INFORMATION TO USERS

This reproduction was made from a copy of a document sent to us for microfilming. While the most advanced technology has been used to photograph and reproduce this document, the quality of the reproduction is heavily dependent upon the quality of the material submitted.

The following explanation of techniques is provided to help clarify markings or notations which may appear on this reproduction.

1. The sign or "target" for pages apparently lacking from the document photographed is "Missing Page(s)". If it was possible to obtain the missing page(s) or section, they are spliced into the film along with adjacent pages. This may have necessitated cutting through an image and duplicating adjacent pages to assure complete continuity.

2. When an image on the film is obliterated with a round black mark, it is an indication of either blurred copy because of movement during exposure, duplicate copy, or copyrighted materials that should not have been filmed. For blurred pages, a good image of the page can be found in the adjacent frame. If copyrighted materials were deleted, a target note will appear listing the pages in the adjacent frame.

3. When a map, drawing or chart, etc., is part of the material being photographed, a definite method of "sectioning" the material has been followed. It is customary to begin filming at the upper left hand corner of a large sheet and to continue from left to right in equal sections with small overlaps. If necessary, sectioning is continued again—beginning below the first row and continuing on until complete.

4. For illustrations that cannot be satisfactorily reproduced by xerographic means, photographic prints can be purchased at additional cost and inserted into your xerographic copy. These prints are available upon request from the Dissertations Customer Services Department.

5. Some pages in any document may have indistinct print. In all cases the best available copy has been filmed.

University
Microfilms
International

300 N. Zeeb Road
Ann Arbor, MI 48106

KOENIG, MARK A.

A DECENTRALIZED ADAPTIVE CONTROL SCHEME FOR ROBOTIC MANIPULATORS

THE UNIVERSITY OF ARIZONA M.S. 1985

A DECENTRALIZED ADAPTIVE CONTROL SCHEME FOR ROBOTIC MANIPULATORS

by

Mark A. Koenig

A Thesis Submitted to the Faculty of the

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

In Partial Fulfillment of the Requirements
For the Degree of

MASTER OF SCIENCE
WITH A MAJOR IN ELECTRICAL ENGINEERING

In the Graduate College

THE UNIVERSITY OF ARIZONA

1 9 8 5

STATEMENT BY AUTHOR

This thesis has been submitted in partial fulfillment of requirements for an advanced degree at The University of Arizona and is deposited in the University Library to be made available to borrowers under rules of the Library.

Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgment of source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the head of the major department or the Dean of the Graduate College when in his or her judgment the proposed use of the material is in the interests of scholarship. In all other instances, however, permission must be obtained from the author.

SIGNED: Mark A. Koenig

APPROVAL BY THESIS DIRECTOR

This thesis has been approved on the date shown below:

M. K. SUNDARESHAN
Associate Professor of Electrical and Computer Engineering

12/7/84
Date

TABLE OF CONTENTS

TABLE OF CONTENTS -- Continued

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS -- Continued

ABSTRACT

Due to the highly nonlinear and uncertain dynamics inherent in manipulators, precision tracking is difficult to obtain with conventional feedback control schemes. This thesis proposes a decentralized adaptive control scheme which automatically adjusts to the varying dynamic coefficients and features feedforward compensation to decouple the joint dynamics. Unlike previous applications of adaptive control to this problem, this scheme includes the motor parameters within the adaptation loop. Furthermore, the decentralized nature of this scheme allows for an economical implementation of the control law, using a dedicated microprocessor for each joint. Simulation results demonstrate the feasibility of this scheme and illustrate the effect of model structure on closed loop performance.

CHAPTER 1

INTRODUCTION

Robotic manipulators are being successfully applied to industrial automation tasks in increasing numbers. There are many industrial processes which require greater accuracy and speed of manipulation than present commercial manipulators can provide, however. The highly nonlinear coupled dynamics of manipulator arms render conventional PID controllers inadequate for these precision tracking tasks. An alternate approach, the computed torque technique, calculates the required joint torques given desired joint positions, velocities, and accelerations. Unfortunately, this involves real-time computation of the full manipulator dynamics and is thus expensive to implement. An adaptive control scheme which alters the control signal in response to changing system parameters and external disturbances seems to be a promising solution to the manipulator control problem. In this study, a decentralized, adaptive control featuring feedforward compensation is proposed.

1.1 Objectives of the Manipulator Control System

The type of manipulators considered is the class of machines currently being used in industry. They consist of a system of rigid bodies (links) connected serially by revolute or prismatic (linear) joints. The joints are driven by control torques generated by electric, hydraulic, or pneumatic actuators. Most industrial manipulators have

six joints, with each joint providing one degree of freedom. Three degrees of freedom are necessary to place an object in an arbitrary X, Y, Z position in cartesian space. An additional three degrees of freedom are necessary to arbitrarily orient (pitch, roll, yaw) an object in space. A six degree of freedom machine will generally have seven links (the first link is a stationary base), six joints, and an end effector which is a gripper or tool. By convention, the joints are designated numerically, starting at the base and proceeding out towards the gripper. Thus, joint 1 designates the joint connecting the base (link 0) to link 1, and joint 6 designates the final wrist joint.

In order to maintain generality, no particular manipulator design is assumed in the control law development. This allows the control to be used with little or no adjustment on various machines.

The control objective is to provide good end-effector tracking and regulation of reference inputs. Tracking is characterized by the ability of the system to follow reference inputs. Regulation is the ability to maintain good tracking response in the presence of deterministic or stochastic disturbances.

Not only must the end-effector move from point A to point B, but the trajectory the end-effector follows between these two points in the work space is specified and must be tracked. This so-called continuous path capability is important in paint-spraying, welding, and machining applications, as well as pick and place applications in cluttered environments.

The reference inputs can be joint displacements only, joint displacements and velocities, or joint displacements, velocities, and accelerations. The reference inputs are functions of time and are generated by a path planning system which is assumed to be separate from the control system. The path planning system may generate the reference inputs in real time using machine vision and image processing techniques, or it may be generated off-line. In regard to control system design, it is not assumed that the refernece trajectory is known a priori.

Digital control systems are emphasized in this study as they are more reliable, flexible, and less susceptible to noise problems than their analog counterparts. The design objective will be to synthesize a control algorithm which gives good performance but is computationally simple enough to be implemented with microprocessor-based hardware.

The maximum acceptable sampling period limits the complexity of the algorithm. There are two factors to consider. It is well known that when the sampling period becomes significantly large when compared to the dominant time constants of the continuous plant, the closed-loop poles will approach or leave the unit circle in the complex Z-plane, resulting in stability problems. The second factor is that the control inputs must not excite the structural resonant frequencies of the manipulator links. Although this and most other control system studies assume that the links are rigid members, in reality they deform under load and may be characterized by some structural resonant frequency, f_{st}. If the sampling period is too long, the sampling rate will approach f_{st}, and the control input modulations will excite the structural

resonant frequencies of the links, resulting in instability. In practice, it is usually this latter factor which limits the complexity of the control algorithm. A recent study (Luh, Walker, and Paul 1980) has shown that the sampling frequency of the digital control system, f_s, should be at least six times the resonant frequency. For the Stanford manipulator this implies that f_s should be at least 60 Hz.

The continuous path objective requires the realization of independent joint control (IJC). That is, in order for the joints to precisely track their individual trajectories, the closed loop system must compensate for any disturbance torques, including those caused by interactions between the joint dynamics. As will be shown in Chapter 2, the joint dynamics are significantly coupled. These coupling torques are difficult to measure and expensive to compute from physical laws. Another unique problem posed by the manipulator dynamics is that the effective inertia of each joint varies as the arm is moved along the trajectory. These effective inertia terms are also very expensive to compute. Thus, the justification for using adaptive control in this application is two-fold: (1) the independent joint control objective requires on-line estimation of the coupling torques; and (2) the precision tracking objective requires on-line estimation of the effective inertia terms.

The adaptive control scheme can be developed within a centralized or a decentralized framework. A single, centralized controller uses a large multiple-input/multiple-output (MIMO) model of the system. With six joints or more to control, and two states per joint (position and

velocity), the matrix manipulations typical of MIMO control algorithms require a fast, powerful, expensive processor.

A decentralized scheme, on the other hand, can be naturally implemented by using distributed processing techniques. The system is treated as N single-input/single-output (SISO) subsystems with 1 controller dedicated to each subsystem or joint. The performance cost, if any, is attributable to the fact that the individual controllers ignore some information about the other subsystem which is normally available. In an effort to design an effective, reliable control system of reasonable cost, a decentralized scheme is pursued.

1.2 Outline of the Thesis

The remainder of this thesis is devoted to determining a decentralized adaptive control scheme for the manipulator dynamics problem. Chapter 2 gives a detailed description of the manipulator dynamics. It is the highly complex and uncertain nature of the dynamics which makes the use of adaptive control an attractive option. A survey of conventional manipulator control schemes and their shortcomings is also presented. In Chapter 3, the basic concepts of adaptive control are presented. A new decentralized adaptive control algorithm is developed in Chapter 4. Past efforts which apply adaptive control to the manipulator dynamics problem are summarized. Similarities and differences between these schemes and the scheme developed here are highlighted. Simulation results are presented in Chapter 5. These results demonstrate the feasibility of this scheme and provide a comparison of the

effectiveness of the three linear models upon which the control can be based. In Chapter 6, the thesis concludes by summarizing the contributions of this work.

CHAPTER 2

AN OVERVIEW OF MANIPULATOR DYNAMICS AND CONVENTIONAL CONTROL TECHNIQUES

2.1 Manipulator Dynamics

The manipulator dynamics are examined for two reasons. First, the plant must be characterized and classified so that the proper control techniques can be used. This includes determining whether a system is accurately described as linear or nonlinear, as a set of single-input/single-output (SISO) subsystems or one multiple-input/multiple-output (MIMO) system. If the plant is a MIMO system, the significance of the coupling terms must be known. To answer these questions, it is convenient to have functionally explicit expressions for the plant dynamics. For the industrial manipulator, which is a mechanical system, the theory of Lagrangian Mechanics provides such information.

The plant dynamics may also be used directly in determining the control signals. In this case, efficient numerical algorithms which compute the equations of motion are necessary. For manipulators, the Newton-Euler algorithm (Luh et al. 1980) performs this function. In this section, useful information derived from both Lagrangian techniques and the Newton-Euler algorithm is presented.

2.1.1 Equations of Motion from the Lagrangian Formulation

Initially, the effects of joint friction, transmission mechanisms, and actuator dynamics are neglected. The manipulator is thus viewed as a dynamic system with joint torques as inputs and joint positions and velocities as outputs. The equations of motion for simple manipulators may be written symbolically by direct use of Newton's Laws. For three or more degrees of freedom, however, the analysis becomes complicated and the theory of Lagrangian Mechanics is used. Using this technique yields the following equations of motion for an N degree of freedom manipulator:

$$D(y)\ddot{y} + f(y,\dot{y}) + g(y) = \tau \qquad (2.1.1)$$

where $y \in R^N$ is the generalized coordinate vector representing joint displacements, $\dot{y}$ and $\ddot{y}$ are the joint velocity and joint acceleration vectors, respectively, $D(y) \in R^{NxN}$ is the symmetric inertia matrix, $f(y,\dot{y}) \in R^N$ is a vector of centripetal and coriolis forces, $g(y) \in R^N$ is a vector of gravity forces, and $\tau \in R^N$ is the vector of joint input torques or forces. The joint displacement vector, y,

$$y = [y_1, y_2, \ldots, y_N]^T \qquad (2.1.2)$$

is the angular displacement measured in radians if joint i is revolute, or a linear displacement measured in meters if joint is is prismatic. The centripetal and coriolis vector, $f(y,\dot{y})$, may be written as:

$$f(y,\dot{y}) = \begin{bmatrix} \dot{y}^T F_1 \dot{y} \\ \dot{y}^T F_2 \dot{y} \\ . \\ . \\ \dot{y}^T F_N \dot{y} \end{bmatrix} \qquad (2.1.3)$$

where,

$$F_i = \begin{bmatrix} 2f_{i_{11}} & f_{i_{12}} & \cdots & f_{i_{1N}} \\ f_{i_{21}} & 2f_{i_{22}} & \cdots & f_{i_{2N}} \\ . & . & \cdots & . \\ . & . & \cdots & . \\ f_{i_{1N}} & f_{i_{2N}} & \cdots & 2f_{i_{NN}} \end{bmatrix} \qquad (2.1.4)$$

It is clear from examination of equations (2.1.1) through (2.1.4) that for an N degree of freedom manipulator, the dynamics are described by a set of N coupled, nonlinear, second order differential equations. In order to obtain some physical insight into the dynamics, expand (2.1.1) for a 3 degree of freedom manipulator to obtain:

$$d_{11}\ddot{y}_1+d_{12}\ddot{y}_2+d_{13}\ddot{y}_3+f_{111}\dot{y}_1^2+f_{122}\dot{y}_2^2+f_{133}\dot{y}_3^2+f_{112}\dot{y}_1\dot{y}_2+f_{113}\dot{y}_1\dot{y}_3+f_{123}\dot{y}_2\dot{y}_3+g_1 = \tau_1$$

$$d_{21}\ddot{y}_1+d_{22}\ddot{y}_2+d_{23}\ddot{y}_3+f_{211}\dot{y}_1^2+f_{222}\dot{y}_2^2+f_{233}\dot{y}_3^2+f_{212}\dot{y}_1\dot{y}_2+f_{213}\dot{y}_1\dot{y}_3+f_{223}\dot{y}_2\dot{y}_3+g_2 = \tau_2$$

$$d_{31}\ddot{y}_1+d_{32}\ddot{y}_2+d_{33}\ddot{y}_3+f_{311}\dot{y}_1^2+f_{322}\dot{y}_2^2+f_{333}\dot{y}_3^2+f_{212}\dot{y}_1\dot{y}_2+f_{313}\dot{y}_1\dot{y}_3+f_{323}\dot{y}_2\dot{y}_3+g_3 = \tau_3 \qquad (2.1.5)$$

The coefficients d_{ii}, d_{ij}, f_{ijp}, and g_i are known as the dynamic coefficients of the manipulator. They are nonlinear functions of y and the

inertia parameters of the links. The dynamic coefficients have the following physical interpretations:

1. A coefficient of the form d_{ii} is known as the effective inertia of joint i. An acceleration at joint i is caused by a torque at joint i equal to $d_{ii}\ddot{y}_i$. In general, d_{ii} is a function of joint displacements y_{i+1} through y_N and of the inertia parameters of links i through N.
2. The coefficients d_{ij}, $(i \neq j)$, are the coupling inertia terms which represent reaction torques of magnitude $d_{ij}\ddot{y}_j$ felt at joint i due to an acceleration at joint j. Since $d_{ij} = d_{ji}$, the D matrix is symmetric.
3. The coefficients f_{ijj} represent centripetal forces felt at joint i due to the velocity of joint j, i.e., $\dot{y}_j$.
4. The coefficients f_{ijp} represent coriolis forces felt at joint i and generated by the velocities of joint j and joint p.
5. The term g_i represents forces or torques applied at joint i due to gravity.

Some of the dynamic coefficients may be identically equal to zero. For instance, the coefficients $f_{iii} = 0$ since a centripetal force will never interact with the joint that creates it. Also, some of the coefficients may be identically equal to zero due to a particular mechanical design of a specific manipulator. Many industrial manipulators, for example, are designed such that the axis of rotation of joint 1 is parallel to the gravity field, making $g_1 = 0$. In general, however, most of the

terms will be present in the equations, and the complexity of the dynamics will be nontrivial.

The relative importance of the various dynamic coefficients is difficult to ascertain due to the wide variety of mechanical designs. One obvious conclusion is that the centripetal and coriolis forces are more significant for fast manipulator movements than slow movements.

