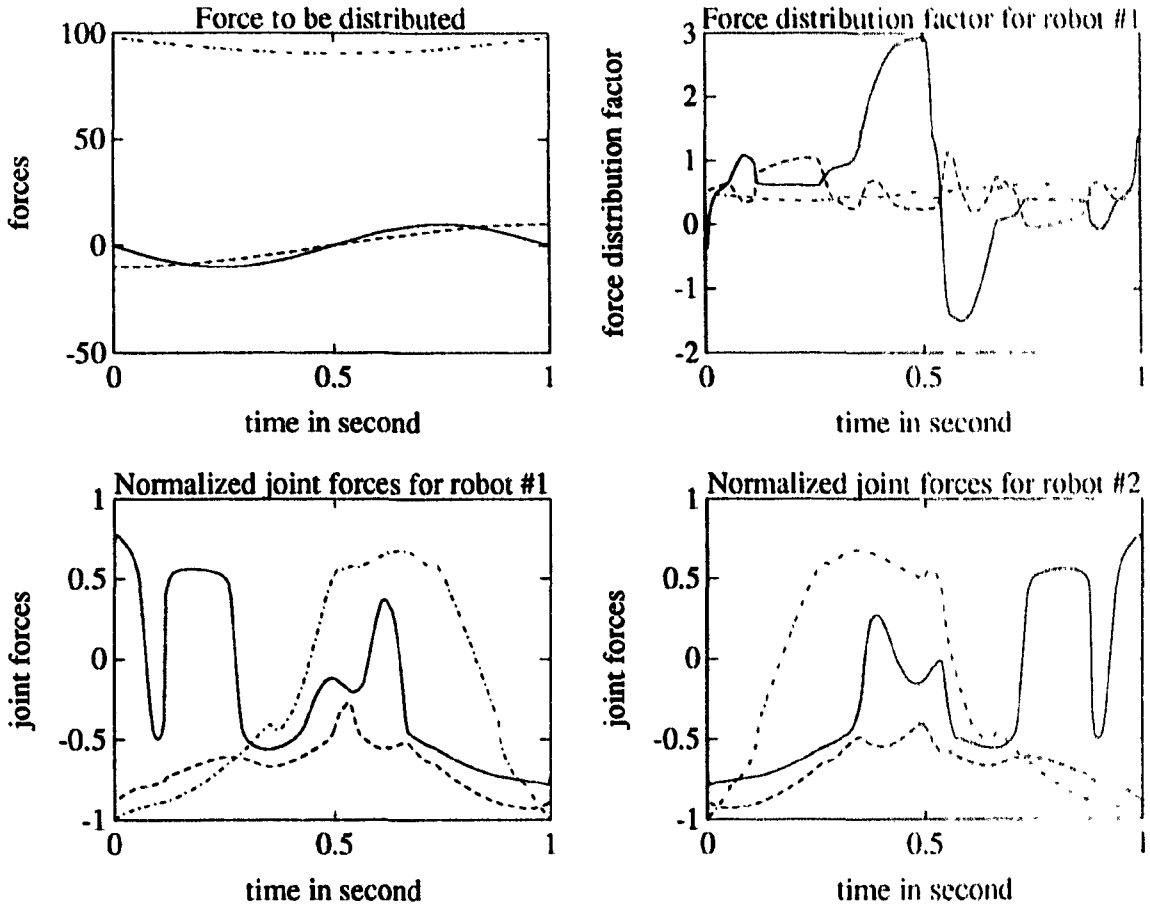


Figure 4.9: Load distribution using Algorithm 4.1 with p=6 and μ=0.02.

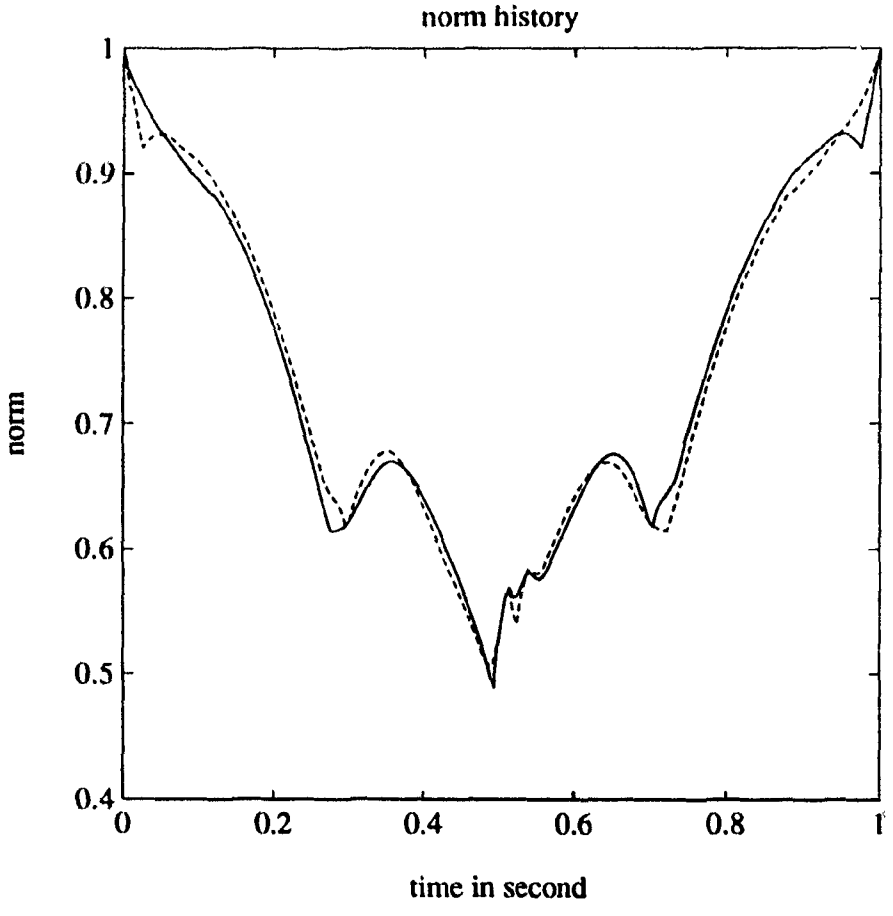


Figure 4.10: Trajectories of $\|\tau_1\|_\infty$ and $\|\tau_2\|_\infty$.

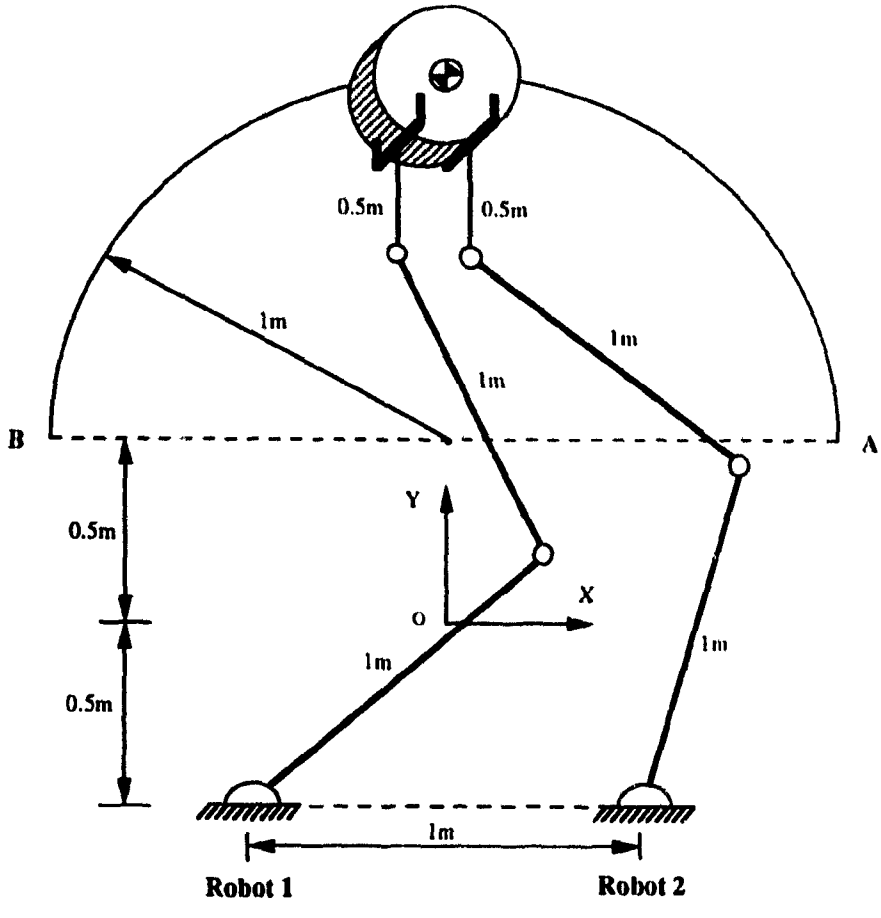


Figure 4.11: Two 3 DOF vertical planar coordinating robots.

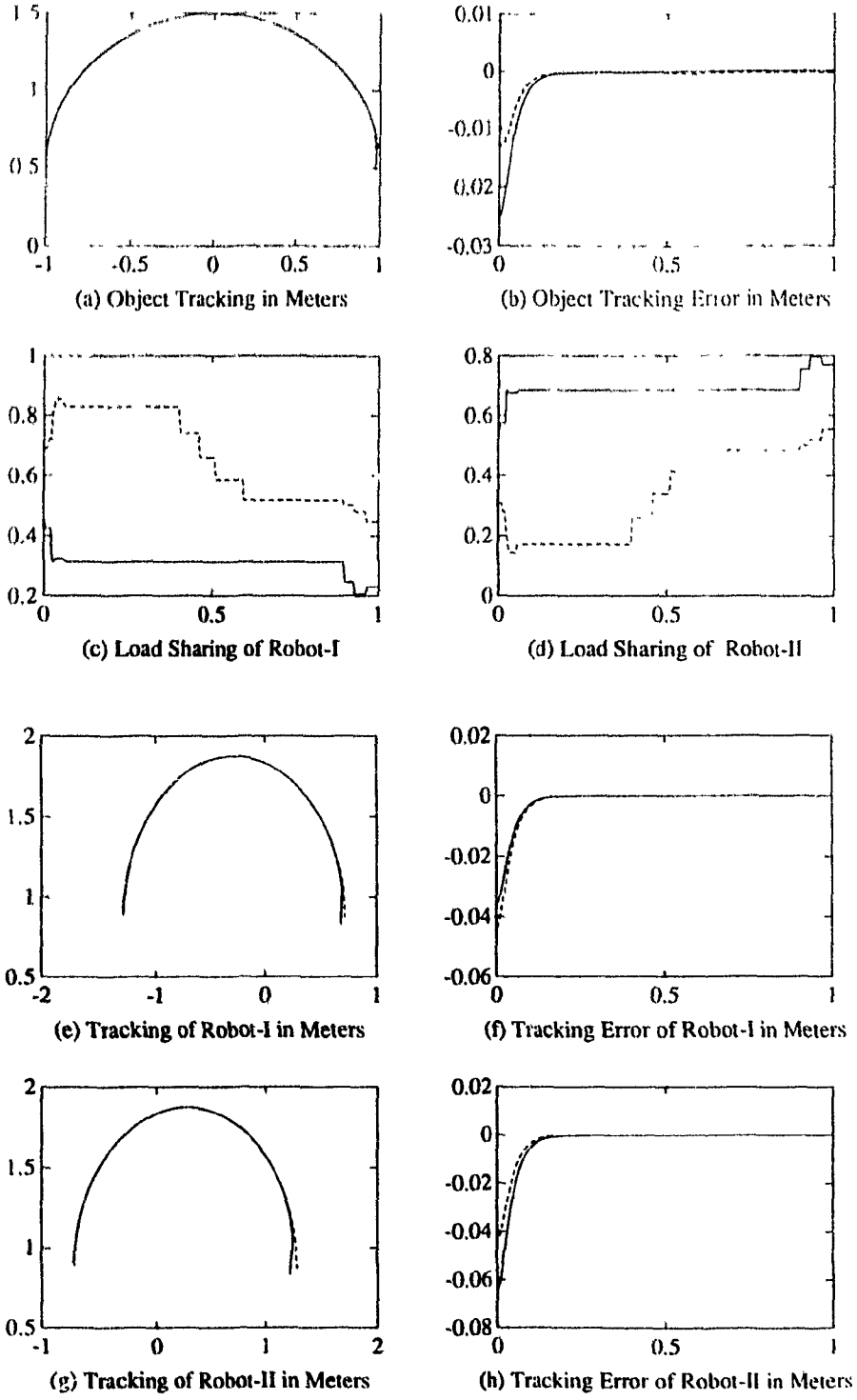
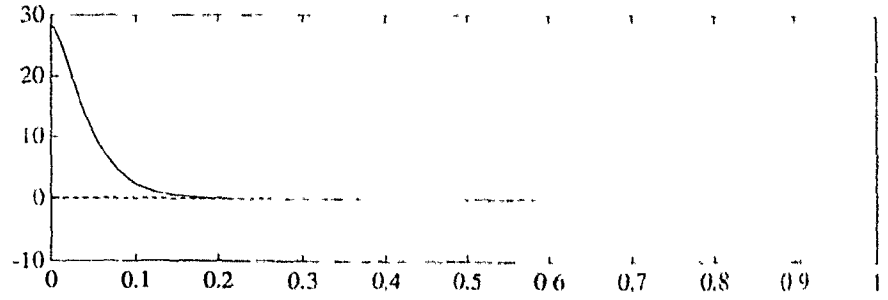
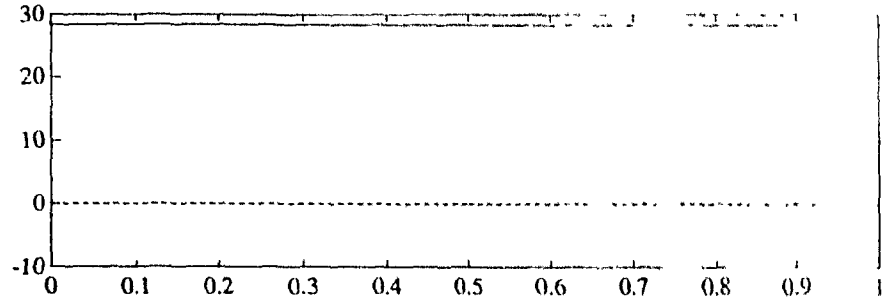


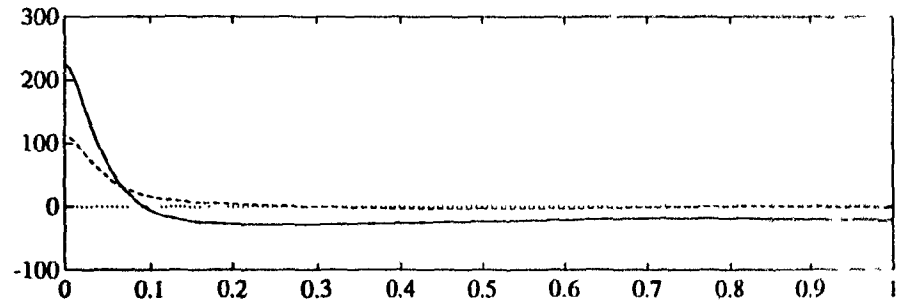
Figure 4.12: Coordination results.



(a) Internal Force Using Proposed Method With Object Stiffness=1000



(b) Internal Force Using Master Slave Method With Object Stiffness=1000



(c) Internal Force Using Parallel Control Method With Object Stiffness=1000

Figure 4.13: Internal Force Comparison.

Chapter 5

Summary and Future Work

5.1 Summary

In this thesis we have investigated several aspects of adaptive position and force control of robot manipulators. The results obtained in Chapters 2, 3, and 4 can be summarized as follows.

- Vector analysis techniques have been applied to both Lagrangian and Newton-Euler formulations to obtain the regressor dynamics formulations that have been found suitable in an adaptive control setting. More specifically, we have presented two algorithms for the evaluation of the regressor of an n DOF robot. The closed-form formulation reveals the structure of the regressor explicitly while the recursive formulation finds applications in real-time implementation of model-based position and force control algorithms. Satisfactory computation efficiency of the formulas, especially the recursive formula, has been demonstrated in our experimental studies.

- As an application of the regressor dynamics formulation, the popular Slotine-Li adaptive control algorithm [37] is modified and the modified Slotine-Li

adaptive control algorithm is then implemented on a PUMA 560 robot.

- Hogan's conventional impedance control has been extended to achieve stable execution of control tasks for the robotic system whose parameters are not precisely known. To achieve this goal, Target Impedance Reference Trajectory is introduced which makes it possible to inject two stable adaptive control algorithms [34, 37] into Hogan's conventional impedance control [63]. These two adaptive impedance control algorithms have been found stable. Implementation of the algorithm on a PUMA 560 robot indicates that the proposed adaptive impedance control algorithm can be used in real-time in position and contact-force control applications.