It is also possible to generalize some results from an exhaustive study (Bejczy 1974) of the dynamics of a fairly representative machine--the JPL/RRP manipulator. The JPL/RRP manipulator is a mid-size robot with six degrees of freedom. The first joint is a revolute joint, providing rotational motion in the horizontal plane. Joint 2 is also revolute, and provides rotational motion in the vertical plane. The third joint is a prismatic joint providing linear motion of a 1.2 meter boom (link 3). The last three joints give pitch, roll, and yaw rotations of the wrist. The working envelope of the JPL/RRP manipulator is roughly a hemisphere with a radius of 1.2 meters.

The primary objective of Bejczy's study was to obtain a simplified but accurate model of the dynamics for control purposes. Bejczy found that the gravity terms, g_i, for joints 2 through 5, were significant compared to the effective inertia terms, $d_{ii}\ddot{y}_i$. It was also determined that with the exception of joint 1, the gravity terms overshadowed the coupling inertia terms, $d_{ij}\ddot{y}_j$, by an order of magnitude or more. The germ g_1 was identically equal to zero since joint one had its axis of rotation parallel to the gravity field, and thus the effective inertia

term, $d_{11}\ddot{y}_1$, and the coupling inertia terms, $d_{1j}\ddot{y}_j$, were deemed significant.

In Bejczy's study, the variation of the effective inertia terms, d_{ii}, with manipulator configuration and loading conditions was also investigated. It was found that the ratio $d_{ii_{max}}/d_{ii_{min}}$ decreased as i approached N. In other words, the variation of effective inertia due to manipulator configuration and load is generally less for the joints in the hand than for the joints close to the base. For the first joint of the JPL/RRP manipulator, the ratio of the maximum effective inertia with a 1.8 kg mass in the gripper to the effective inertia with no load was, $d_{ii_{max}}/d_{ii_{min}} = 6.75$. The same ratio with no load in the gripper was found to be $d_{ii_{max}}/d_{ii_{min}} = 4.5$

In conclusion, the manipulator dynamics are extremely complex. A manipulator operating in the continuous path mode will have several or all joints moving simultaneously, and the motion of one joint or a torque applied at one joint will have a dynamic effect on the other joints. Furthermore, since the dynamic coefficients are functions of y, the magnitude of the dynamic coupling is a function of the configuration of the manipulator.

2.1.2 The Newton-Euler Algorithm

The above analysis was based on Lagrangian mechanics which gives explicit symbolic relations for the dynamic coefficients. The disadvantage of Lagrangian mechanics is that many calculations are required if one must actually compute the dynamic coefficients. With a six degree

of freedom manipulator, for example, 66,271 floating point multiplications and 51,548 additions are required to compute all the dynamic coefficients. For this reason, real-time control systems which compute the dynamics based on physical laws must use a more efficient algorithm.

The most efficient algorithm to date is the recursive Newton-Euler algorithm (Luh et al. 1980). In contrast to the Lagrangian formulation, the Newton-Euler algorithm computes the joint torques required to produce a given motion without solving explicitly for the dynamic coefficients of Equation (2.1.1). This is, of course, very useful for control purposes. The algorithm consists of two sets of equations, a forward recursion and a backward recursion. The forward recursion involves the link-by-link propagation of kinematics information (link velocities and accelerations) from the base out to the gripper. In the backward recursion, forces and moments are successively transformed from the gripper back to the base to obtain the required joint torques. The Newton-Euler algorithm requires 150N-48 multiplications and 131N-48 additions per control step or setpoint, where N is the number of joints. Thus, for a 6 degree of freedom manipulator, 1590 floating point operations must be executed in real-time. Although the Newton-Euler algorithm is more efficient than the Lagrangian formulation, it is still computationally intensive.

2.1.3 State Variable Model

If modern control theory techniques are to be applied to the manipulator problem, a state variable representation of the dynamics is needed. The inertia matrix, D(y), is symmetric and is also shown to be

positive definite (Horowitz and Tomizuka 1980). Hence, it is invertible, and thus,

$$\ddot{y} = D^{-1}(y)[-f(y,\dot{y}) - g(y) + \tau] \tag{2.1.6}$$

which follows from equation (2.1.1). Defining the state vector, $x \epsilon R^{2N}$ as

$$x_i = y_i$$

$$x_{i+N} = \dot{y}_i \tag{2.1.7}$$

for i = 1, 2, . . . N, equation (2.1.6) may be written in state variable form:

$$\dot{x} = l(x,u) \tag{2.1.8}$$

where $u \epsilon R^N$ is the control vector and is identically equal to τ. The same problems are encountered in computing the vector l as in computing the dynamic coefficients of equation (2.1.1).

2.1.4 Dynamics of Actuators and Transmission Mechanisms

So far, only the mechanical dynamics of manipulators have been considered. The entire system, however, is composed of an open, kinematic chain of links characterized by the mechanical dynamics as well as actuators and transmission mechanisms which contribute to the overall dynamics.

Transmission mechanisms transfer forces or torques from the actuator shafts to the manipulator joints. Gears, belts, and lead screws are often used to perform this function. Unfortunately, these mechanisms

introduce static friction (stiction), dynamic friction, and backlash into the system. These characteristics are difficult to identify and are often left out of the manipulator model. Recent research efforts have been made to design direct drive manipulators to alleviate this problem (Asada, Kanade, and Takeyama 1982).

Actuators are electro-mechanical devices which convert electrical control signals to a force or torque. Hydraulic actuators and armature controlled D.C. motors are the two types of actuators most commonly used in manipulators. For convenience, only armature controlled D.C. motors are considered in this study.

An armature controlled D.C. motor is shown schematically in Figure 2.1, where

V_a = applied voltage (the control signal)

R_m = motor resistance

L_m = motor inductance

i_a = armature current

i_f = field current

J = combined inertia of armature assembly, motor shaft, and load

T_L = load disturbance torque

E = back e.m.f. voltage

If the inductance is small, its contribution to the overall dynamics is insignificant and it can therefore be neglected. Thus, the applied voltage is:

$$V_a = E + i_a R_m \tag{2.1.9}$$

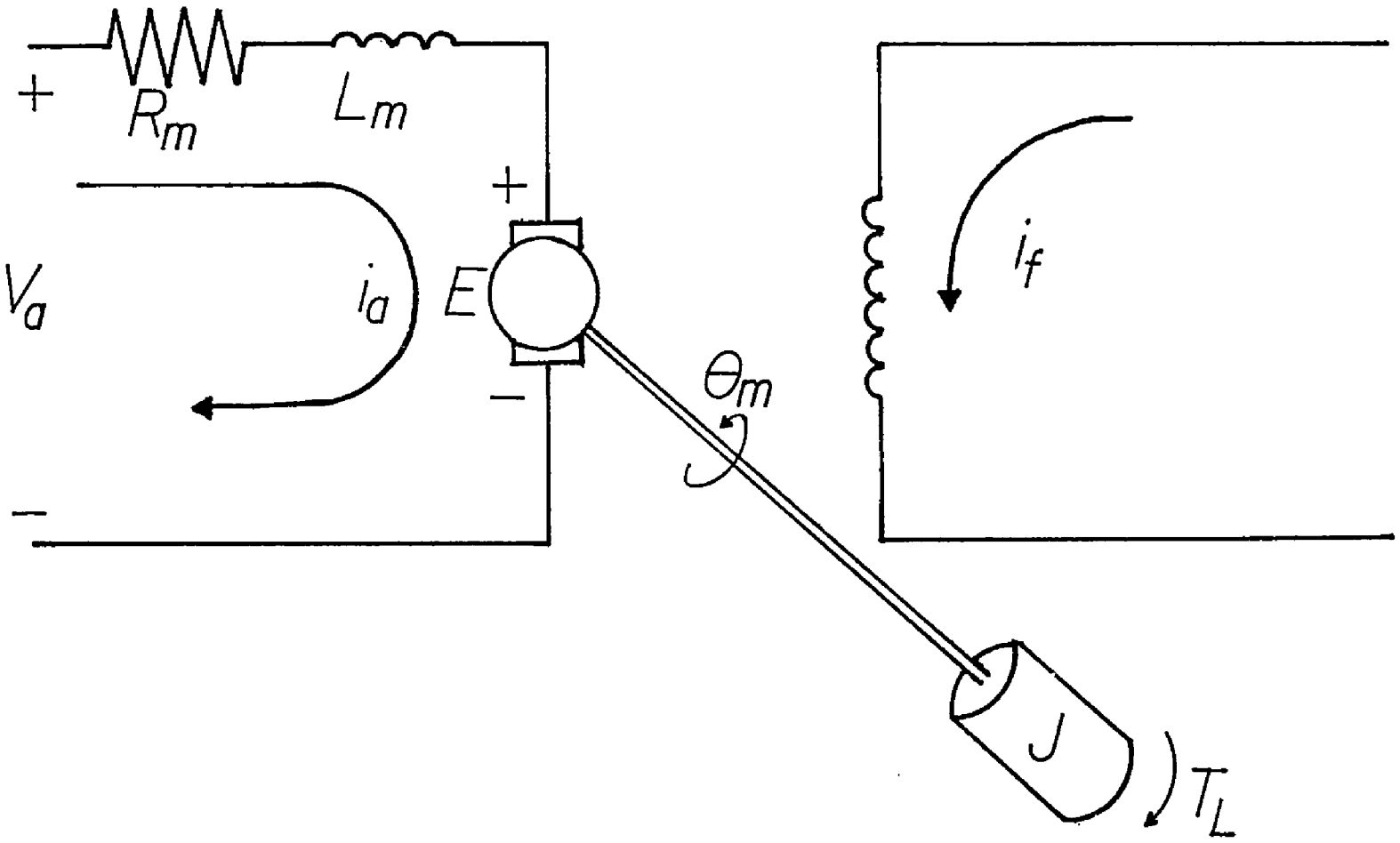

Figure 2.1. Armature Controlled D.C. Motor.

The back e.m.f. is expressed as:

$$E = K_B\dot{\theta}_m \tag{2.1.10}$$

where

K_B = motor back e.m.f. constant

$\dot{\theta}_m$ = angular velocity of the motor shaft.

The developed torque, T_d, may be expressed as:

$$T_d = K_t i_a \tag{2.1.11}$$

where K_t is the motor torque constant. Substituting for E and i_a from (2.1.10) and (2.1.11) into (2.1.9) yields the equation for the motor dynamics:

$$V_a = K_B\dot{\theta}_m + (R_m/K_t)T_d \tag{2.1.12}$$

2.1.5 System Integration

The motor dynamics must be combined with the manipulator model to obtain the overall system dynamics. This may be done in a variety of ways. The specific representation which is used depends on the nature of the control system.

Many manipulator controllers compute the developed torque necessary for the manipulator to follow the desired joint angle trajectory. The control torque computation is based solely on the manipulator model with no regard for the actuator dynamics. In this case, the motor/manipulator dynamics integration is very easy since all that is needed is an expression relating the controller output, V_a, to the desired joint

torque. Equation (2.1.12) gives the relationship. A functional block diagram of the system is shown in Figure 2.2.

Another method for integrating the motor dynamics with the manipulator dynamics is to view each joint motor as a subsystem (Vukobratovic and Stokic 1981). The subsystems are coupled by disturbance torques, T_L, which represent the coupling inertia, centripetal, coriolis, and gravity terms in equation (2.1.1). A functional block diagram is shown in Figure (2.3).

The manipulator dynamics for subsystem i are expressed as:

$$J_i\ddot{y}_i + B_{v_i}\dot{y}_i + T_{L_i}/n_i = T_{d_i}/n_i \qquad (2.1.13)$$

where

$$J_i = J_{m_i} + (1/n_i^2)d_{ii}(y) \qquad (2.1.14)$$

and

J_i = total inertia of subsystem i

J_{m_i} = combined moment of inertia of motor drive shaft and armature assembly

d_{ii} = effective inertia of joint i

n_i = gear, pulley, or lead-screw transmission ratio (e.g., ratio of load gear teeth to motor gear teeth)

B_{v_i} = combined friction coefficient of the motor shaft and joint i

T_{L_i} = disturbance torque at joint i representing coupling inertia, centripetal, coriolis, and gravity torques

T_{d_i} = developed motor torque

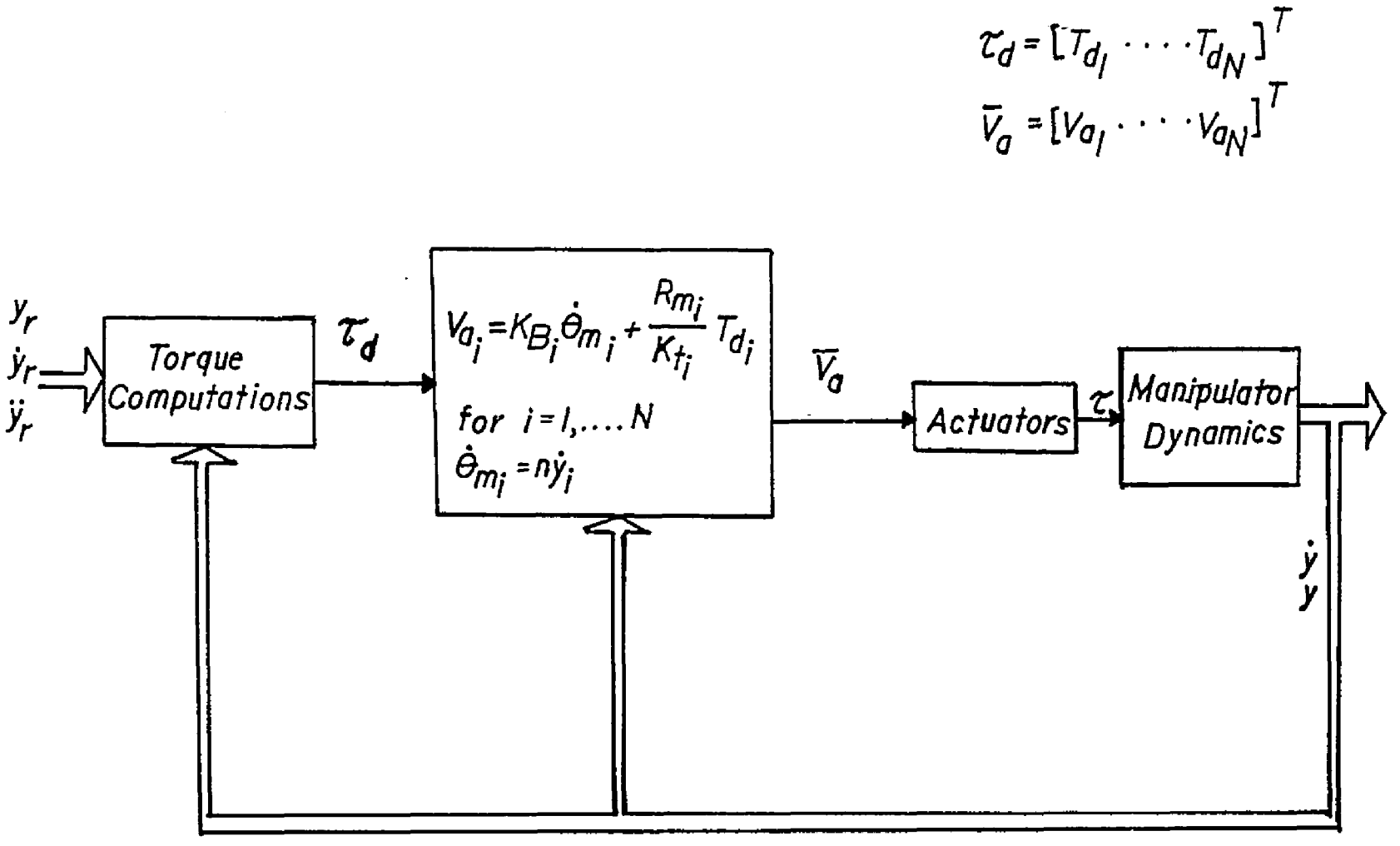

Figure 2.2. System Integration for Computed Torque Control.