- A study on optimal load distribution and coordination of multiple robots has been conducted. Optimal load distribution schemes using a p-norm type optimization approach are proposed. The algorithms are then adopted to dynamically link the two-level controllers in a coordination framework. Simulation results are presented to show the performance of the proposed algorithms in comparison with several existing approaches.

5.2 Future Work

The work presented in this thesis can only be considered preliminary as many challenging and possibly more important problems have not been touched upon in this thesis. As the very last section of the thesis, we would like to propose a number of problems as possible future work in this field.

- Further work may be done on regressor dynamics in the following three directions.

1. To extend regressor formulations to the redundant and flexible robots and multi-robot systems;
2. To investigate the possibility of using simplified versions of the regressor dynamics so as to further improve the computational efficiency;
3. To design and fabricate a hardware version of the regressor formulation so as to expand its real-time applications.

- As a continuation of our work in force control, it is quite natural to seek adaptive force control strategies for redundant manipulators, and for multi-robot systems. By using the concept of TIRT, reliable algorithms may also be developed for contact-force tracking.

- In the field of multiple robot coordination, many problems need to be solved. Among other things, it is felt that particular attention needs to be given to the problem of motion coordination for multi-robot systems where the rigid grasp assumption is removed. Another problem of interest is the optimal load sharing among flexible manipulators. With the recent advances in modeling and control of flexible manipulators, the chances of finding a computationally feasible solution to the problem are better than ever before. Above all, solutions of this type will likely play a significant role in dynamic coordination of flexible robots.

Appendix A

Parameter Extraction via Vector Analysis

Throughout this appendix we assume that $m \in \mathbf{R}^{1\times 1}$, $\mathbf{p} = [p_x \quad p_y \quad p_z]^T$, $\mathbf{d} = [d_1 \quad d_2 \quad d_3]^T$, and $\mathbf{x} = [x_1 \quad x_2 \quad x_3]^T$.

1) Consider

$$m(\mathbf{J}_\omega \dot{\mathbf{q}} \times \mathbf{Rp})^T \mathbf{g} \tag{A.1}$$

where $\mathbf{J}_\omega \in \mathbf{R}^{3\times n}$, $\mathbf{R} \in R^{3\times 3}$ and $\mathbf{g} \in \mathbf{R}^{3\times 1}$ are given. Denoting

$$\boldsymbol{\theta}_1 = [mp_x \quad mp_y \quad mp_z]^T \tag{A.2}$$

and applying (2.41) to (A.1), we have

$$m(\mathbf{J}_\omega \dot{\mathbf{q}} \times \mathbf{Rp})^T = (\mathbf{g} \times \mathbf{J}_\omega \dot{\mathbf{q}})^T \mathbf{R}\boldsymbol{\theta}_1$$

If we denote $\mathbf{J}_\omega = [\mathbf{J}_{\omega_1} \quad \dots \quad \mathbf{J}_{\omega_n}]$ with $\mathbf{J}_{\omega_i} \in \mathbf{R}^{3\times 1}$ for $1 \le i \le n$, then

$$m(\mathbf{J}_\omega \dot{\mathbf{q}} \times \mathbf{Rp})^T \mathbf{g} = \dot{\mathbf{q}}^T \mathbf{Y}_e \boldsymbol{\theta}_1 \tag{A.3}$$

where

$$\mathbf{Y}_e = [\mathbf{g} \times \mathbf{J}_{\omega_1} \quad \mathbf{g} \times J_{\omega_2} \quad \dots \quad \mathbf{g} \times \mathbf{J}_{\omega_n}]^T \mathbf{R}$$

is an $n \times 3$ known matrix.

2) Consider the vector given by

$$m[\dot{\boldsymbol{\omega}} \times \mathbf{p} + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{p}) + \dot{\mathbf{v}}] \tag{A.4}$$

where m and $\mathbf{p}$ are parameters to be extracted. Define

$$\boldsymbol{\theta}_2 = [m \quad mp_x \quad mp_y \quad mp_z] \tag{A.5}$$

and note that

$$\dot{\boldsymbol{\omega}} \times \mathbf{p} \;=\; \boldsymbol{\Omega}(\dot{\boldsymbol{\omega}})\mathbf{p} \tag{A.6}$$

where $\boldsymbol{\Omega}(\dot{\boldsymbol{\omega}})$ is a skew symmetric matrix characterized by

$$\boldsymbol{\Omega}(\mathbf{x}) \;=\; \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix} \tag{A.7}$$

From (2.42) it follows that

$$\boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{p}) \;=\; (\mathbf{p}^T\boldsymbol{\omega})\boldsymbol{\omega} - (\boldsymbol{\omega}^T\boldsymbol{\omega})\mathbf{p} \;=\; [\mathbf{U} - \| \boldsymbol{\omega}\|^2\mathbf{I}]\mathbf{p}$$

where $\mathbf{U} = \boldsymbol{\omega}^T\boldsymbol{\omega}$ is the outer product of $\boldsymbol{\omega}$. Hence

$$m[\dot{\boldsymbol{\omega}} \times \mathbf{p} + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{p}) + \dot{\mathbf{v}}] \;=\; [\dot{\mathbf{v}} \quad \boldsymbol{\Omega}(\dot{\boldsymbol{\omega}}) + \mathbf{U} - \|\boldsymbol{\omega}\|^2\mathbf{I}]\boldsymbol{\theta}_2 \tag{A.8}$$

3) Let

$${}^c\mathbf{I}_i \;=\; \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz} \end{bmatrix} \tag{A.9}$$

be the unknown inertia matrix. Define

$$\boldsymbol{\theta}_3 \;=\; [I_{xx} \quad I_{yy} \quad I_{zz} \quad I_{xy} \quad I_{xz} \quad I_{yz}], \tag{A.10}$$

it can readily be verified that

$$^{c}\mathbf{I}_i\mathbf{d} = \mathbf{B}(\mathbf{d})\boldsymbol{\theta}_3 \tag{A.11}$$

where

$$\mathbf{B}(\mathbf{d}) = \begin{bmatrix} d_1 & 0 & 0 & -d_2 & -d_3 & 0 \\ 0 & d_2 & 0 & -d_1 & 0 & -d_3 \\ 0 & 0 & d_3 & 0 & -d_1 & -d_2 \end{bmatrix} \tag{A.12}$$

4) Consider the vector product

$$m\mathbf{p} \times (\dot{\mathbf{v}} + \mathbf{H}\mathbf{p}) \tag{A.13}$$

where $\mathbf{H}$ is a 3×3 known matrix. By (A.6), the first vector product in (A.13) is

$$m\mathbf{p} \times \dot{\mathbf{v}} = -\boldsymbol{\Omega}(\dot{\mathbf{v}})\boldsymbol{\theta}_2$$

If we write

$$\mathbf{H} = \begin{bmatrix} h_1^T \\ h_2^T \\ h_3^T \end{bmatrix}$$

and define

$$\tilde{\boldsymbol{\Psi}} = \begin{bmatrix} 0 & h_3^T & -h_2^T \\ -h_3^T & 0 & h_1^T \\ h_2^T & -h_1^T & 0 \end{bmatrix}_{3\times 9} \tag{A.14}$$

then

$$m\mathbf{p} \times \mathbf{H}\mathbf{p} = \begin{bmatrix} mp_x \\ mp_y \\ mp_z \end{bmatrix} \times \begin{bmatrix} h_1^T\mathbf{p} \\ h_2^T\mathbf{p} \\ h_3^T\mathbf{p} \end{bmatrix} \tag{A.15}$$

$$= \begin{bmatrix} h_3^T(mp_y\mathbf{p}) - h_2^T(mp_z\mathbf{p}) \\ h_1^T(mp_z\mathbf{p}) - h_3^T(mp_x\mathbf{p}) \\ h_2^T(mp_x\mathbf{p}) - h_3^T(mp_y\mathbf{p}) \end{bmatrix} = \tilde{\boldsymbol{\Psi}} \begin{bmatrix} mp_x\mathbf{p} \\ mp_y\mathbf{p} \\ mp_z\mathbf{p} \end{bmatrix}_{9\times 1} \tag{A.16}$$

By defining

$$\boldsymbol{\theta}_4 = m[p_x^2 \;\; p_y^2 \;\; p_z^2 \;\; p_x p_y \;\; p_x p_z \;\; p_y p_z]^T \tag{A.17}$$