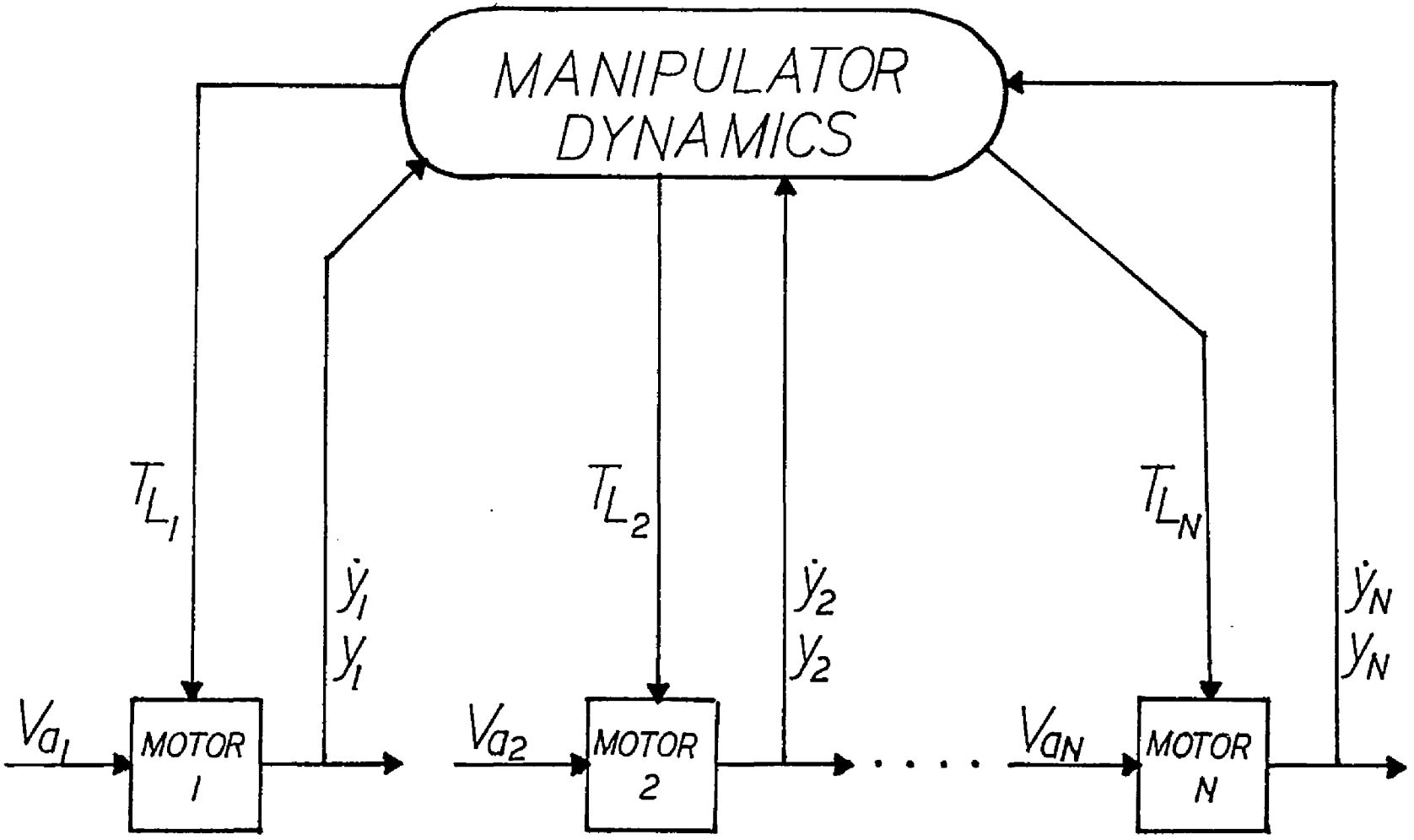

Figure 2.3. Distributed System Approach.

Note that in general, $d_{ii}(y) >> J_{m_i}$, and thus for a direct drive joint ($n_i = 1$) or a low gear ratio joint, the total inertia will be dominated by the varying effective inertia, $d_{ii}(y)$. If $n_i >> 1$, however, the total inertia will be much less dependent on $d_{ii}(y)$. Combining the motor dynamics equation, (2.1.12), with the equation for the manipulator dynamics (2.1.13), yields:

$$J_i\ddot{y}_i + B_{v_i}\dot{y}_i + T_{L_i}/n_i = [(K_{t_i}/(n_i R_{m_i})]V_{a_i} - (K_{B_i}/R_{m_i})K_{t_i}\dot{y}_i \tag{2.1.15}$$

which is the differential equation for the overall system. An explicit block diagram is shown in Figure 2.4.

Letting $x_{i_1} = y_i$ and $x_{i_2} = \dot{y}_i$, the subsystem dynamics may also be expressed in state variable form:

$$S : \dot{x}_i = A_i x_i + b_i u_i + d_i T_{L_i} \tag{2.1.16}$$

where

$$u_i = V_{a_i} \tag{2.1.17}$$

$$A_i = \begin{bmatrix} 0 & 1 \\ 0 & \frac{-1}{J_i}\left(\frac{K_{B_i}K_{t_i}}{R_{m_i}} + B_{v_i}\right) \end{bmatrix} \tag{2.1.18}$$

$$b_i = \begin{bmatrix} 0 \\ \frac{K_{t_i}}{J_i R_{m_i} n_i} \end{bmatrix} \tag{2.1.19}$$

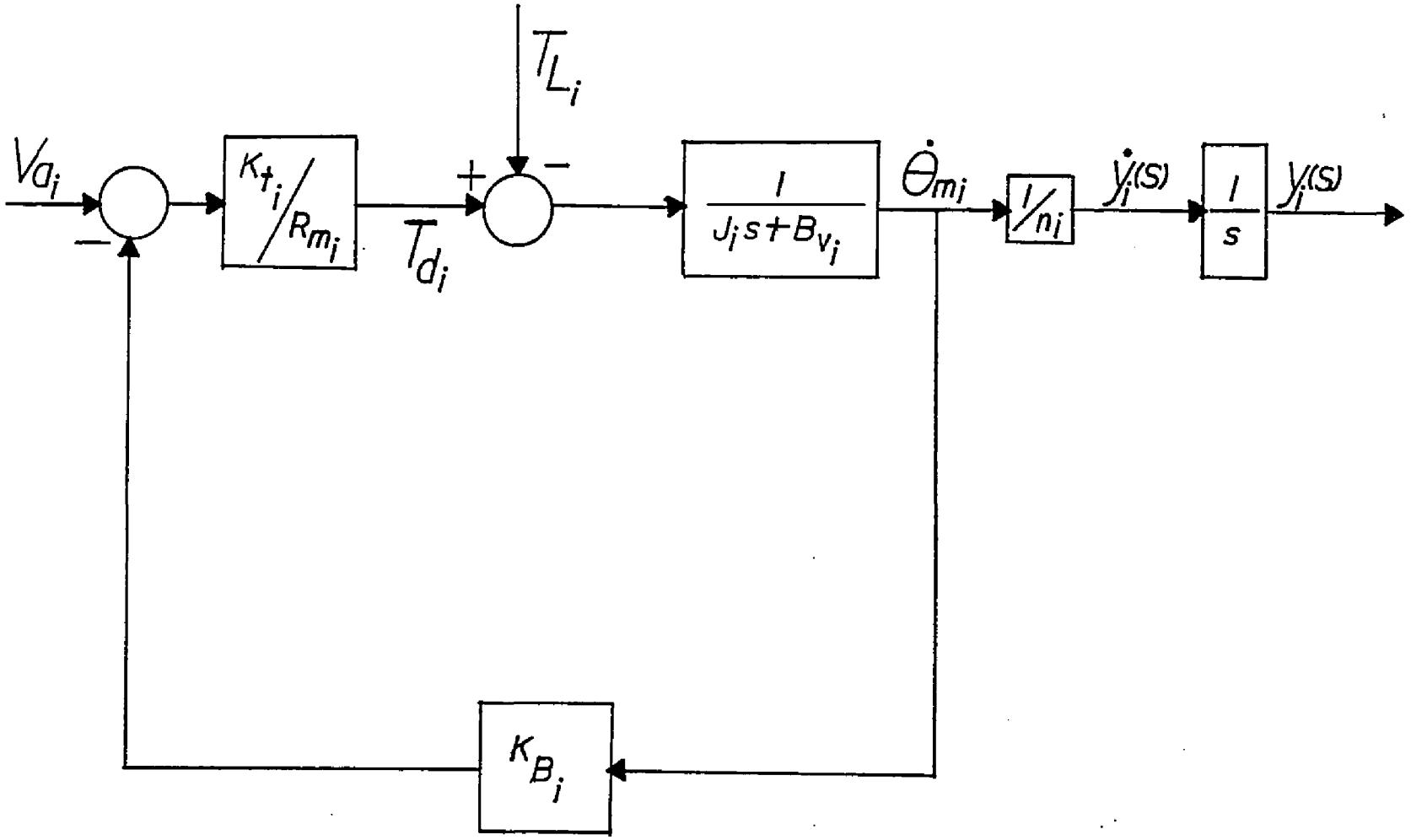

Figure 2.4. Subsystem Dynamics.

$$d_i = \begin{bmatrix} 0 \\ \\ \frac{1}{n_i J_i} \end{bmatrix} \qquad (2.1.20)$$

This dynamic formulation is useful for control schemes which use direct decoupling. It is clear that if the terms $d_i T_{L_i}$ can be compensated, the joint subsystems, S_i, will be decoupled and can then be controlled independently.

This concludes the survey on manipulator dynamics. The purpose of studying the dynamics is to obtain an adequate model on which the design of a control scheme can be based. Note that even for manipulators with no stiction, friction, or backlash, the models which result from using the laws of mechanics are complicated and the actual computation of the dynamic coefficients is computationally burdensome. Furthermore, all commercial manipulators do have transmission mechanisms that contribute to further uncertainties in any model obtained strictly from physical laws.

2.2 Conventional Feedback/Feedforward Controls

The previous sections have underscored the difficulties in obtaining suitable dynamic models of manipulators based on physical laws. All conventional feedback/feedforward control schemes such as PID controllers, the computed torque technique, and optimal state variable feedback controllers are based on such a model. In this section, these three control methods are investigated and their shortcomings are highlighted.

2.2.1 PID Control Schemes

One of the most popular manipulator control schemes used in industrial practice today is the PID controller (Markiewicz 1973), so named because it uses proportional/derivative feedback and integral feedforward compensation. This scheme is based on the manipulator/actuator model given in Figure 2.4. A block diagram for one of the N PID controllers for an N jointed direct drive manipulator ($n_i = 1$) is shown in Figure 2.5. The position error and rate error feedback gains are chosen to achieve a desired transient response. No effort is made to compute the disturbance torques--the non-linear gravity, centripetal, coriolis, and coupling inertia terms which are left out of the linear model. Instead, integral feedforward is used in an attempt to reduce tracking errors due to the disturbance torques. Also note from equation (2.1.14) that the inertia of system i, J_i, is a non-linear function of y. Thus, J_i is a varying quantity. It is shown that except for slow and restricted motion, this scheme yields poor response for continuous-path operation. Although discrete-time control systems are of primary interest in this study, a continuous PID controller is considered here, as it exhibits the same problems as the discrete PID controller.

Referring to Figure 2.5, the input, $Y_{d_i}(s)$, is the Laplace transform of the desired joint poisition in radians if the joint is rotational or in centimeters if the joint is prismatic. Both the actual joint position and the joint velocity are assumed to be measurable. The selectable control parameters are the position feedback gain, K_{p_i}, the velocity feedback gain, K_{v_i}, and the integral gain, K_{I_i}. The plant

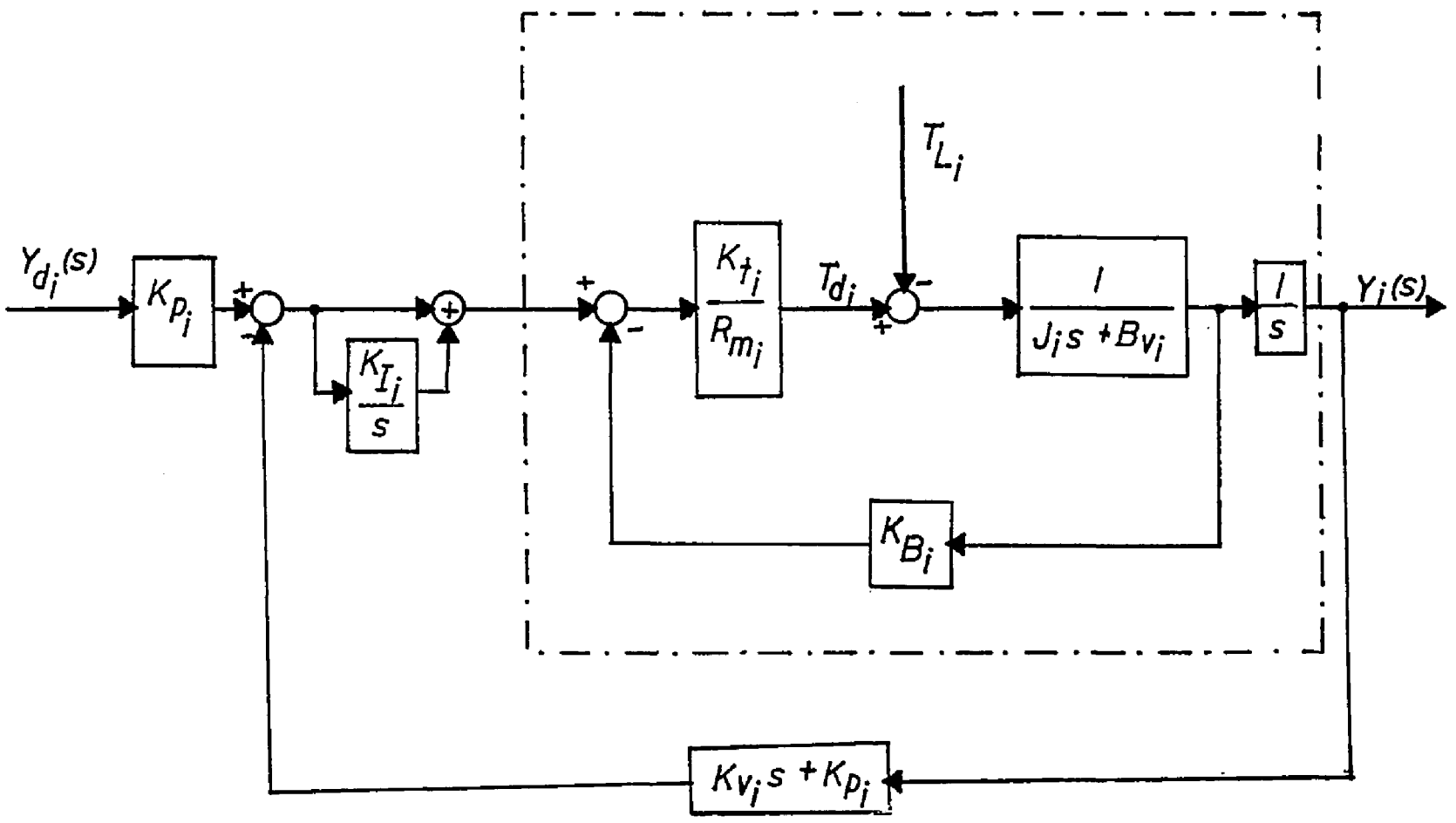

Figure 2.5. PID Control Scheme.

parameters shown in Figure 2.5 are defined in Section 2.1.4. Assuming J_i is constant, the closed loop transfer function is found to be:

$$\frac{Y(s)}{Y_d(s)} = \frac{K_p(s + K_I)}{\frac{R_m}{K_t}Js^3 + (B_v \frac{R_m}{K_t} + K_B + K_v)s^2 + (K_p + K_vK_I)s + K_pK_I} \tag{2.2.1}$$

where the i subscripts have been dropped for the sake of brevity. The design strategy is to initially assume $K_I = 0$, and select K_p and K_v to give desired second order tracking dynamics as characterized by a damping ratio, ζ, and a natural frequency, ω_n. A proper value of K_I is then selected such that the integral action does not significantly degrade the transient response.