(A.16) can be written as

$$m\mathbf{p} \times \mathbf{Hp} = \boldsymbol{\Psi}\boldsymbol{\theta}_4 \tag{A.18}$$

where

$$\boldsymbol{\Psi} = \tilde{\boldsymbol{\Psi}}[\mathbf{e}_1 \quad \mathbf{e}_5 \quad \mathbf{e}_9 \quad \mathbf{e}_2+\mathbf{e}_4 \quad \mathbf{e}_3+\mathbf{e}_7 \quad \mathbf{e}_6+\mathbf{e}_8] \tag{A.19}$$

and $\mathbf{e}_i$ is the ith column of the 9×9 identity matrix. Hence

$$m\mathbf{p} \times (\dot{\mathbf{v}} + \mathbf{Hp}) = \boldsymbol{\Phi}[\boldsymbol{\theta}_1^T \quad \boldsymbol{\theta}_4^T]^T \tag{A.20}$$

with

$$\boldsymbol{\Phi} = [-\boldsymbol{\Omega}(\dot{\mathbf{v}}) \quad \boldsymbol{\Psi}] \tag{A.21}$$

5) Finally, let us consider

$$m[\frac{\partial}{\partial \dot{\mathbf{q}}}(\mathbf{J}_v\dot{\mathbf{q}} + \mathbf{J}_\omega\dot{\mathbf{q}} \times {}_i^0\mathbf{Rp})]^T(\mathbf{J}_v\dot{\mathbf{q}} + \mathbf{J}_\omega\dot{\mathbf{q}} \times {}_i^0\mathbf{Rp}) \tag{A.22}$$

By (A.6),

$$\mathbf{J}_\omega\dot{\mathbf{q}} \times {}_i^0\mathbf{Rp} = -\boldsymbol{\Omega}({}_i^0\mathbf{Rp})\mathbf{J}_\omega\dot{\mathbf{q}}$$

So

$$\frac{\partial}{\partial \dot{\mathbf{q}}}(\mathbf{J}_v\dot{\mathbf{q}} + J_\omega\dot{\mathbf{q}} \times {}_i^0\mathbf{Rp}) = \mathbf{J}_v - \boldsymbol{\Omega}({}_i^0\mathbf{Rp})\mathbf{J}_\omega \tag{A.23}$$

Thus (A.22) can be written as

$$m[\mathbf{J}_v^T\mathbf{J}_v\dot{\mathbf{q}} + \mathbf{J}_v^T(\mathbf{J}_\omega\dot{\mathbf{q}} \times {}_i^0\mathbf{Rp}) - \mathbf{J}_\omega^T\boldsymbol{\Omega}^T({}_i^0\mathbf{Rp})\mathbf{J}_v\dot{\mathbf{q}} - \mathbf{J}_\omega^T\boldsymbol{\Omega}^T({}_i^0\mathbf{Rp})(\mathbf{J}_\omega\dot{\mathbf{q}} \times {}_i^0\mathbf{Rp})] \tag{A.24}$$

By (2.41),

$$\mathbf{J}_v^T(\mathbf{J}_\omega\dot{\mathbf{q}} \times {}_i^0\mathbf{Rp}) = \mathbf{p}^T\,{}_i^0\mathbf{R}(\mathbf{J}_v \times \mathbf{J}_\omega\dot{\mathbf{q}}) = (\mathbf{J}_v \times \mathbf{J}_\omega\dot{\mathbf{q}})^T\,{}_i^0\mathbf{Rp}$$

the second term in (A.24) is equal to

$$(\mathbf{J}_v \times \mathbf{J}_\omega\dot{\mathbf{q}})^T\,{}_i^0\mathbf{R}(m\mathbf{p}) \tag{A.25}$$

where

$$\mathbf{J}_v \times \mathbf{J}_\omega \dot{\mathbf{q}} \equiv [\mathbf{J}_{v_1} \times J_\omega \dot{\mathbf{q}} \quad \ldots \quad \mathbf{J}_{v_n} \times \mathbf{J}_\omega \dot{\mathbf{q}}] \tag{A.26}$$

Since $\boldsymbol{\Omega}(\mathbf{Rp})$ is skew symmetric, $-\boldsymbol{\Omega}({}^0_i\mathbf{Rp}) = \boldsymbol{\Omega}({}^0_i\mathbf{Rp})$. By (A.6) and (2.41), the third term in (A.24) can be written as

$$-\mathbf{J}_\omega^T(\mathbf{J}_v\dot{\mathbf{q}} \times {}^0_i\mathbf{Rp}) = (\mathbf{J}_v\dot{\mathbf{q}} \times \mathbf{J}_\omega)^T {}^0_i\mathbf{Rp} \tag{A.27}$$

where

$$\mathbf{J}_v\dot{\mathbf{q}} \times \mathbf{J}_\omega = [\mathbf{J}_v\dot{\mathbf{q}} \times \mathbf{J}_{\omega_1} \quad \ldots \quad \mathbf{J}_v\dot{\mathbf{q}} \times \mathbf{J}_{\omega_n}] \tag{A.28}$$

Using (A.6) and (2.42), the last term in (A.24) can be expressed as

$$\begin{aligned} \mathbf{J}_\omega^T {}^0_i\mathbf{Rp} \times (\mathbf{J}_\omega\dot{\mathbf{q}} \times {}^0_i\mathbf{Rp}) &= \mathbf{J}_\omega^T[\|\mathbf{p}\|^2\mathbf{J}_\omega\dot{\mathbf{q}} - (\mathbf{p}^T {}^0_i\mathbf{R}\mathbf{J}_\omega\dot{\mathbf{q}}){}^0_i\mathbf{Rp}] \\ &= \|\mathbf{p}\|^2\mathbf{J}_\omega^T\mathbf{J}_\omega\dot{\mathbf{q}} - (\mathbf{p}^T {}^0_i\mathbf{R}\mathbf{J}_\omega\dot{\mathbf{q}})\mathbf{J}_\omega^T {}^0_i\mathbf{Rp} \end{aligned} \tag{A.29}$$

From (A.24), (A.25), (A.27) and (A.29), it follows that if we define a 10-dimensional vector $\boldsymbol{\theta}_5$ as

$$\boldsymbol{\theta}_5 = m[1 \quad p_x \quad p_y \quad p_z \quad p_x^2 \quad p_y^2 \quad p_z^2 \quad p_xp_y \quad p_xp_z \quad p_yp_z] = [\boldsymbol{\theta}_2^T \quad \boldsymbol{\theta}_4^T]^T \tag{A.30}$$

then (A.22) can be expressed as

$$m[\frac{\partial}{\partial\dot{\mathbf{q}}}(\mathbf{J}_v\dot{\mathbf{q}} + \mathbf{J}_\omega\dot{\mathbf{q}} \times {}^0_i\mathbf{Rp})]^T(\mathbf{J}_v\dot{\mathbf{q}} + \mathbf{J}_\omega\dot{\mathbf{q}} \times {}^0_i\mathbf{Rp}) = \mathbf{W}_1(\mathbf{q},\dot{\mathbf{q}})\boldsymbol{\theta}_5 \tag{A.31}$$

where

$$\mathbf{W}_1(\mathbf{q},\dot{\mathbf{q}}) = [\mathbf{J}_v^T\mathbf{J}_v\dot{\mathbf{q}} \quad (\mathbf{J}_v \times \mathbf{J}_\omega\dot{\mathbf{q}} + \mathbf{J}_v\dot{\mathbf{q}} \times \mathbf{J}_\omega)^T {}^0_i\mathbf{R} \quad -\mathbf{D}] \tag{A.32}$$

with

$$\mathbf{D} = \tilde{\mathbf{D}} - [\mathbf{J}_\omega^T\mathbf{J}_\omega\dot{\mathbf{q}} \quad \mathbf{J}_\omega^T\mathbf{J}_\omega\dot{\mathbf{q}} \quad \mathbf{J}_\omega^T\mathbf{J}_\omega\dot{\mathbf{q}} \quad 0 \quad 0 \quad 0]_{n\times 6} \tag{A.33}$$

and $\tilde{\mathbf{D}}$ is determined by

$$(\mathbf{p}^T {}^0_i\mathbf{R}\mathbf{J}_\omega\dot{\mathbf{q}})\mathbf{J}_\omega^T {}^0_i\mathbf{Rp} = \tilde{\mathbf{D}}[p_x^2 \quad p_y^2 \quad p_z^2 \quad p_xp_y \quad p_xp_z \quad p_yp_z]^T \tag{A.34}$$

Appendix B

Strictly Positive Real Functions

The concept of Strictly Positive Realness (SPR) has been extensively studied for more than two decades [42, 171], and found numerous applications in network analysis, robustness analysis, nonlinear control and adaptive control. A proper rational function $f(s)$ is said to be SPR if the real part of $f(j\omega)$ is strictly greater than zero for all $\omega \in R$. As an example, consider second-order stable transfer function

$$f(s) = \frac{s+\psi}{s^2+k_v s+k_p} \tag{B.1}$$

where $k_v > 0$, $k_p > 0$ are given and ψ is a system parameter. It is easy to verify that $f(s)$ is SPR if $0 < \psi < k_v$. When a stable SPR rational function is realized in a state-space, the passive nature of such a transfer function provides a system-theoretic property that is more valuable than a single Lyapunov equation does. Indeed, if function $f(s)$ given by (B.1) with $\psi \in (0, k_v)$ is realized as $\sum = (\mathcal{A}, \mathcal{B}, \mathcal{C})$

where

$$\mathcal{A} = \begin{bmatrix} 0 & 1 \\ -k_p & -k_v \end{bmatrix}$$

$$\mathcal{B} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{B.2}$$

$$\mathcal{C} = [\psi \quad 1]$$

it can then be shown that there exist two positive definite matrices $\mathcal{P}$ and $\mathcal{Q}$ such that

$$\begin{aligned} \mathcal{A}^T\mathcal{P} + \mathcal{P}\mathcal{A} &= -\mathcal{Q} \\ \mathcal{P}\mathcal{B} &= \mathcal{C}^T \end{aligned} \tag{B.3}$$

The combination of the two equations in (B.3) plays a crucial role in deriving an adaptive control law for unconstrained motion control. Notice that one solution to (B.3) is given by

$$\mathcal{P} = \begin{bmatrix} k_p + \psi k_v & \psi \\ \psi & 1 \end{bmatrix}$$

$$\mathcal{Q} = \begin{bmatrix} 2\psi k_p & 0 \\ 0 & 2(k_v - \psi) \end{bmatrix}$$

both of which are positive definite with $\psi \in (0, k_v)$. A thorough discussion on time domain and frequent domain conditions for strict positive realness can be found in [171].

Bibliography

[1] S. Arimoto and F. Miyazaki, "Stability and robustness of PID feedback control for robot manipulators of sensory capability," in *Proc. of the Third Int. Symp. of Robotics Research*, 1985.

[2] S. Arimoto and M. Takegaki, "Asymptotic stability of feedback controls for robot manipulators," in *Proc. of the 1st IFAC Symposium on Robot Control*, 1985.

[3] J. J. Craig, *Introduction to Robotics.* Reading, MA: Addison-Wesley, second ed., 1989.

[4] R. P. Paul, "Modeling, trajectory calculation and servoing of a computer controlled arm," MEMO AIM-177, Stanford University Artificial Intelligence Laboratory, 1972.

[5] B. R. Markiewicz, "Analysis of the computed torque drive method and comparison with conventional position servo for a computer-controlled manipulator," Technical Memo 33-601, JPL NASA, 1973.

[6] A. K. Bejczy, "Robot arm dynamics and control," Technical Memo 33-669, JPL, NASA, 1974.

[7] R. A. Lewis, "Autonomous manipulation on a robot: summary of manipulator software functions," Technical Memo 33-679, JPL, NASA, 1974.

[8] E. Freund, "The structure of decoupled non linear systems," *Int. J. Control*, vol. 21, no. 3, pp. 443–450, 1975.

[9] E. Freund, "A non-linear control concept for computer-controlled manipulators," in *Proc. of the 7th Int. Symp. Intelligent Robots*, 1977.

[10] J. Y. S. Luh, M. W. Walker, and R. P. Paul, "Resolved-acceleration control of mechanical manipulators," *IEEE Trans. on Automatic Control*, vol. 25, no. 3, pp. 468–474, 1980.

[11] A. Liegois, A. Fournier, and M. J. Aldon, "Model reference control of high velocity industrial robots," in *Proc. of 1980 Joint Automatic Control Conf.*, 1980.

[12] P. Tomei, "Adaptive PD controller for robot manipulators," *IEEE Trans. on Robotics and Automation*, vol. 7, no. 4, pp. 565–570, 1991.

[13] W. S. Lu and Q. H. Meng, "An operational space adaptive control scheme for mechanical manipulators," in *Proc. of the 1989 Canadian Conf. on Electrical and Computer Engineering*, pp. 199–202, 1989.

[14] S. Dubowsky and D. T. DesForges, "The application of model-reference adaptive control to robotic manipulators," *ASME J. of Dynamic Systems, Measurement, and Control*, vol. 101, pp. 193–200, 1979.

[15] R. Horowitz and M. Tomizuka, "An adaptive control scheme for mechanical manipulators– compensation of nonlinearity and decoupling control," in *Proc. of ASME Winter Annual Meeting*, 1980.

[16] R. Horowitz and M. Tomizuka, "Discrete time model reference adaptive control of mechanical manipulators," *Compu. in Eng.*, vol. 2, pp. 107–112, 1982.

[17] M. Tomizuka, R. Horowitz, and Y. D. Landau, "On the use of model reference control techniques for mechanical manipulator," in *Proc. of 2nd IASTED Symp. Identification, Control and Robotics*, 1982.

[18] M. Tomizuka and R. Horowitz, "Model reference adaptive control of mechanical manipulators," in *Proc. of IFAC Conf. Adaptive Systems in Control and Signal Processing*, 1983.

[19] R. P. Anex and M. Hubbard, "Modeling and adaptive control of a mechanical manipulator," *ASME J. of Dynamic Systems, Measurement, and Control*, vol. 106, 1984.

[20] A. J. Koivo and T. M. Guo, "Adaptive linear controller for robotic manipulators," *IEEE Trans. on Automatic Control*, vol. 28, no. 2, pp. 162–171, 1983.

[21] G. G. Leininger, "Adaptive control of manipulators using self-tuning methods," in *Proc. of 1st ISRR*, 1983.

[22] P. G. Backes, G. G. Leininger, and C. H. Chung, "Real-time Cartesian coordinate hybrid control of a PUMA 560 manipulator," in *Proc. of IEEE Conf. Robotics and Automation*, 1985.

[23] C. S. G. Lee and B. H. Lee, "Resolved motion adaptive control for mechanical manipulators," in *Proc. of the 3rd Yale Workshop on Applications of Adaptive System Theory*, pp. 190–196, 1983.

[24] C. S. G. Lee, M. J. Chung, and B. H. Lee, "Adaptive control for robot manipulators in joint and Cartesian coordinates," in *Proc. of IEEE Conf. Robotics and Automation*, 1984.

[25] C. S. G. Lee and M. J. Chung, "An adaptive control strategy for mechanical manipulators," *IEEE Trans. on Automatic Control*, vol. 29, no. 9, 1984.

[26] C. S. G. Lee, M. J. Chung, and B. H. Lee, "An approach of adaptive control for robotic manipulators," *J. of Robotic Systems*, vol. 1, no. 1, pp. 27–57, 1984.

[27] B. K. Kim and K. G. Shin, "An adaptive model following control of industrial manipulators," *IEEE Trans. on Aerosp. Elect. Syst.*, vol. 19, pp. 805–814, 1983.

[28] S. Nicosia and P. Tomei, "Model reference adaptive control of robot manipulators," *Automatica*, vol. 20, no. 5, pp. 635–644, 1984.

[29] S. N. Singh, "Adaptive model following control of nonlinear robotic systems," *IEEE Trans. on Automatic Control*, vol. 30, no. 11, 1985.

[30] K. Y. Lim and E. Eslami, "Adaptive controller designs for robot manipulator systems using Lyapunov direct method," *IEEE Trans. on Automatic Control*, vol. 30, no. 12, 1985.

[31] T. C. Hsia, "Adaptive control of robot manipulators - a review," in *Proc. of IEEE Conf. Robotics and Automation*, pp. 183–189, 1986.

[32] Y. Stepanenko and J. Yuan, "Robust adaptive control of a class of nonlinear mechanical systems with unbounded and fast-varying uncertainties," *Automatica*, vol. 28, no. 2, pp. 265–276, 1992.

[33] R. Ortega and M. W. Spong, "Adaptive motion control of rigid robots: a tutorial," in *Proc. of the 27th Conf. on Decision and Control*, pp. 1575–1584, 1988.

[34] J. J. Craig, P. Hsu, and S. S. Sastry, "Adaptive control of mechanical manipulators," in *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp. 190–195, 1986.