If $K_I = 0$, the transfer function reduces to,

$$\frac{Y(s)}{Y_d(s)} = \frac{(K_tK_p)/(R_mJ)}{s^2 + \frac{K_t}{R_mJ}(B_v\frac{R_m}{K_t} + K_B + K_v)s + (K_pK_t)/(R_mJ)} \tag{2.2.2}$$

and expressions for the damping ratio and natural frequency are obtained:

$$\zeta = 1/2\sqrt{\frac{K_t}{JK_pR_m}}\,(\frac{B_vR_m}{K_t} + K_B + K_v) \tag{2.2.3}$$

$$\omega_n = \sqrt{\frac{K_pK_t}{JR_m}} \tag{2.2.4}$$

Thus, K_p is selected to obtain a desired value for ω_n, and K_v is selected to obtain a desired value for ζ.

The magnitude of the disturbance term relative to the other terms in the plant dynamics is quite significant, and therefore the ability of the integral action to compensate for these terms is critical. From Figure 2.5, the output to disturbance torque transfer function may be obtained as,

$$\frac{Y(s)}{T_L(s)} = \frac{-(R_m/K_t)s}{\frac{R_m}{K_t}Js^3 + (K_v + K_B + \frac{R_m}{K_t}B_v)s^2 + (K_vK_I + K_p)s + K_pK_I} \tag{2.2.5}$$

where it is desired that,

$$\lim_{t\to\infty} y(t) = 0 \tag{2.2.6}$$

Using the final value theorem,

$$\lim_{t\to\infty} y(t) = \lim_{s\to o} sY(s) = \lim_{s\to o} \frac{-(R_m/K_t)T_L(s)s^2}{\frac{R_m}{K_t}Js^3 + (K_v+K_B+\frac{R_m}{K_t}B_v)s^2 + (K_vK_I+K_p)s + K_pK_I} \tag{2.2.7}$$

Thus the steady-state error due to torque disturbances will be equal to zero for a step disturbance, a bounded non-zero constant for a ramp disturbance, and unbounded for higher order disturbances. As stated previously, however, the disturbance torques are non-linear functions of y and $\dot{y}$. A well-behaved function such as a step or ramp will not adequately represent the disturbance.

Another problem with the linear PID controller is the effect of changing effective inertia, d_{ii}, on the system closed loop dynamics. As

discussed earlier, $d_{ii}(y)$ can be the major component of the total joint inertia, J_i. Using the results from the JPL/RRP manipulator study ($d_{ii_{max}}/d_{ii_{min}} = 4.36$ with no load on the gripper, and $d_{ii_{max}}/d_{ii_{min}} = 6.75$ with a 1.8 kg object in the gripper), it follows from equations (2.2.3) and (2.2.4) that for a direct-drive manipulator, these variations in effective inertia will create the following variations in damping ratio and natural frequency:

$$\frac{\zeta_{max}}{\zeta_{min}} = \frac{\omega_{n_{max}}}{\omega_{n_{min}}} = \begin{cases} 2.09 \text{ no load} \\ 2.60 \text{ loaded} \end{cases} \qquad (2.2.8)$$

In many practical situations, these variations in dynamic characteristics will be intolerable.

In conclusion, while the linear PID controller is easy to implement and is usually adequate for pick and place applications, it fails to cope with the nonlinear dynamics, making it unsuitable for continuous path operation.

2.2.2 Computed Torque Scheme

The computed torque technique, on the other hand (Paul 1981; Markiewicz 1973) uses a complex manipulator model to compute the nonlinear disturbance terms. These terms are then fedforward, effectively decoupling the joints. If the model is exact, the manipulator could theoretically track the trajectory in an open-loop fashion. This is not possible in practice, however, due to the accumulated modeling errors and the presence of real external disturbance torques. Thus,

rate and position feedback are used to compute correction torques to maintain good tracking response.

As shown in Figure 2.6, a trajectory planning system computes the desired trajectory as a function of time. This sequence of desired position, velocity, and acceleration setpoints $y_r(k)$, $\dot{y}_r(k)$, and $\ddot{y}_r(k)$ are input to the controller.

Neglecting joint friction, stiction, and backlash, the manipulator equation of motion as given in equation (2.2.1) is:

$$D(y)\ddot{y} + f(y,\dot{y}) + g(y) = \tau \tag{2.2.9}$$

The feedback/feedforward control is given by:

$$\tau = D^*(y)[\ddot{y}_r + K_v(\dot{y}_r - \dot{y}) + K_p(y_r - y)] + f^*(y,\dot{y}) + g^*(y) \tag{2.2.10}$$

where

$D^*(y)$ = computed inertia matrix

$f^*(y,\dot{y})$ = computed centripetal and coriolis torques

$g^*(y)$ = computed gravity torques

Substituting (2.2.10) into (2.2.9), the closed loop dynamics are obtained:

$$D(y)\ddot{y} + f(y,\dot{y}) + g(y) = D^*(y)[\ddot{y}_r + K_v(\dot{y}_r - \dot{y}) + K_p(y_r - y)] + f^*(y,\dot{y}) + g^*(y) \tag{2.2.11}$$

where K_p and K_v are the position and velocity feedback gains, respectively. If the modeling is exact, the computed matrices are identical to their actual counterparts, i.e.:

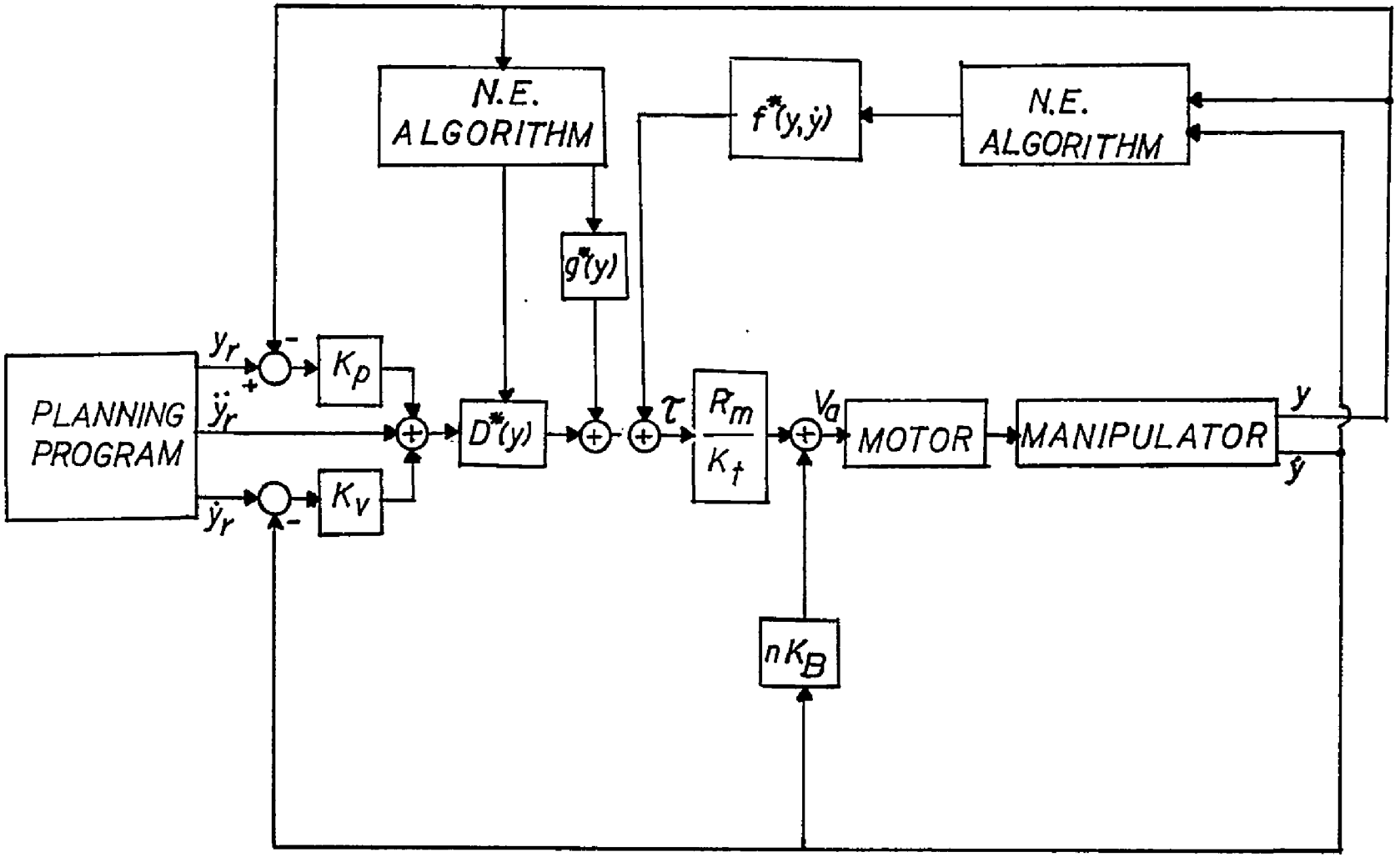

Figure 2.6. Computed Torque Scheme.

$$D^*(y) = D(y) \tag{2.2.12}$$

$$f^*(y,\dot{y}) = f(y,\dot{y}) \tag{2.2.13}$$

$$g^*(y) = g(y) \tag{2.2.14}$$

Substituting (2.2.12) through (2.2.14) into (2.2.11), the closed loop dynamics become:

$$\ddot{y}_r - \ddot{y} + K_v(\dot{y}_r - \dot{y}) + K_p(y_r - y) = 0 \tag{2.2.15}$$

Defining the positional error, e, as $e = y - y_r$, the regulation dynamics are given by:

$$\ddot{e} + K_v\dot{e} + K_p e = 0 \tag{2.2.16}$$

Thus the position error can be regulated with the second order dynamics specified by the designer. Since the control input is a sequence of control voltages, the voltage/torque relation of equation (2.1.12) is used to convert the computed torque to applied motor voltage.

The performance obtained with the computed torque controller compares favorably with the performance of the PID controller. With the computed torque technique, compensation of nonlinearities through feedforward essentially makes the joints dynamically independent. If the model is accurate, transient response characteristics are only a function of the designer specified feedback gains, K_p and K_v. In the PID case, the closed loop dynamics are functions of the varying effective inertias. Also as a result of feedforward, the feedback gains necessary to achieve certain values for ζ and ω_n are much lower than for the PID controller. Thus control signal saturation is less of a problem with the computed torque scheme.

One drawback of this technique is that to achieve good response, the dynamic model used to compute the control, equation (2.2.9), must be accurate. As stated in Section 2.1.4, all commercial manipulators have transmission mechanisms which create uncertainties in the model. Also, in many practical situations, the manipulator may be moving an object of unknown mass and inertia properties. This will lead to an inaccurate computation of the gravity force vector, and equation (2.2.14) will no longer hold. If this is the case, the position error equation becomes:

$$\ddot{e} + K_v\dot{e} + K_p e = D^{*-1}(y)[g^*(y) - g(y)] \qquad (2.2.17)$$

and it is obvious that steady-state tracking error will exist. The accumulation of this error along the trajectory may be unacceptable for precise tracking tasks.

From an implementation point of view, the computed torque technique is not very attractive. The most efficient method for computing the control torques based on equation (2.2.9) is the Newton-Euler algorithm, which requires approximately 1400 floating point operations per setpoint for a six degree of freedom manipulator. Since a minimum sampling rate of roughly 60 hertz must be maintained (as discussed in Section 1.1), the control computer must have a 84,000 floating point operations per second (FLOPS) capability. One possible solution is to base the computed torque computation on a simpler model than equation (2.2.9) and use a more sophisticated scheme than proportional/derivative feedback to generate the corrective torques.

2.2.3 Optimal Control Schemes

Another class of control schemes that is seemingly well suited for manipulators is optimal control. It is well known that optimal control theory can be directly applied to nonlinear systems. The problem could be formulated in many ways, including minimum time or optimal tracking control. Unfortunately, these methods usually involve extensive off-line calculations which require a priori knowledge of the reference input. If the trajectory is being generated in real time by a vision system, however, such a priori knowledge is not available. Thus, a control algorithm based solely upon dynamic programming, minimum time, or optimal tracking theory is not practical.

Use of optimal control theory is possible, however, if used in conjunction with a nominal control (Vukobratovic and Stokic 1981). The computed torque technique based on a simple dynamic model is a good candidate for the nominal control. The purpose of the nominal control is to drive the manipulator within a neighborhood of the desired trajectory. It also reduces, but does not eliminate the coupling between the joint dynamics. Linear state equations which describe the perturbation dynamics of the manipulator are written. The perturbation control consists of a global control and a local control. The global control is used to directly decouple the joint dynamics, enabling the original system of 2N state variables to be written as N subsystems with 2 state variables per subsystem. A local optimal control is applied to each subsystem to regulate the perturbations from the nominal trajectory. The difficulty with this approach lies in finding the perturbation

dynamics, especially for manipulators with three or more degrees of freedom. Recalling the state equations for an N degree of freedom manipulator from Section 2.1.3:

$$\dot{x} = l(x,u) \tag{2.2.18}$$

where $x \in R^{2N}$, $\dot{x} \in R^{2N}$, $l \in R^{2N}$ and

$$\dot{x}_i = \dot{y}_i \qquad \text{for } i = 1, 2, \ldots, N$$

$$\dot{x}_i = \{D^{-1}(y)[-f(y,\dot{y}) - g(y) + \tau]\}^{i-N} \qquad \text{for } i = N+1, \ldots, 2N \tag{2.2.19}$$

where $\{\ \}^{i-N}$ indicates the $(i-N)^{th}$ element of the resulting vector. Thus it can be seen that $l(x,u)$ is a complex, nonlinear vector function. Taking the Taylor series expansion of equation (2.2.18) about the nominal trajectory, x_n, and retaining only the first order terms, the perturbation equations are obtained:

$$\Delta\dot{x} = \left.\frac{\partial l(x,u)}{\partial x}\right|_{\substack{x=x_n \\ u=u_n}} \Delta x + \left.\frac{\partial l(x,u)}{\partial u}\right|_{\substack{x=x_n \\ u=u_n}} \Delta u \tag{2.2.20}$$

where Δx is the state deviation from the nominal trajectory, $\Delta x = x - x_n$. The same comments regarding the difficulty in computing the elements of D, f, and g obviously apply to computing the perturbation dynamics as well.

In conclusion, the goal of any control scheme is to provide acceptable system performance with an efficient, computationally simple control algorithm. All of the conventional feedback controllers studied

in this chapter fail in this respect. The basic problem is that control schemes based on the complete physical model (e.g., computed torque) are computationally burdensome due to the complexity of computing the parameters of the model. The PID and the nominal-plus-perturbation controllers are based on structurally simpler models. Unfortunately, these schemes sacrifice performance due to the uncertain and time varying model parameters.

In recent times, adaptive control schemes have been developed which provide high performance when the plant model parameters are unknown or even time varying. The resulting algorithms are generally simple enough to be implemented on a microcomputer. In the following chapters, recent results from this field will be applied to the continuous-path tracking problem for manipulators.

CHAPTER 3

ADAPTIVE CONTROL CONCEPTS

The objective of adaptive control is to obtain good closed loop performance when the plant parameters and disturbances are unknown or time-varying. The system parameters may be time-varying due to environmental changes or the use of a linearized model for a nonlinear system. In the latter case, changes in the linear model parameters correspond to changes in the operating point of the system. These perturbations in the plant model parameters can be considered to be parameter disturbances. Thus the goal of adaptive control is to compensate for these parameter disturbances. This contrasts with the goal of conventional feedback/feedforward control, which is to suppress the effect of disturbances on the controlled variable only.

The basic premise upon which all adaptive controllers are designed is that for any possible value of plant parameters and disturbances, there exists a linear controller of fixed structure which yields the desired closed-loop performance. Thus, all adaptive control schemes are based on a conventional feedback controller which would be used if the plant dynamics were known and time invariant.

3.1 Basic Structures of Adaptive Controllers

The general structure of adaptive controllers is shown in Figure 3.1. Some criterion of system performance is measured and compared with

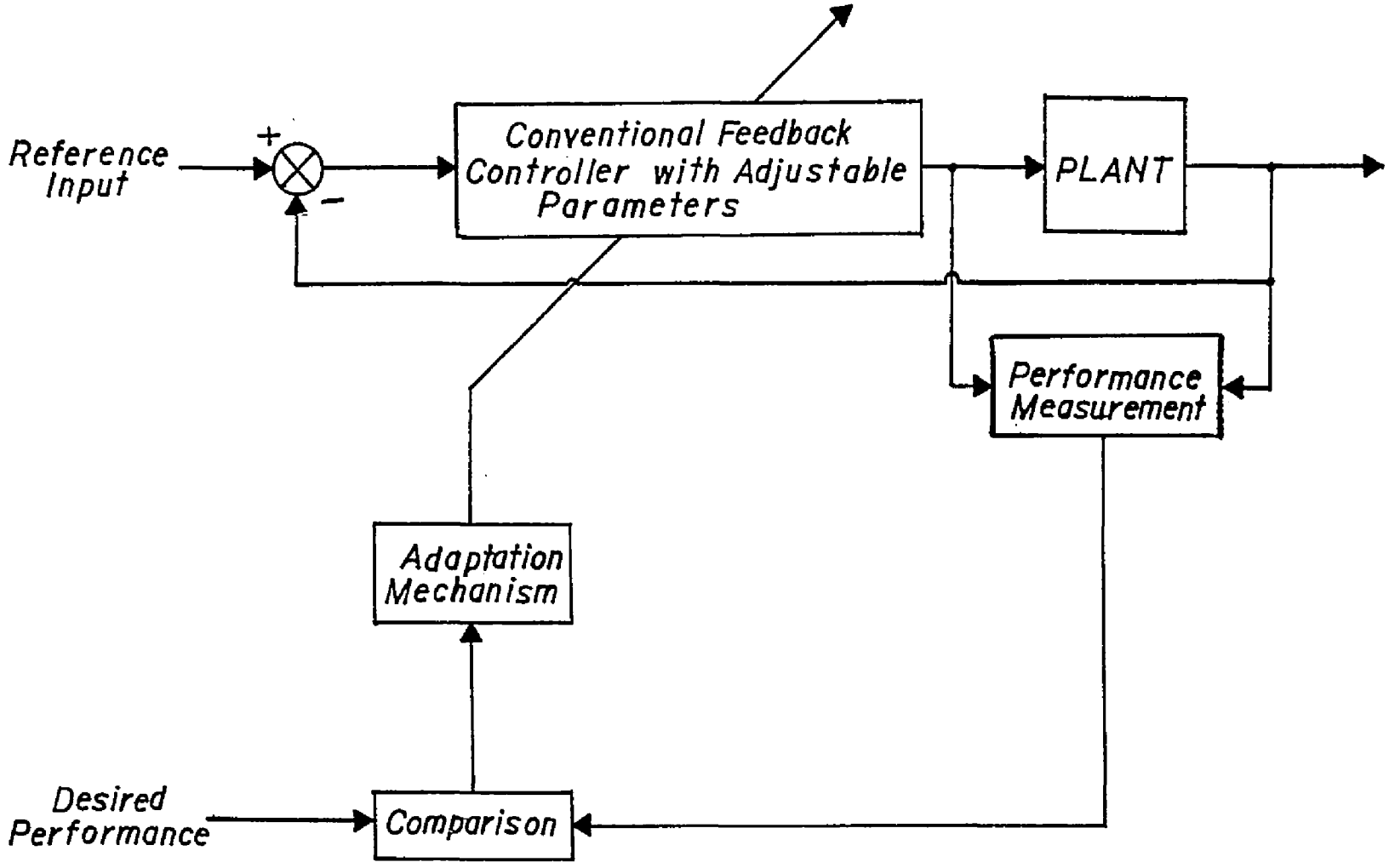

Figure 3.1. General Adaptive Control Structure.

the desired performance. The deviation is input to an adaptation mechanism which modulates the controller parameters. The adjustment mechanism consists of a recursive parameter adaptation algorithm (PAA), which is generally some form of recursive least squares (RLS) estimation.

The adaptation may be done indirectly or directly. Indirect adaptation is a two-step process. In the first step, a PAA uses current and past input/output data to estimate the plant model parameters explicitly. The most recent plant parameter estimates are used in the second step to calculate the control parameters. This process is repeated every sampling period. Applicability to both minimum and non-minimum phase systems and allowing many estimator/controller combinations are advantages of the indirect scheme.

In the direct approach, the PAA adjusts the control parameters directly. The plant model is rewritten to include the controller parameters. The controller parameters are then updated by the PAA as part of the plant estimation. Fewer computations are involved in the direct scheme, but in general, direct schemes can only be applied to minimum phase systems.

Adaptive control theory has been developed from two different points of view--model reference adaptive control (MRAC) and self-tuning controllers (STC). In MRAC, the desired closed-loop performance is stated in terms of an explicit reference model. The reference model may be a transfer function or difference equation. As shown in Figure 3.2, the difference between the actual system output and the reference model

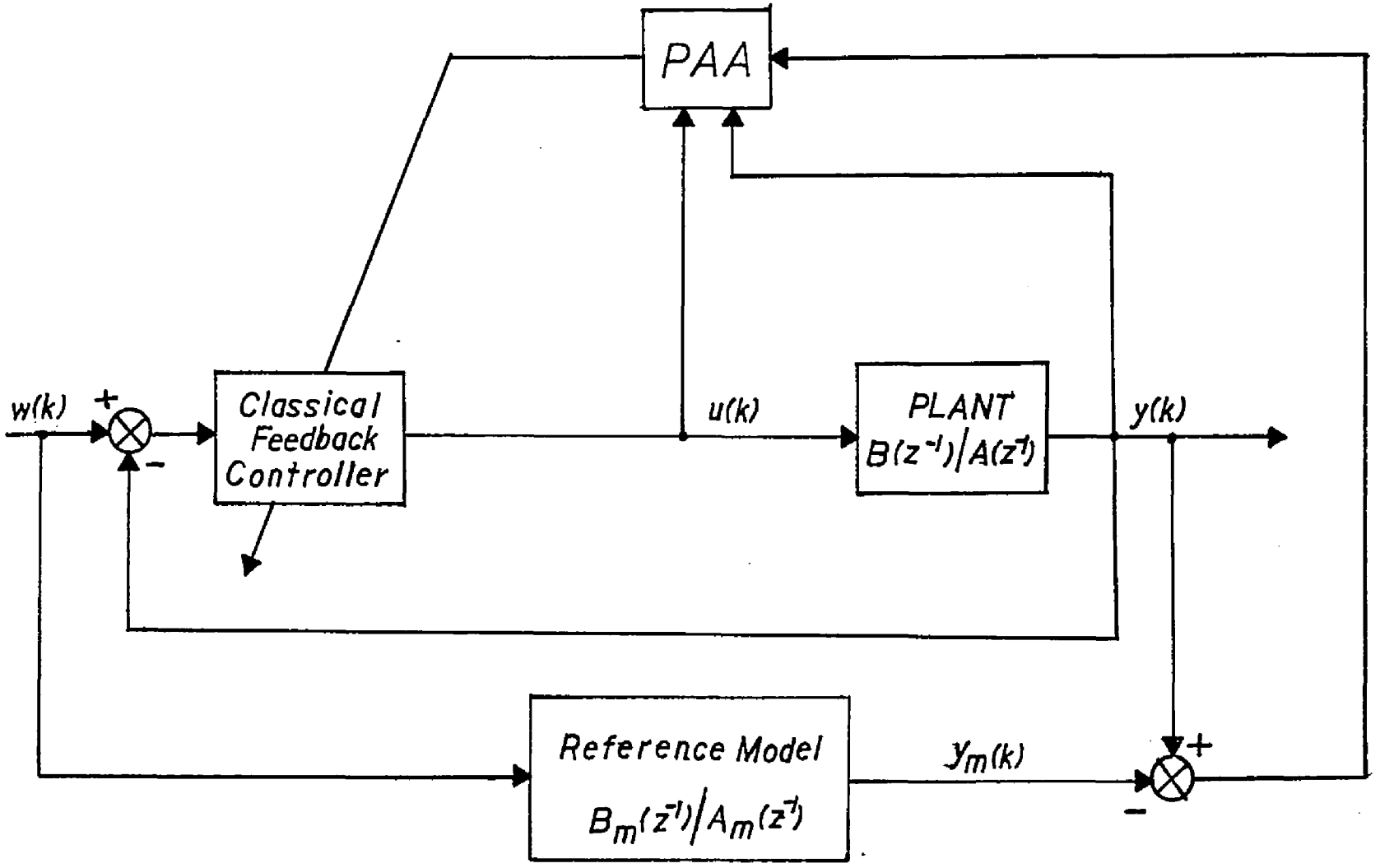

Figure 3.2. Model Reference Adaptive Control Scheme.

(desired) output is input to the PAA. The design goal is to synthesize the PAA such that

$$\lim_{k\to\infty} [y(k) - y_m(k)] = 0 \qquad (3.1.1)$$

This is achieved by cancellation of the open-loop poles and zeros of the plant. This can only be done for minimum phase plants. Indirect MRAC schemes can be used for nonminimum phase plants where it is desired to replace only the poles and stable zeros of the plant.

Self-tuning controllers are a more general class of controllers than MRAC. The control parameters are updated indirectly (Figure 3.3) or directly, based on estimates from an adaptive predictor. The controller may be pole placement, model following, minimum variance, or linear quadratic gaussian. In a deterministic environment, the control algorithms which result from a model following STC and MRAC are very similar (Landau 1981).

3.2 Parameter Adaptation Algorithms

The PAA is the core of all adaptive controllers as it updates the parameters of the adjustable controller in direct schemes and updates the plant model parameter estimates in indirect schemes. For manipulator applications, the following considerations impact the PAA design: (1) the resulting algorithm must require only modest computing power; (2) estimates should converge to the actual plant parameters (indirect) or to the ideal control parameters (direct); (3) time-varying model parameters must be tracked.

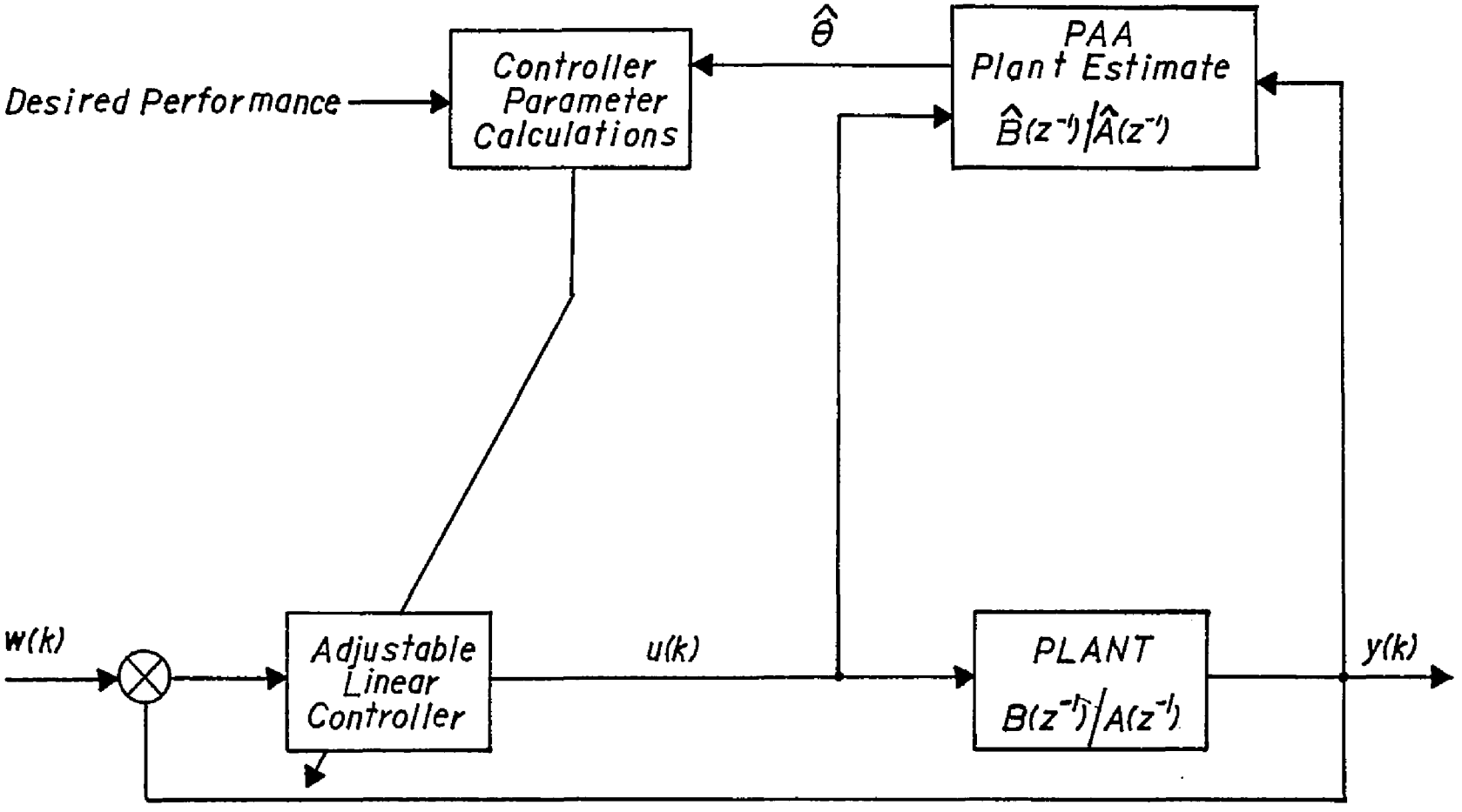

Figure 3.3. Self-Tuning Control Scheme.

Since the estimation must be done in real-time, the algorithms are recursive in nature, and are therefore always implemented digitally. Thus, discrete time models of the continuous plant are obtained with difference equation and transfer function representations being the most popular. Consider a single-input single-output (SISO) continuous plant described by the transfer function:

$$G(s) = \frac{Y(s)}{U(s)} = \frac{B(s)}{A(s)} = \frac{b_o + b_1 s + \ldots + b_{m_c} s^{m_c}}{1 + a_1 s + \ldots + a_{n_c} s^{n_c}} \qquad (3.2.1)$$

A discrete-time representation is obtained by placing an ideal sampler and a zero-order hold in front of the continuous plant. The resulting discrete transfer function yields the following difference equation:

$$y(k+1) = -A(z^{-1})y(k) + B(z^{-1})u(k) \qquad (3.2.2)$$

where

$$A(z^{-1}) = a_1 + a_2 z^{-1} + \ldots a_n z^{-n+1} \qquad (3.2.3)$$

$$B(z^{-1}) = b_o + b_1 z^{-1} + \ldots b_m z^{-m} \qquad (3.2.4)$$

It has been shown (Clark 1981) that many minimum phase, continuous plants will have nonminimum phase discrete-time models if sampled sufficiently fast, thus reducing the number of adaptive control schemes which can be applied to such a system.