[35] R. H. Middleton and G. C. Goodwin, "Adaptive computed torque control for rigid link manipulator," *System and Control Letters*, vol. 10, pp. 9–16, 1988.

[36] J. E. Slotine and W. Li, "Adaptive manipulator control: a case study," in *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp. 1392–1400, 1987.

[37] J. E. Slotine and W. Li, "Adaptive manipulator control: a case study," *IEEE Trans. on Automatic Control*, vol. 33, no. 11, pp. 995–1003, 1988.

[38] N. Sadegh and R. Horowitz, "Stability analysis of an adaptive controller for robot manipulators," in *Proc. of IEEE Conf. Robotics and Automation*, pp. 1223–1229, 1987.

[39] W. S. Lu and Q. H. Meng, "Impedance control with adaptation for robotic manipulations," *IEEE Trans. on Robotics and Automation*, vol. 7, no. 3, pp. 408–415, 1991.

[40] W. S. Lu and Q. H. Meng, "Authors' reply to "comments on 'impedance control with adaptation for robotic manipulations' "," *IEEE Trans. on Robotics and Automation*, vol. 7, no. 6, p. 881, 1991.

[41] J. J. Craig, *Adaptive Control of Mechanical Manipulators.* Reading, MA: Addison-Wesley, 1988.

[42] B. D. O. Anderson and S. Vongpanitlerd, *Network Synthesis: A State Space Approach.* Englewood Cliffs, N.J.: Prentice-Hall, 1973.

[43] R. C. Groome, *Force feedback steering of a teleoperator system.* Master's thesis, MIT, 1972.

[44] D. E. Whitney, "The mathematics of coordinated control of prosthetic arms and manipulators," *ASME J. of Dynamic Systems, Measurement, and Control*, vol. 94, pp. 303–309, 1972.

[45] R. Bolles and R. P. Paul, "The use of sensory feedback in a programmable assembly system," Memo AIM-220, Stanford University Artificial Intelligence Laboratory, 1973.

[46] J. L. Nevins and D. E. Whitney, "Computer-controlled assembly," *Scientific American*, vol. 238, pp. 62–74, 1978.

[47] S. H. Drake, "Using compliance in lieu of sensory feedback for automatic assembly," Report T-675, The C. S. Draper Laboratory, 1977.

[48] R. Paul and B. Shimano, "Compliance and control," in *Proc. of Joint Automatic Control Conf.*, pp. 694–699, 1976.

[49] D. E. Whitney, "Historical perspective and state of the art in robot force control," *Int. J. of Robotics Research*, vol. 6, pp. 3–14, 1987.

[50] C. S. G. Lee, R. C. Gonzalez, and K. S. Fu, *Tutorial on Robotics.* IEEE Computer Society Press, second ed., 1986.

[51] M. Brady, J. M. Hollerbach, T. L. Johnson, T. Lozano-Perez, and M. T. Mason, *Robot Motion: Planning and Control.* Cambridge, MA: The MIT Press, 1982.

[52] O. Khatib, J. J. Craig, and T. Lozano-Perez, *The Robotics Review 1.* Cambridge, MA: The MIT Press, 1989.

[53] C. H. An, C. G. Atkeson, and J. M. Hollerbach, *Model-Based Control of a Robot Manipulator.* Cambridge, MA: The MIT Press, 1988.

[54] N. Hogan, "Mechanical impedance control in assistive devices and manipulators," in *Proc. of Joint Automatic Control Conf.*, pp. TA–10B, 1980.

[55] N. Hogan, "Impedance control of a robotic manipulator," in *Winter Annual Meeting of the ASME*, 1981.

[56] N. Hogan, "Programmable impedance control of industrial manipulators," in *Proc. of the Conf. on CAD/CAM in Mechanical Engineering*, 1982.

[57] N. Hogan and S. L. Cotter, "Cartesian impedance control of a nonlinear manipulator," in *Robotics Research and Advanced Applications*, (W. J. Book, ed.), ASME Press, 1982.

[58] N. Hogan, "Adaptive control of mechanical impedance by coactivation of antagonist muscles," *IEEE Trans. on Automatic Control*, vol. 29, no. 8, pp. 681–690, 1984.

[59] N. Hogan, "Impedance control: an approach to manipulation," in *Proc. of 1984 American Control Conf.*, pp. 304–313, 1984.

[60] N. Hogan, "Impedance control: an approach to manipulation: Part I - theory, Part II - implementation, part III - applications," *ASME J. of Dynamic Systems, Measurement, and Control*, vol. 107, no. 1, pp. 1–24, 1985.

[61] J. R. Andrews and N. Hogan, "Impedance control as a framework for implementing obstacle avoidance in a manipulator," in *Control of Manufacturing and Robotic Systems*, (D. E. Hardt and W. J. Book, eds.), pp. 243–251, ASME Press, 1983.

[62] N. Hogan, "On the stability of manipulators performing contact tasks," *IEEE J. of Robotics and Automation*, vol. 4, pp. 677–686, 1988.

[63] N. Hogan, "Stable execution of contact tasks using impedance control," in *Proc. of IEEE Int. Conf. on Robotics and Automation*, 1987.

[64] E. Colgate and N. Hogan, "An analysis of contact instability in terms of passive physical equivalents," in *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp. 404–409, 1989.

[65] F. A. Mussa-Ivaldi and N. Hogan, "Solving kinematic redundancy with impedance control: a class of integrable pseudoinverses," in *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp. 283–288, 1989.

[66] A. Sharon, N. Hogan, and D. E. Hardt, "Control design in the physical domain (application to robot impedance control)," in *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp. 552–559, 1989.

[67] P. J. McKerrow, *Introduction to Robotics.* Reading, MA: Addison-Wesley, 1991.

[68] D. A. Lawrence, "Impedance control stability properties in common implementations," in *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp. 1185-1190, 1988.

[69] R. J. Anderson and M. W. Spong, "Hybrid impedance control of robotic manipulators," *IEEE J. of Robotics and Automation*, vol. 4, no. 5, pp. 549–556, 1988.

[70] R. Kelly, R. Carelli, M. Amestegul, and R. Ortega, "On adaptive impedance control of robot manipulators," in *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp. 572–577, 1989.

[71] R. Carelli, R. Kelly, and R. Ortega, "Adaptive force control of robot manipulators," *Int. J. of Control*, vol. 52, no. 1, pp. 37–54, 1990.

[72] W. S. Lu and Q. H. Meng, "Adaptive impedance control of mechanical manipulators," in *Robotics and Manufacturing: Recent Trends in Research, Educa-*

tion, and Applications, (M. Jamshidi and M. Saif, eds.), pp. 329–335, ASME Press, 1990. Volume 3.

[73] Q. H. Meng and W. S. Lu, "Adaptive contact force control of redundant robots using manipulator regressor dynamics," in *Proc. of the Second Annual IRIS/Precarn Conf.*, 1992.

[74] Z. Lu and A. A. Goldenberg, "Implementation of robust impedance and force control using VSS theory," in *Robotics and Manufacturing: Recent Trends in Research, Education, and Applications*, (M. Jamshidi and M. Saif, eds.), pp. 279–285, ASME Press, 1990. Volume 3.

[75] C. Abdallah, R. Jordan, and B. Horne, "A positive-real impedance control design for robots," in *Robotics and Manufacturing: Recent Trends in Research, Education, and Applications*, (M. Jamshidi and M. Saif, eds.), pp. 323–328, ASME Press, 1990. Volume 3.

[76] M. Raibert and J. Craig, "Hybrid position/force control of manipulators," *ASME J. of Dynamic Systems, Measurement, and Control*, vol. 102, pp. 126–133, 1981.

[77] M. T. Mason, "Compliance and force control for computer controlled manipulators," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 11, no. 6, pp. 418–432, 1981.

[78] O. Khatib, "A unified approach for motion and force control of robot manipulators: the operational space formulation," *IEEE J. of Robotics and Automation*, vol. 3, no. 1, pp. 43–53, 1987.

[79] H. Zhang and R. P. Paul, "Hybrid control of robot manipulators," in *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp. 602–607, 1985.

[80] H. West and H. Asada, "A method for the design of hybrid position/force controllers for manipulators constrained by contact with the environment," in *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp. 251–259, 1985.

[81] H. West and H. Asada, "A method for the control of robot arms constrained by contact with the environment," in *Proc. of 1985 American Control Conf.*, pp. 383–386, 1985.

[82] T. Yoshikawa, "Dynamic hybrid position/force control of robot manipulators - description of hand constraints and calculation of joint driving force," *IEEE J. of Robotics and Automation*, vol. 3, no. 5, pp. 386–392, 1987.

[83] T. Yoshikawa, T. Sugie, and M. Tanaka, "Dynamic hybrid position/force control of robot manipulators - controller design and experiment," *IEEE J. of Robotics and Automation*, vol. 4, no. 6, pp. 699–705, 1988.

[84] J. K. Mills, "Hybrid control: a constrained motion perspective," *J. of Robotic Systems*, vol. 8, no. 2, pp. 135–158, 1991.

[85] D. L. Wedel and G. N. Saridis, "An experiment in hybrid position/force control of a six dof revolute manipulator," in *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp. 1638–1642, 1988.

[86] J. E. Slotine and W. Li, "Adaptive strategies in constrained manipulation," in *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp. 595–601, 1987.

[87] W. Li and J. E. Slotine, "A unified approach to compliant motion control," in *Proc. of 1989 American Control Conf.*, pp. 1944–1949, 1989.

[88] G. Niemeyer and J. E. Slotine, "Computational algorithms for adaptive compliant motion," in *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp. 566–571, 1989.

[89] H. Seraji, "Adaptive force and position control of manipulators," *J. of Robotic Systems*, vol. 4, no. 4, pp. 551–578, 1987.

[90] L. Mo and M. M. Bayoumi, "Hybrid adaptive control for robot manipulators," *Systems and Control Letters*, vol. 15, pp. 405–409, 1990.

[91] J. C. H. Chung and G. G. Leininger, "Task-level adaptive hybrid manipulator control," *Int. J. of Robotics Research*, vol. 9, no. 3, pp. 63–73, 1990.

[92] C. Y. Kuo and S. T. Wang, "Nonlinear robust hybrid control of robot manipulators," *ASME J. of Dynamic Systems, Measurement, and Control*, vol. 112, no. 1, pp. 48–54, 1990.

[93] A. De Luca, C. Manes, and G. Ulivi, "Robust hybrid dynamic control of robot arms," in *Proc. of the 28th Conf. on Decision and Control*, pp. 2641–2646, 1989.

[94] R. P. Paul, "Problems and research areas associated with the hybrid control of force and displacement," in *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp. 1966–1971, 1987.

[95] J. Duffy, "The fallacy of modern hybrid control theory that is based on 'orthogonal complements' of twist and wrench spaces," *J. of Robotic Systems*, vol. 7, no. 2, pp. 139–144, 1990.

[96] J. K. Salisbury, "Active stiffness control of a manipulator in cartesian coordinates," in *Proc. of the 19th Conf. on Decision and Control*, pp. 95–100, 1980.

[97] D. E. Whitney, "Quasi-static assembly of compliantly supported rigid parts," *ASME J. of Dynamic Systems, Measurement, and Control*, vol. 104, pp. 65–77, 1982.

[98] H. Hanafusa and H. Asada, "Stable prehension by a robot hand with elastic fingers," in *Proc. of the 7th Int. Symposium on Industrial Robots*, pp. 361–368, 1977.

[99] H. Hanafusa and H. Asada, "A robot hand with elastic fingers and its application to assembly process," in *Proc. of IFAC Symposium on Information and Control Problems in Manufacturing Technology*, pp. 127–138, 1977.

[100] H. A. E. Maraghy and B. Johns, "An investigation into the compliance of SCARA robots. Part I: analytical model, Part II: experimental and numerical validation," *ASME J. of Dynamic Systems, Measurement, and Control*, vol. 110, pp. 18–30, 1988.

[101] J. DeSchutter and H. V. Brussel, "Compliant robot motion I. a formalism for specifying compliant motion tasks, II. a control approach based on external control loops," *Int. J. of Robotics Research*, vol. 7, no. 4, pp. 3–33, 1988.

[102] H. Kazerooni, T. B. Sheridan, and P. K. Houpt, "Robust compliant motion for manipulators, Part I: the fundamental concepts of compliant motion, Part II: design method," *IEEE J. of Robotics and Automation*, vol. 2, no. 2, pp. 83–105, 1986.

[103] H. Kazerooni, "Compliant motion control for robot manipulators," *Int. J. of Control*, vol. 49, no. 3, pp. 745–760, 1989.

[104] H. Kazerooni, "On the robot compliant motion control," *ASME J. of Dynamic Systems, Measurement, and Control*, vol. 111, pp. 416–425, 1989.

[105] H. Kazerooni, B. J. Waibel, and S. Kim, "On the stability of robot compliant motion control: theory and experiments," *ASME J. of Dynamic Systems, Measurement, and Control*, vol. 112, pp. 417–426, 1990.

[106] H. Kazerooni, K. G. Bouklas, and J. Guo, "Theory and experiments on the compliant control of redundant robot manipulators," *ASME J. of Dynamic Systems, Measurement, and Control*, vol. 112, pp. 653–660, 1990.

[107] K. G. Shin and C. Lee, "Compliant control of robotic manipulators with resolved acceleration," in *Proc. of the 24th Conf. on Decision and Control*, pp. 350–357, 1985.

[108] J. K. Salisbury and J. Craig, "Articulated hands: force control and kinematic issues," *Int. J. of Robotics Research*, vol. 1, no. 1, pp. 4–17, 1982.

[109] H. McClamroch and D. Wang, "Feedback stabilization and tracking of constrained robots," *IEEE Trans. on Automatic Control*, vol. Robotics and Automation33, no. 5, pp. 419–426, 1988.

[110] H. McClamroch and D. Wang, "Linear feedback control of position and contact force for a nonlinear constrained mechanism," *ASME J. of Dynamic Systems, Measurement, and Control*, vol. 112, pp. 640–645, 1990.

[111] O. Khatib and J. L. Maitre, "Dynamic control of manipulators operating in a complex environment," in *Proc. of the 3rd Int. CISM-IFToMM Symposium*, pp. 267–282, 1978.

[112] O. Khatib, "Real time obstacle avoidance for manipulators and mobile robots," *Int. J. of Robotics Research*, vol. 5, no. 1, pp. 90–99, 1986.

[113] D. S. Necsulasau, "Robot force control," Technical Memo DSM 89-03, Canadian Space Agency, 1989.

[114] C. H. Wu and R. Paul, "Manipulator compliance based on joint torque control," in *Proc. of the 19th Conf. on Decision and Control*, pp. 88–94, 1980.

[115] J. Y. S. Luh, W. D. Fisher, and R. Paul, "Joint torque control by a direct feedback for industrial robots," *IEEE Trans. on Automatic Control*, vol. Robotics and Automation28, pp. 153–160, 1983.

[116] L. Liu, Y. Han, R. Lingarkar, N. K. Sinha, and M. A. Elbestawi, "On adaptive force/motion control of constrained robots," in *Proc. of 15th Annual Conf. IEEE Industrial Electronics*, pp. 433–438, 1989.

[117] J. J. E. Slotine and W. P. Li, "On the adaptive control of robot manipulators," *Int. J. Robotics Research*, vol. 6, no. 3, 1987.

[118] Q. H. Meng and W. S. Lu, "Implementations of model-based adaptive position and force control algorithms on a puma 560 robot," Proc. of IASTED Int. Conf. on Control and Robotics, 1992.

[119] Q. H. Meng and W. S. Lu, "Experimental evaluation of the effect of dynamics on robot controllers," accepted for presentation at the Canadian Conf. on Electrical and Computer Engineering, 1992.

[120] T. Ishia, "Force control in coordination of two arms," in *Proc. of 5th Int'l Conf. on A.I.*, pp. 717–722, 1977.

[121] C. Alford and S. Belyeu, "Coordinated control of two robot arms," in *Proc. of IEEE Int. Conf. on Robotics and Automation*, 1984.

[122] Y. F. Zheng and J. Y. S. Luh, "Joint torques for control of two coordinated moving robots," in *Proc. of IEEE Int. Conf. on Robotics and Automation*, 1986.

[123] Y. F. Zheng and J. Y. S. Luh, "A real-time distributed computer system for coordinated motion control of two industrial robots," in *Proc. of IEEE Int. Conf. on Robotics and Automation*, 1987.

[124] T. J. Tarn, A. K. Bejczy, and X. Yun, "Design of dynamic control of two cooperating robot arms: closed chain formulation," in *Proc. of IEEE Int. Conf. on Robotics and Automation*, 1987.

[125] T. E. Alberts and D. I. Soloway, "Force control of multi-arm robot system," in *Proc. of IEEE Int. Conf. on Robotics and Automation*, 1988.

[126] Y. Nakamura, K. Nagai, and T. Yoshikawa, "Dynamics and stability in coordination of multiple robotic mechanisms," *Int. J. of Robotics Research*, vol. 8, no. 2, 1989.