Most parameter adaptation algorithms are based on some form of least squares estimation. Consider the SISO prediction model:

$$\hat{y}(k+1) = -\hat{A}(z^{-1})y(k) + \hat{B}(z^{-1})u(k) \tag{3.2.5}$$

where

$\hat{y}(k+1)$ = the predicted output at time k+1 based on information at time k

$y(k)$ = the system output at time k

$u(k)$ = the system input at time k

The polynomials $\hat{A}(z^{-1})$ and $\hat{B}(z^{-1})$ are the estimates of the system polynomials $A(z^{-1})$ and $B(z^{-1})$ of equation (3.2.2) in the case of indirect adaptation or are the estimates of some reparameterized form of equation (3.2.2) in direct adaptation. Equation (3.2.5) may be rewritten as

$$\hat{y}(k+1) = \hat{\theta}^T(k)\phi(k) \tag{3.2.6}$$

where $\hat{\theta}$, the parameter estimation vector, is expressed as:

$$\hat{\theta}^T = [\hat{a}_1\ \hat{a}_2\ \ldots\ \hat{a}_n,\ \hat{b}_o\ \hat{b}_1\ \ldots\ \hat{b}_m] \tag{3.2.7}$$

and $\phi(k)$, the measurement vector, is written:

$$\phi^T(k) = [y(k)\ y(k-1)\ \ldots\ y(k-n+1),\ u(k)\ u(k-1)\ \ldots\ u(k-m)] \tag{3.2.8}$$

In least squares estimation, the criterion:

$$J(k) = \sum_{i=1}^{k} [y(i) - \hat{\theta}^T(k)\phi(i-1)]^2 \tag{3.2.9}$$

is minimized with respect to $\hat{\theta}$. The resulting equations for $\hat{\theta}$ are written recursively (Astrom and Eykhoff 1971) such that the parameter

estimates are updated at each sampling instant. This yields the recursive least squares algorithm:

$$\hat{\theta}(k+1)= \hat{\theta}(k)+ F(k)\phi(k)\varepsilon(k+1) \tag{3.2.10}$$

$$F(k+1)^{-1} = F(k)^{-1} + \phi(k)\phi^{T}(k) \tag{3.2.11}$$

$$F(k+1) = F(k) - \frac{F(k)\phi(k)\phi^{T}(k)F(k)}{1 + \phi^{T}(k)F(k)\phi(k)} \tag{3.2.12}$$

$$\varepsilon(k+1) = \frac{y(k+1) - \hat{\theta}^{T}(k)\phi(k)}{1 + \phi^{T}(k)F(k)\phi(k)} \tag{3.2.13}$$

where $\hat{\theta}(k+1)$ is the new parameter estimate based on measurements obtained up to and including time k+1, $\hat{\theta}(k)$ is the old parameter estimate, F(k) is the adaptation gain, and $\varepsilon(k+1)$ is the adaptation error. Equation (3.2.10) can be interpreted as:

NEW ESTIMATE = OLD ESTIMATE + ADAPTATION GAIN *

ADAPTATION ERROR

The algorithm is started by making an initial guess, $\hat{\theta}(0)$, and taking $F(0) = \alpha I$, where α is a very large scalar. The large adaptation gain produces large initial variations in $\hat{\theta}(k)$. This corresponds to having very little confidence in $\hat{\theta}(0)$. Although the algorithm presented here is for SISO systems, the MIMO version is very similar (Borrison, 1979).

The basic RLS algorithm can be augmented and fine-tuned for various situations. Note from equation (3.2.12) that the adaptation gain, F(k), decreases in time since F(0) is positive definite. This is a desirable feature for systems with constant, unknown parameters, but

is unacceptable for tracking time-varying parameters. A general scheme for updating the adaptation gain is given by:

$$F(k+1) = \frac{1}{\lambda_1(k)} \left[F(k) - \frac{F(k)\phi(k)\phi^T(k)F(k)}{(\lambda_1(k)/\lambda_2(k)) + \phi^T(k)F(k)\phi(k)}\right] \tag{3.2.14}$$

where $\lambda_1(k)$ and $\lambda_2(k)$ are forgetting factors with $0 \leq \lambda_1(k) \leq 1$ and $0 \leq \lambda_2(k) \leq 2$. If $\lambda_1(k) < 1$ and $\lambda_2(k) = 1$, the adaptation algorithm will gradually forget old data, minimizing the criteria (Isermann 1982).

$$J(k) = \sum_{i=1}^{k} \lambda_1^{k-i}(i)[y(i) - \hat{\theta}^T(k)\phi(i-1)]^2 \tag{3.2.15}$$

Thus the most recent estimation errors will have the most weight in updating $\hat{\theta}(k)$.

Recursive least squares is a simple algorithm which gives good estimates in the deterministic environments assumed in this study. The estimates will be substantially biased for noisy systems with signal to noise ratios less than 10, however (Clark 1981). If this is the case, extended least squares or recursive maximum likelihood estimation can be used to obtain unbiased estimates.

3.3 Convergence

As stated previously, the goal of any adaptive controller is to converge to the conventional feedback/feedforward control that would be used if the system model parameters were known exactly. As the adaptive control and plant form a nonlinear time-varying system, the convergence

proofs are generally quite involved. An overview of convergence is presented here, and specific proofs for the algorithms applied to the manipulator problem will be cited in Chapter 4.

All convergence proofs make two basic assumptions. The first is that the real plant structure of equation (3.2.2) is identical to the assumed plant model of equation (3.2.5). The second assumption is that the system is time-invariant. It has been demonstrated through simulations, however, that the second assumption is valid for slowly time-varying systems (Astrom et al. 1977).

For direct MRAC, several schemes have been shown to converge to the desired reference model. In one particular scheme (Lozano and Landau 1981), the question of convergence is transformed to a stability problem by examining the asymptotic behavior of $\varepsilon(k) = y(k) - y_m(k)$. The closed-loop system is shown to satisfy the hyperstability theorem, thus proving that $\lim_{k\to\infty} \varepsilon(k) = 0$.

For indirect self-tuning controllers, convergence is more difficult to show. A sufficient condition for convergence to the desired performance is

$$\lim_{k\to\infty} \hat{\theta}(k) = \theta \tag{3.3.1}$$

Equation (3.3.1) will be satisfied if the following sufficient conditions are met (Isermann 1982):

1. The plant model is characterized by an irreducible transfer function.
2. The order of $A(z^{-1})$, n, and the order of $B(z^{-1})$, m, are known.

3. The steady-state components of u(k) and y(k) are known.

4. The input signal, u(k), is sufficiently rich in frequencies.

Since u(k) is being generated in the closed loop, condition 4 may not be satisfied. If conditions 1-3 are satisfied, this guarantees that

$$\lim_{k\to\infty} [y(k+1) - \hat{y}(k+1)] = 0 \qquad (3.3.2)$$

Thus the asymptotic behavior of the real process and the model output will be the same for the specific input sequence that is applied. In many cases the controller parameters will approach the optimal values if this weaker condition is satisfied (Astrom et al. 1977).

This concludes the overview of adaptive control. In the next chapter, a specific adaptive scheme will be synthesized and applied to the manipulator dynamics formulations developed in Chapter 2.

CHAPTER 4

DESIGN OF A DECENTRALIZED ADAPTIVE CONTROL SCHEME FOR MANIPULATORS

A desired objective of the manipulator control system is to provide independent joint control (IJC). As discussed in Chapter 2, the dynamical interactions between the joints and the varying effective inertia must be effectively compensated by the control system before independent joint control can be achieved. In this chapter, the adaptive control concepts of Chapter 3 are applied to the IJC problem. The result is a decentralized, adaptive control scheme which features on-line adaptation of the varying effective inertia and feedforward compensation to eliminate the dynamic coupling between joints. This scheme is then compared to earlier applications of adaptive control to the manipulator dynamics problem. The comparison includes a centralized MRAC scheme proposed by Horowitz and Tomizuka (1983), Lee and Chung's proposed nominal plus adaptive perturbation control scheme (1982), and finally, an adaptive scheme based on an autoregressive model of the manipulator dynamics (Koivo and Guo 1983).

4.1 Decentralized Control

The distributed system representation of Figure (2.3) is used. This model views the overall system (actuator dynamics plus manipulator dynamics) as a set of subsystems (individual actuators) with disturbance

torques acting on each subsystem. These disturbance torques are the interactions from the manipulator dynamics coupling terms. That is, for subsystem i, the disturbance torque T_{L_i} is,

$$T_{L_i} = P_i/n_i \tag{4.1.1}$$

where

$$P_i = \sum_{\substack{j=1 \\ j\neq i}}^{N} d_{ij}\ddot{y}_j + f_i(y,\dot{y}) + g_i(y) \tag{4.1.2}$$

and y is the vector of joint displacements, the other terms having been defined in Chapter 2. This subsystem model is shown in Figure (4.1). Note the presence of the nonlinear disturbance torque, T_{L_i}, given by equations (4.1.1) and (4.1.2) and the nonlinear total inertia term given by equation (2.1.14).

This distributed model lends itself to the design of a decentralized control scheme, in which case there is an autonomous controller dedicated to each joint. Such a scheme has definite advantages over a centralized controller. Each control algorithm requires fewer computations than the single algorithm, which results from a centralized, MIMO control strategy. By using distributed processing techniques, the computational delay of the decentralized controllers is much less than the delay for the centralized controller, thus enhancing performance.

4.2 Control Scheme

As observed in Chapter 3, direct MRAC techniques are, in general, stable only for minimum phase systems. When the subsystem of Figure (4.1)

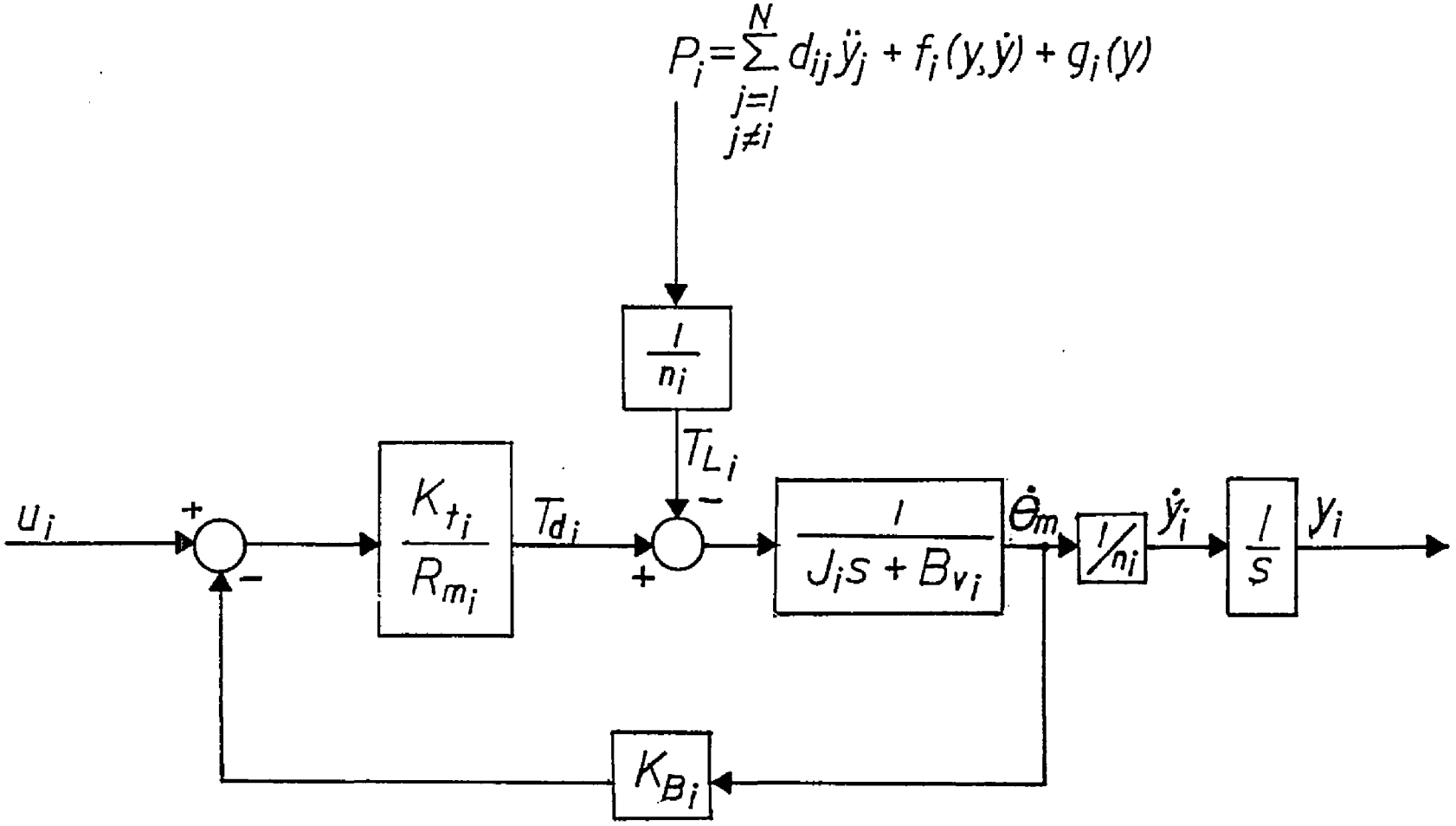

Figure 4.1. Explicit Subsystem Model.

is linearized and discretized, the open loop transfer function, G(z) = Y(z)/U(z), is found to contain a zero extremely close to the unit circle for all operating conditions of the manipulator. The transfer function $G_v(z) = \dot{Y}(z)/U(z)$ has no zeroes, however. For this reason, MRAC is only used to control the inner velocity loop. The reference velocities, $\dot{y}_d(k)$, are provided by a proportional position controller.

$$\dot{y}_d(k+1) = \dot{y}_r(k+1) + c[y_r(k) - y(k)] \tag{4.2.1}$$

where

$\dot{y}_d$ = reference velocity input to adaptive controller

$\dot{y}_r$ = reference velocity from path planner (computed a priori)

y_r = reference position input from path planner

y = actual joint position

c = proportional gain ($c > 0$)

The structure of this adaptive controller is, of course, dependent on the structure of the linearized, discretized subsystem dynamics. The nonlinear differential equation which describes the input/output behavior of the ith subsystem is:

$$J_i\ddot{y}_i(t) + [B_{v_i} + (K_{t_i}/R_{m_i})K_{B_i}]\dot{y}_i(t) =$$

$$[K_{t_i}/(n_i R_{m_i})]u_i(t) - (1/n_i^2)P_i \tag{4.2.2}$$

Note that this equation contains the nonlinear inertia term, $J_i = J_{m_i} + (1/n_i^2)d_{ii}$, and the nonlinear coupling terms of equation (4.1.2). The linearization and discretization of equation (4.2.2), as detailed in Appendix A, yields the following difference equation,

$$\dot{y}_i(k+1) = -a_1\dot{y}_i(k) + b_o u_i(k) + \sum_{j=1}^{N} C_j(z^{-1})v_j(k) \qquad (4.2.3)$$

where

$$a_1 = -e^{-\beta T} \qquad (4.2.4)$$

$$b_o = \alpha/\beta(1-e^{-\beta T}) \qquad (4.2.5)$$

$$\alpha = K_{t_i}/(J_i R_{m_i} n_i) \qquad (4.2.6)$$

$$\beta = (1/J_i)[B_{v_i} + (K_{t_i} K_{B_i})/R_{m_i}] \qquad (4.2.7)$$

and $v_j(k) = y_j(k)$ or $v_j(k) = \dot{y}_j(k)$. The last term on the right-hand side of equation (4.2.3) represents the coupling torques as a moving average of the other subsystem outputs.

The justification for using a linear model for the subsystem dynamics is as follows. If the adaptation speed of the controller is fast compared to the variation of the nonlinear terms, the nonlinear system may be treated as a linear, time-varying system. In other words, changes in the linear model coefficients correspond to changes in the operating point of the nonlinear system. Thus, the closed loop system will be stable if the parameter adaptation algorithm coverges quickly. Otherwise, performance will be sacrificed. Therefore, the adaptation rate will be of utmost importance to the success of this control scheme.

As seen by equation (4.2.3), the coupling terms are represented by a moving average of either the other joint positions or velocities. This is done to represent accelerations in terms of finite differences of either positions or velocities. Three different subsystem models are considered. They only differ in the way in which the coupling terms are

represented. Model 1 represents all the coupling terms as moving averages of the joint positions. For a three degree of freedom manipulator, model 1 is,

$$\dot{y}_i(k+1) = -a_1\dot{y}_i(k) + b_o u_i(k) + C_1(z^{-1})y(k) + C_2(z^{-1})y_2(k) + C_3(z^{-1})y_3(k) \qquad (4.2.8)$$

where

$$C_j(z^{-1}) = c_{j_o} + c_{j_1} z^{-1} + c_{j_2} z^{-2} \qquad j = 1, 2, 3 \qquad (4.2.9)$$

The coupling terms are lumped into one parameter for model 2. For a three degree of freedom manipulator, model 2 is,

$$\dot{y}_i(k+1) = -a_1\dot{y}_i(k) + b_o u_i(k) + h_i \qquad (4.2.10)$$

In model 3, coupling inertias and velocity terms are represented by moving averages of the other joint velocities. All gravity terms are lumped in the parameter, h_i. For the first joint of a three joint manipulator, the model 3 representation is,

$$\dot{y}_1(k+1) = -a_1\dot{y}_1(k) + b_o u_1(k) + c_{2_o}\dot{y}_2(k) + c_{2_1}\dot{y}_2(k-1) + c_{3_o}\dot{y}_3(k) + c_{3_1}\dot{y}_3(k-1) + h_1 \qquad (4.1.11)$$

Model 1 is the most complex and accurate representation, followed by model 3, and then model 2 in the degree of simplicity. Note that all the linearized subsystem models are of the form,

$$A_i(z^{-1})\dot{y}_i(k+1) = B_i(z^{-1})u_i(k) + \sum_{j=1}^{N} C_j(z^{-1})v_j(k) + h_i \tag{4.2.12}$$

where

$$A_i(z^{-1}) = 1 + a_{i_1} z^{-1} + \ldots + a_{i_n} z^{-n} = 1 + z^{-1}A_i^*(z^{-1}) \tag{4.2.13}$$

$$B_i(z^{-1}) = b_{i_o} + b_{i_1} z^{-1} + \ldots + b_{i_m} z^{-m} = b_{i_o} + z^{-1}B_i^*(z^{-1}) \tag{4.2.14}$$

$$C_j(z^{-1}) = c_{j_o} + c_{j_1} z^{-1} + \ldots + c_{j_\ell} z^{-\ell} \tag{4.2.15}$$

$$v_j(k) = y_j(k) \quad \text{or} \quad v_j(k) = \dot{y}_j(k) \tag{4.2.16}$$

This general model is used in the control law development which follows.

4.3 Development of the Control Algorithm

The adaptive control algorithm is based on a scheme developed by Lozano and Landau (1981). Lozano and Landau have proved that their scheme is asymptotically stable under the assumptions that the plant is linear and time-invariant, with no disturbances present.

For the manipulator application, adaptive feedforward compensation has been added to null the effects of gravity and dynamic interactions between joints. Except for the addition of this feedforward compensation, the algorithm derivation follows very closely with Lozano and Landau's development.

The plant dynamics are given by equations (4.2.12) through (4.2.16). All model reference adaptive control schemes are based on a linear model following controller (LMFC) that would be used if the plant

parameters were known. Therefore temporarily assume that the A_i, B_i, and C_j polynomials are known in order to find the LMFC. The control objective is for the velocity tracking error to decay with the dynamics of a designer specified polynomial, $C_{R_i}(z^{-1})$. This is expressed as,

$$C_{R_i}(z^{-1})[\dot{y}_i(k+1) - \dot{y}_{d_i}(k+1)] = 0 \tag{4.3.1}$$

where

$$C_{R_i}(z^{-1}) = 1 + c^*_{R_{i_1}} z^{-1} + \ldots + c_{R_{i_n}} z^{-n}$$

$$= 1 + z^{-1} C_{R_i}(z^{-1}) \tag{4.3.2}$$

The C_{R_i} polynomial may have very fast dynamics (e.g., $C_{R_i}(z^{-1}) = 1$), in which case a deadbeat response will be achieved, or it may have highly damped roots, in which case smoother response is obtained with lower control effort. Adding $C_{R_i}\dot{y}_i(k+1)$ to both sides of equation (4.2.12) and defining a new polynomial, $R_i(z^{-1}) = C^*_{R_i}(z^{-1}) - A^*_i(z^{-1})$, the plant dynamics may be rewritten in terms of the control specification as,

$$C_{R_i}\dot{y}_i(k+1) = B_i u_i(k) + R_i\dot{y}_i(k) + \sum_{j=1}^{N} C_j v_j(k) + h_i \tag{4.3.3}$$

By introducing a parameter vector, θ_i, and a measurement vector $\phi_i(k)$ defined as,

$$\theta_i = [b_{i_o} \ldots, b_{i_m}, r_{i_o} \ldots, r_{i_{n-1}}, c_{1_o} \ldots, c_{1_\ell}$$

$$c_{N_o} \ldots, c_{N_\ell}, h_i]^T \tag{4.3.4}$$

$$\phi_i(k) = [u_i(k) \ , \ . \ . \ u_i(k-m), \ y_i(k) \ , \ . \ . \ y_i(k-n+1),$$

$$v_1(k) \ , \ . \ . \ v_1(k-\ell) \ , \ . \ . \ v_N(k) \ , \ . \ .$$

$$v_N(k-\ell), \ 1] \qquad (4.3.5)$$

the plant equation, (4.3.3), can be rewritten using parameter estimation notation:

$$C_{R_i}\dot{y}_i(k+1) = \theta_i^T\phi_i(k) \qquad (4.3.6)$$

Since the control objective is given by equation (4.3.1), the control equation can be obtained by simply replacing the left-hand sides of equations (4.3.3) and (4.3.6) with $C_{R_i}y_{d_i}(k+1)$. Solving for $u_i(k)$, the control which satisfies the objective of equation (4.3.1) is,

$$u_i(k) = (1/b_{i_o})[C_{R_i}\dot{y}_{d_i}(k+1) - B_i^*u_i(k-1) - R_i\dot{y}_i(k)$$

$$- \sum_{j=1}^{N} C_jv_j(k) - h_i] \qquad (4.3.7)$$

or in parameter estimation notation, the control equation may be expressed as,

$$C_{R_i}\dot{y}_{d_i}(k+1) = \theta_i^T\phi_i(k) \qquad (4.3.8)$$

In conclusion, if the A, B, and C polynomials, and the disturbance parameter h are known exactly, the control objective is identically satisfied.

Equation (4.3.7) is the control which would be used if the

parameters are known. The subsystem parameters are unknown, however, and thus the control objective is relaxed:

$$\lim_{k\to\infty} C_{R_i}(z^{-1})[\dot{y}_i(k+1) - \dot{y}_{d_i}(k+1)] = 0 \tag{4.3.9}$$

The plant dynamics of equation (4.3.6) remain the same, but the actual control equations are now functions of the parameter estimates. Thus the control becomes,

$$u_i(k) = (1/\hat{b}_{i_o})[C_{R_i}\dot{y}_{d_i}(k+1) - \hat{B}_i^* u_i(k-1) - \hat{R}_i y_i(k) - \sum_{j=1}^{N} \hat{C}_j v_j(k) - \hat{h}_i] \tag{4.3.10}$$

or

$$C_{R_i}\dot{y}_{d_i}(k+1) = \hat{\theta}_i^T(k)\phi_i(k) \tag{4.3.11}$$

where the notation "^" denotes an estimated value.

Having defined the basic control law, an adaptation error and a parameter adaptation algorithm must be selected. The adaptation error, $\varepsilon^o(k)$, is given by,

$$\begin{aligned}\varepsilon_i^o(k) &= C_{R_i}[\dot{y}_i(k) - \dot{y}_{d_i}(k)] \\ &= [\theta_i - \hat{\theta}_i(k-1)]^T\phi_i(k-1)\end{aligned} \tag{4.3.12}$$

The PAA used to update the parameter vector is known as recursive least squares with forgetting factor (Isermann 1982). It is obtained from the general PAA of equation (3.2.14) by setting $\lambda_2(k) = 1$. This algorithm minimizes the performance index,

$$J_i(k) = \sum_{j=0}^{k} \lambda_i^{k-j}(j)\{[\theta_i - \hat{\theta}_i(k)]^T \phi_i(j-1)\}^2 \qquad (4.3.13)$$

where $0 < \lambda_i < 1$, $i = 1, \ldots, N$

Thus, the algorithm given by

$$\hat{\theta}_i(k) = \hat{\theta}_i(k-1) + F_i(k-1)\phi_i(k-1)\epsilon_i(k)/$$

$$[1 + \phi_i^T(k-1)F_i(k-1)\phi_i(k-1)] \qquad (4.3.14)$$

$$F_i(k) = (1/\lambda_i(k-1))[F_i(k-1) - F_i(k-1)\phi_i(k-1)\phi_i^T(k-1)F_i(k-1)/$$

$$\{\lambda_i(k-1) + \phi_i^T(k-1)F_i(k-1)\phi_i(k-1)\}] \qquad (4.3.15)$$

exponentially forgets old input/output data in making the new parameter estimate. The smaller the parameter λ_i is, the faster the algorithm forgets. Since the system parameters being estimated are time varying, this forgetting factor is essential as old input/output data is not indicative of the behavior of the system at the present time.

The control algorithm sequence is shown in Figure (4.2). First, the input/output measurements are sampled and along with old input/output data, form the measurement vector, $\phi(k)$. The adaptation error, $\epsilon^o(k)$, is computed as per equation (4.3.12). The parameter vector is updated by implementing equation (4.3.14). The control is then updated by equation (4.3.10). Finally, the adaptation gain, F(k), is updated for use in the next sampling period by implementing equation (4.3.15).

To begin execution of the algorithm, an initial estimate of the parameter vector, $\hat{\theta}_i(0)$, must be made. If $\hat{\theta}_i(0)$ deviates signifi-

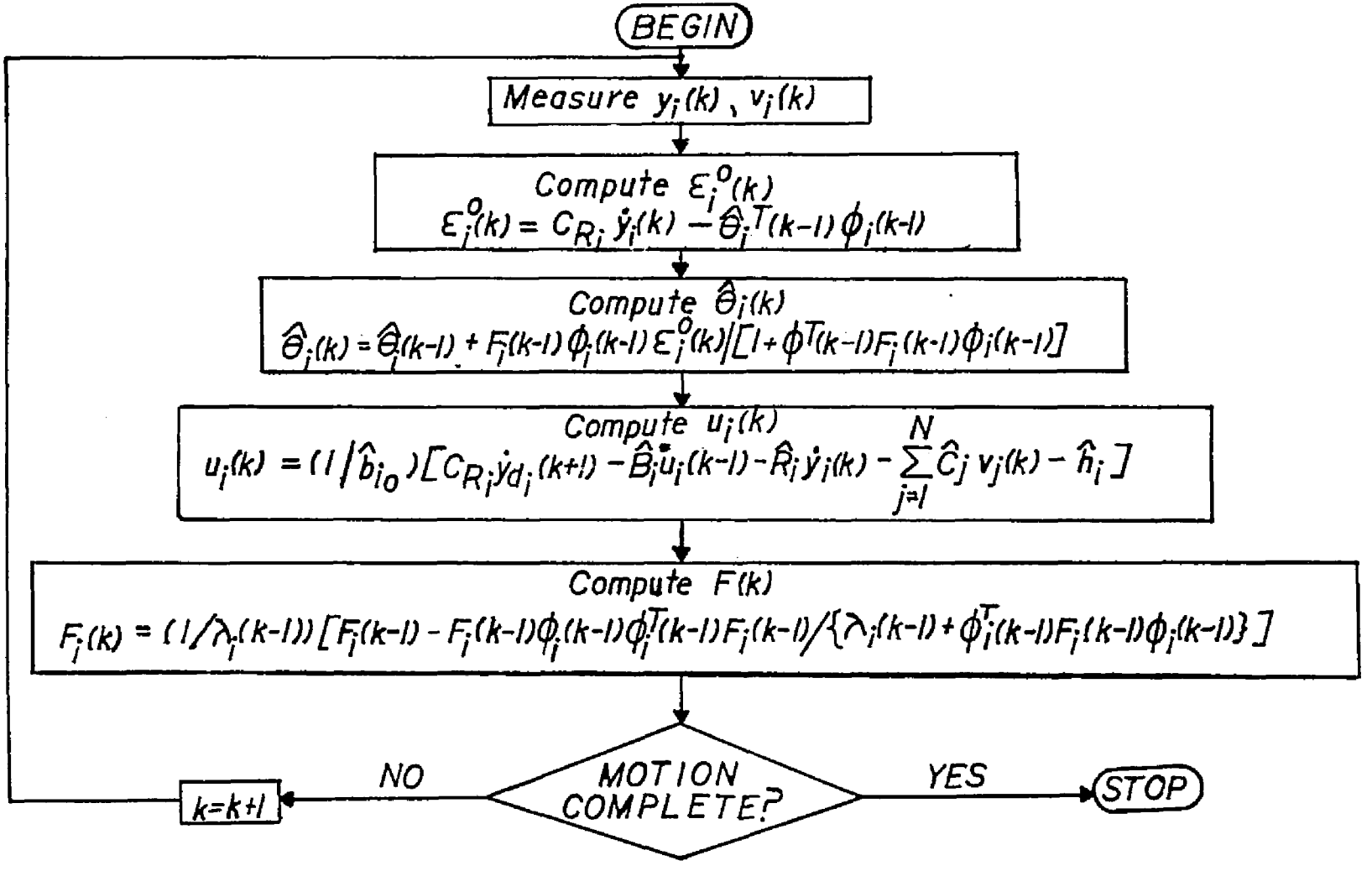

Figure 4.2. Control Algorithm Sequence.

cantly from the actual parameter vector $\theta_i(0)$, instability or at the very least, degraded performance will result during the initial part of the trajectory. To avoid this, $a_1(0)$ and $b_o(0)$ can be calculated from equations (4.2.4) through (4.2.7) using crude approximations for the inertia and motor parameters. Alternately, a small amplitude learning signal can be applied to the system before any useful work is performed. This will allow $\hat{\theta}$ to converge to the proper values before the trajectory of interest begins.

A block diagram of the adaptive closed loop subsystem is shown in Figure (4.3). A diagram of the overall system showing the global information necessary to implement the control for model 1 (equation (4.2.8)) is shown in Figure 4.4. Note that only local joint velocity measurements must be made available, while both local and global joint position measurements must be available. Thus, for a six-degree-of-freedom manipulator, six signals must be available on the global bus.

The algorithm described by equations (4.3.10), (4.3.12), (4.3.14), and (4.3.15) is general. By using the subsystem models 1, 2 and 3, represented by equations (4.2.8), (4.2.10), and (4.2.11), respectively, three different control algorithms of varying complexities are generated. They will henceforth be referred to as controls 1, 2, and 3. The performance of these three algorithms is tested and analyzed in the next chapter.