[127] P. Hsu, "Control of multi-manipulator systems - trajectory tracking, load distribution, internal force control, and decentralized architecture," in *Proc. of IEEE Int. Conf. on Robotics and Automation*, 1989.

[128] J. T. Wen, "Motion and force control of multiple cooperative manipulators," in *Proc. of IEEE Int. Conf. on Robotics and Automation*, 1989.

[129] S. A. Schneider and R. H. Cannon, Jr., "Object impedance control for cooperative manipulation: theory and experimental results," in *Proc. of IEEE Int. Conf. on Robotics and Automation*, 1989.

[130] T. Hu and A. A. Goldenberg, "An adaptive approach to motion and force control of multiple coordinated robot arms," in *Proc. of IEEE Int. Conf. on Robotics and Automation*, 1989.

[131] M. E. Pittelkau, "Adaptive load-sharing force control for two-arm manipulators," in *Proc. of IEEE Int. Conf. on Robotics and Automation*, 1988.

[132] M. W. Walker, D. Kim, and J. Dionise, "Adaptive coordinated motion control of two manipulator arms," in *Proc. of IEEE Int. Conf. on Robotics and Automation*, 1989.

[133] R. Mehrotra and M. R. V. (ed), *Multirobot Systems*. Los Alamitos, CA: IEEE Computer Society Press, 1990.

[134] D. E. Orin and S. Y. Oh, "Control of force distribution in robotic manipulator mechanism containing closed kinematics chains," *ASME J. Dyna. Syst. Meas. Contr.*, vol. 102, pp. 134–41, 1981.

[135] Y. F. Zheng and J. Y. S. Luh, "Optimal load distribution for two industrial robots handling a single object," *ASME J. Dyna. Syst. Meas. Contr.*, vol. 111, pp. 232–237, 1989.

[136] Y. F. Zheng and J. Y. S. Luh, "Load distribution between two coordinating robots by nonlinear programming," in *Proc. of American Control Conf.*, 1988.

[137] Y. Nakamura, K. Nagai, and T. Yoshikawa, "Mechanics of coordinative manipulation by multiple robotic mechanisms," in *Proc. of IEEE Int. Conf. Robotics and Automation*, 1987.

[138] V. Kumar and K. J. Waldron, "Force distribution in closed kinematic chains," in *Proc. of IEEE Int. Conf. Robotics and Automation*, 1988.

[139] F. T. Cheng and D. E. Orin, "Efficient algorithm for optimal force distribution in multiple-chain robotic systems – the compact-dual lp method," in *Proc. of IEEE Int. Conf. Robotics and Automation*, 1989.

[140] W. S. Lu and Q. H. Meng, "On optimal force distribution of coordinating manipulators," *Int. J. of Robotics and Automation*, p. To appear in No. 2 issue, 1992.

[141] W. S. Lu and Q. H. Meng, "A load distribution scheme for multi-arm coordinating robots," in *Proc. of the 33rd IEEE Midwest Symposium on Circuits and Systems*, pp. 673–676, 1990.

[142] Asada and J. E. Slotine, *Robot Analysis and Control.* New York, NY: Wiley, 1986.

[143] P. K. Khosla, "Categorization of parameters in the dynamic robot model," *IEEE Trans. Robotics and Automation*, vol. 5, no. 3, pp. 261–268, 1989.

[144] A. Liegois, W. Khalil, J. M. Dumas, and M. Renaud, "Mathematical models of interconnected mechanical systems," in *Proc. of the Symp. on Theory and Practice of Robots and Manipulators*, 1976.

[145] Y. Stepanenko and M. Vukobratovic, "Dynamics of articulated open-chain active mechanisms," *Math-Biosciences*, vol. 28, pp. 137–170, 1976.

[146] W. W. Armstrong, "Recursive solution to the equations of motion of an n-link manipulator," in *Proc. of the 5th World Congress on the Theory of Machines and Mechanisms*, 1979.

[147] J. Y. S. Luh, M. W. Walker, and R. P. Paul, "On-line computational scheme for mechanical manipulators," *ASME J. of Dynamic Systems, Measurement, and Control*, vol. 102, pp. 69–76, 1980.

[148] Y. Stepanenko, "A method of dynamic analysis of spatial mechanisms," *Mechanics of Machines (USSR)*, vol. 23, pp. 117–124, 1970.

[149] J. Y. S. Luh, "An anatomy of industrial robots and their controls," *IEEE Trans. Automatic Control*, vol. 28, pp. 133–153, 1983.

[150] C. H. An, C. G. Atkeson, and J. M. Hollerbach, "Estimation of inertial parameters of rigid body links of manipulators," in *Proc. of the 24th Conf. Decision and Control*, 1985.

[151] W. S. Lu and Q. H. Meng, "Regressor formulation of robot dynamics: computation and applications," Submitted to IEEE Trans. on Robotics and Automation, 1991.

[152] P. K. Khosla and T. Kanade, "Parameter identification of robot dynamics," in *Proc. of the 24th Conf. on Decision and Control*, pp. 1754–1760, 1985.

[153] W. S. Lu and Q. H. Meng, "A closed-form formula for the manipulator regressor," in *Proc. of the 1990 Canadian Conf. on Electrical and Computer Engineering*, pp. 2.5.1–2.5.4, 1990.

[154] W. S. Lu and Q. H. Meng, "Recursive computation of manipulator regressor and its application to adaptive motion control of robots," in *Proc. of the 1991 IEEE Pacific Rim Conf.*, pp. 170–173, 1991. (Invited).

[155] J. E. Slotine and W. Li, "Composite adaptive control of robot manipulators," *Automatica*, vol. 25, no. 4, pp. 509–519, 1989.

[156] J. Yuan and Y. Stepanenko, "Computing a manipulator regressor without acceleration feedback," *Robotica*, vol. 10, pp. 269–275, 1992.

[157] B. Armstrong, O. Khatib, and J. Burdick, "The explicit dynamic model and inertial parameters of the puma 560 arm," in *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp. 510–518, 1986.

[158] C. S. G. Lee, "Robot arm kinematics, dynamics, and control," *Computer*, vol. 15, no. 12, pp. 62–80, 1982.

[159] M. B. Leahy Jr., K. P. Valavanis, and G. N. Saridis, "Evaluation of dynamic models for puma robot control," *IEEE Trans. on Robotics and Automation*, vol. 5, no. 2, pp. 242–245, 1989.

[160] T. Yoshikawa, *Foundations of Robotics.* Cambridge, MA: The MIT Press, 1990.

[161] M. W. Spong and R. Ortega, "On adaptive inverse dynamics control of rigid robots," *IEEE Trans. on Automatic Control*, vol. 35, no. 1, pp. 92–95, 1990.

[162] M. Erlic, *A Study in Unconstrained Motion Control of A Robotic Manipulator.* Master's thesis, University of Victoria, 1989.

[163] M. Erlic, K. Jones, and W. S. Lu, "Hardware interface configuration for motion control of the puma 560 and the mitsubushi rm-501 robots," in *Proc. of the IEEE Pacific Rim Conf.*, 1991.

[164] A. J. Koivo and G. A. Bekey, "Report of workshop on coordinated multiple robot manipulators: planning, control, and applications," *IEEE J. Robotics and Automation*, vol. 4, no. 2, 1988.

[165] Q. H. Meng and W. S. Lu, "On motion coordination of multirobot systems," in *Proc. of the Third Int. Conf. on Advances in Communication and Control Systems*, 1991.

[166] C. Wu and R. P. Paul, "Resolved motion force control of robot manipulator," *IEEE Trans. on Systems, Man and Cybernetics*, vol. 12, no. 3, pp. 266–275, 1982.

[167] D. G. Luenberger, *Linear and Nonlinear Programming.* Reading:MA: Addison-Wesley, second ed., 1984.

[168] J. Cea, *Lectures on Optimization - Theory and Algorithms.* Bombay: TATA Inst. of Fundamental Research, 1978.

[169] K. I. Kim and Y. F. Zhang, "Unknown load distribution of two industrial robots," in *Proc. of IEEE Int. Conf. Robotics and Automation*, 1991.

[170] R. Dubey and J. Y. S. Luh, "Redundant robot control for higher flexibility," in *Proc. of IEEE Int. Conf. Robotics and Automation*, 1987.

[171] J. T. Wen, "Time domain and frequency domain conditions for strict positive realness," *IEEE Trans. on Automatic Control*, vol. 33, no. 10, pp. 988–992, 1988.