4.4 Comparison with Previous Adaptive Schemes

How does this scheme compare with previous attempts to apply adaptive control to the manipulator dynamics problem? Horowitz and

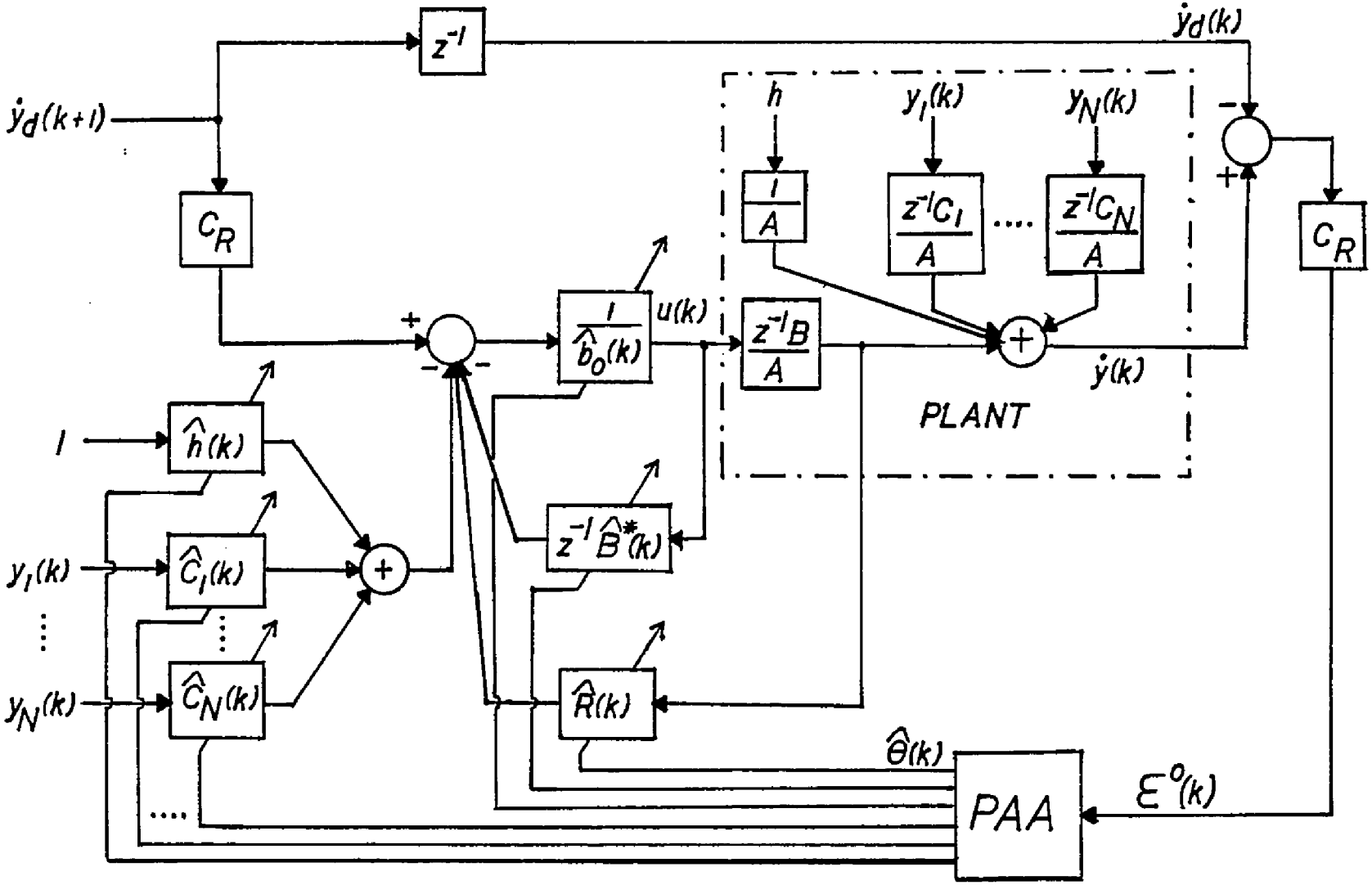

Figure 4.3. Adaptive Closed Loop Subsystem.

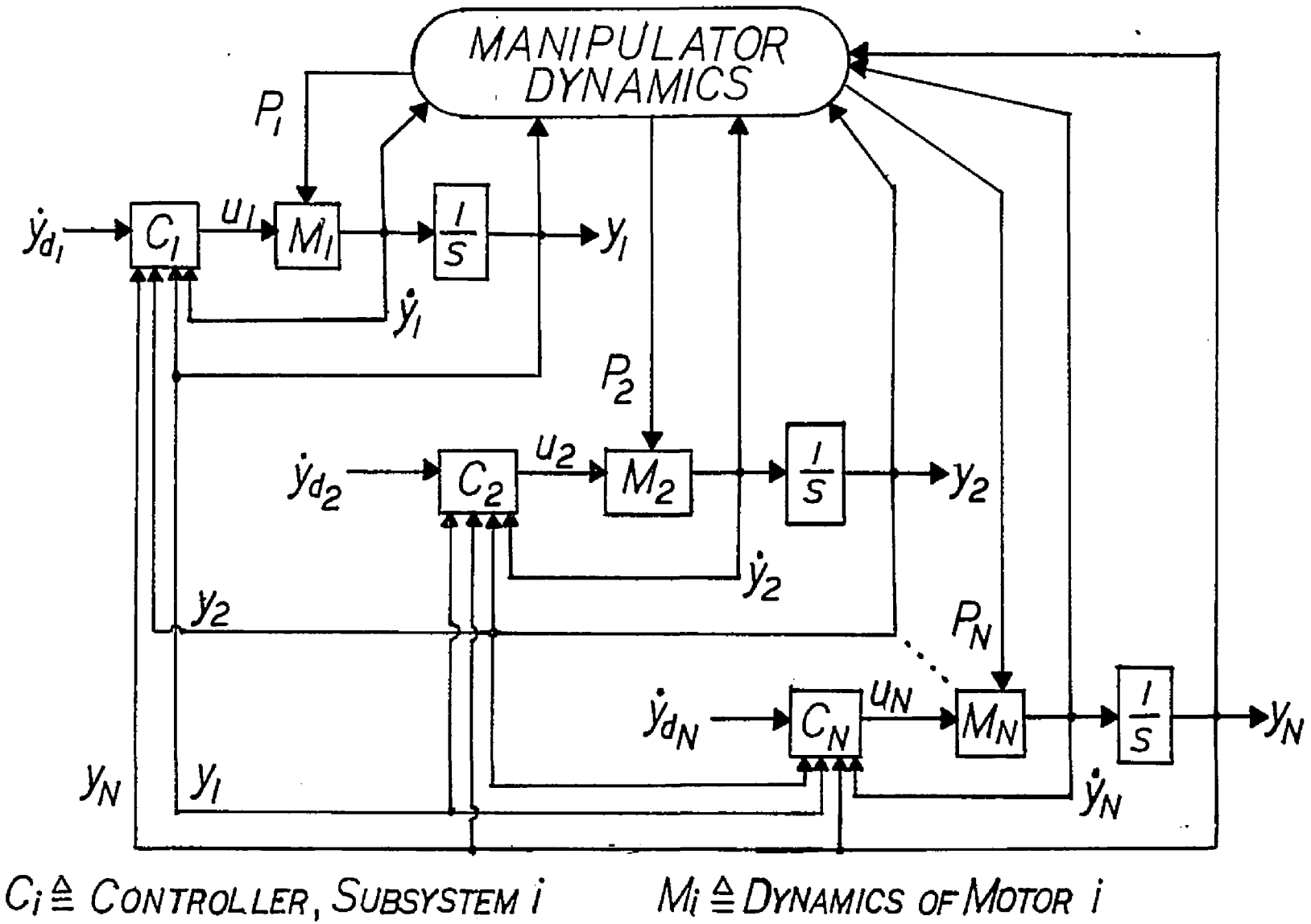

Figure 4.4. Decentralized Manipulator Control System.

Tomizuka (1983) proposed a centralized, direct, MRAC scheme. It is based on a state variable, MIMO model of the manipulator dynamics. The control has all the advantages and disadvantages of centralized controllers as discussed in Section 1.1. In developing the control law, Horowitz and Tomizuka include the effect of the computational delay which is inherent in all digital control systems. The control variables computed by this scheme are the joint torques. Equation (2.1.12) must be implemented to compute the actuator voltages. Thus, unlike the control scheme developed in this thesis, the motor dynamics are left out of the adaption loop and any significant motor parameter drift will cause tracking errors. In terms of the number of parameters which must be estimated, Horowitz and Tomizuka's scheme is very efficient. For an N jointed manipulator, N^2+N parameters are estimated.

A nominal plus perturbation control scheme has been proposed by Lee and Chung (1982). It is conceptually similar to the nominal plus perturbation control scheme discussed in Section 2.2.3. An indirect adaptive control regulates the perturbation dynamics of Equation (2.2.20). The computed torque technique generates the nominal control signal. A linear model can more accurately represent the perturbation dynamics than the full dynamics of the manipulator. Thus Lee and Chung's scheme is expected to be more robust and more tolerant of adaptation sluggishness than the scheme developed here. This performance gain is offset by implementation cost, as both the computed torque (nominal) algorithm and the adaptive (perturbation) algorithm must be executed in real time.

Koivo and Guo (1983) developed an indirect self-tuning controller based on an autoregressive model of the manipulator dynamics. They developed this scheme within both a centralized and decentralized framework. The centralized control is based on a MIMO model, while the decentralized control is based on N independent SISO models (one for each joint). Unlike the adaptive control developed in this thesis, Koivo and Guo's decentralized scheme completely ignores the dynamic coupling between the joints; i.e., feedforward compensation is not used. Surprisingly, Koivo and Guo were unable to produce any simulation results in which the centralized controller, which models the dynamic coupling between joints, outperformed the decentralized control. In the next chapter, the simulation results will show that there are, in fact, trajectories in which the coupling torques become quite significant. In such cases, the models must include an adequate representation of the dynamic coupling terms for precision tracking to be realized.

CHAPTER 5

PERFORMANCE ANALYSIS OF THE CONTROL ALGORITHMS

The purpose of the computer simulation study conducted here is to answer two questions concerning the performance of the present manipulator control scheme. Firstly, does this adaptive scheme give good high speed tracking performance under the influence of coupling torques, rapid variations of effective inertia, and when manipulating objects of unknown inertia properties? Secondly, how does the choice of the linearized model (equations (4.2.8), (4.2.10), (4.2.11)), upon which the control is based, affect tracking performance? The first question is answered by examining the tracking errors during high speed, straight line trajectories, i.e., when the end effector traces a straight line in cartesian space during the motion. The second question is studied by examining the tracking errors of the controllers based on models 1, 2, and 3 during tracking of various trajectories. These trajectories are selectively designed to emphasize one dynamic effect at a time (e.g., coupling inertia disturbance only, or rapid variation of effective inertia only).

5.1 Description of Simulation Environment

The simulation is a set of Fortran routines executed on VAX 11/780 hardware. The physical and dynamic parameters of the first three links of the Stanford manipulator (Paul 1981) are used. Pertinent

details of the Stanford manipulator are given in Appendix B. As the last three joints (roll, pitch, and yaw wrist joints) are not as demanding of the control system as the first three, they were neglected to save computer time. Figure (5.1) shows the configuration of the Stanford manipulator. Of the manipulator dynamics, only the rigid body modes are simulated; flexible modes are assumed negligible. Other assumptions include negligible inductance effects in the motors and only dynamic friction present in the joints.

The simulation consists of a control section and a dynamics section. Implementing the algorithm of the previous chapter, the control section computes the armature voltages with an assumed sampling period of 10 milliseconds. The control signals are passed to the dynamics section which uses the Newton-Euler technique (Luh et al. 1980; Walker and Orin 1982) to calculate the dynamic coefficients. Equations (2.1.1) are then numerically integrated using a fourth order Runge-Kutta routine with an integration step size of 2.5 milliseconds.

5.2 High Speed Straight Line Motion

A straight line motion of the gripper in cartesian space often transforms into complex and demanding joint trajectories. In this particular simulation run, the end-effector tracks a straight line from (0.0, -0.5, -0.5) to (0.325, 0.4, 0.5) in the X_o, Y_o, Z_o coordinate system (units in meters). This 1.5 second motion maps into the joint trajectories of Figure (5.2). These trajectories are characterized by large joint displacements, continuous acceleration profiles, and large

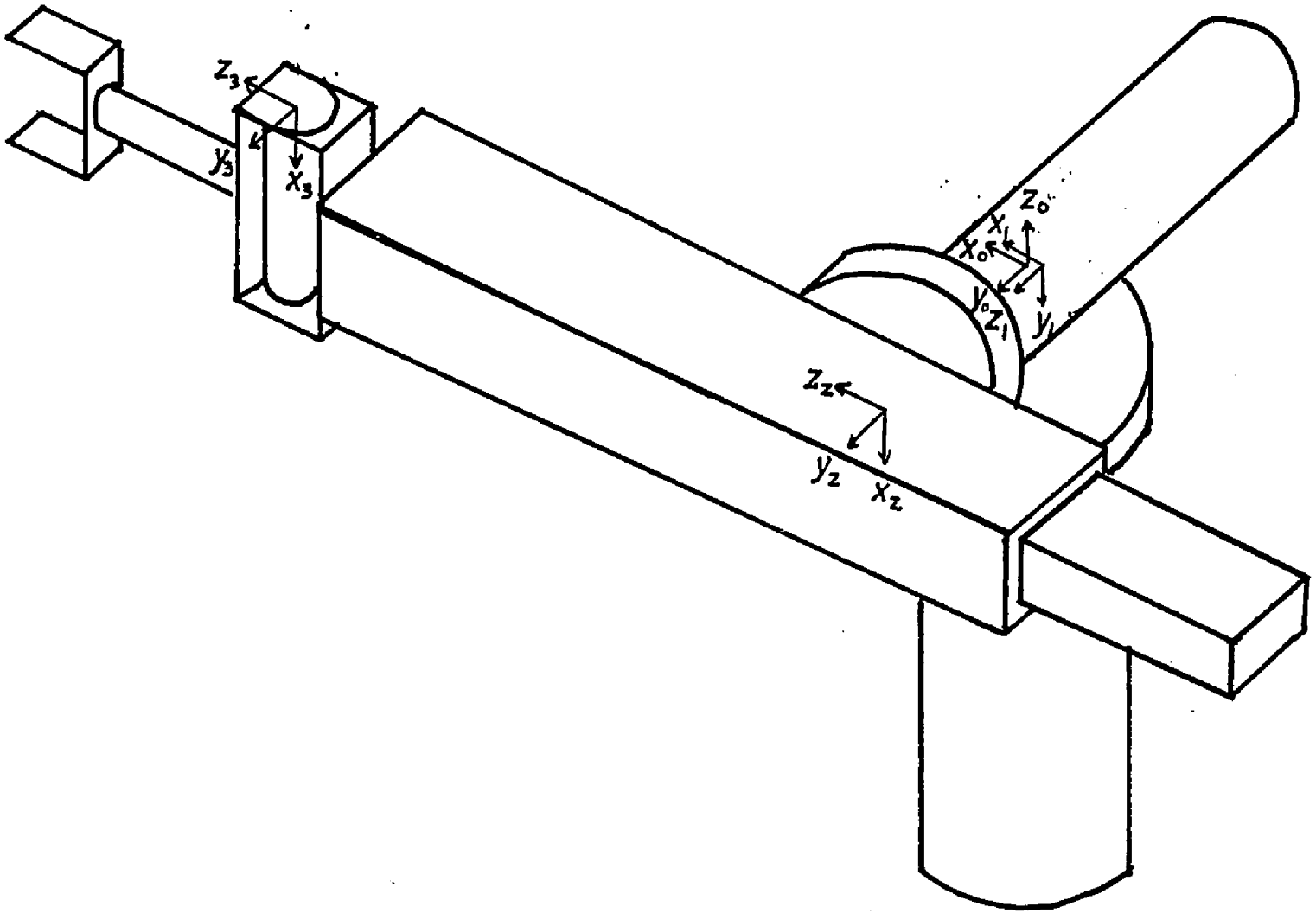

Figure 5.1. The Stanford Manipulator.

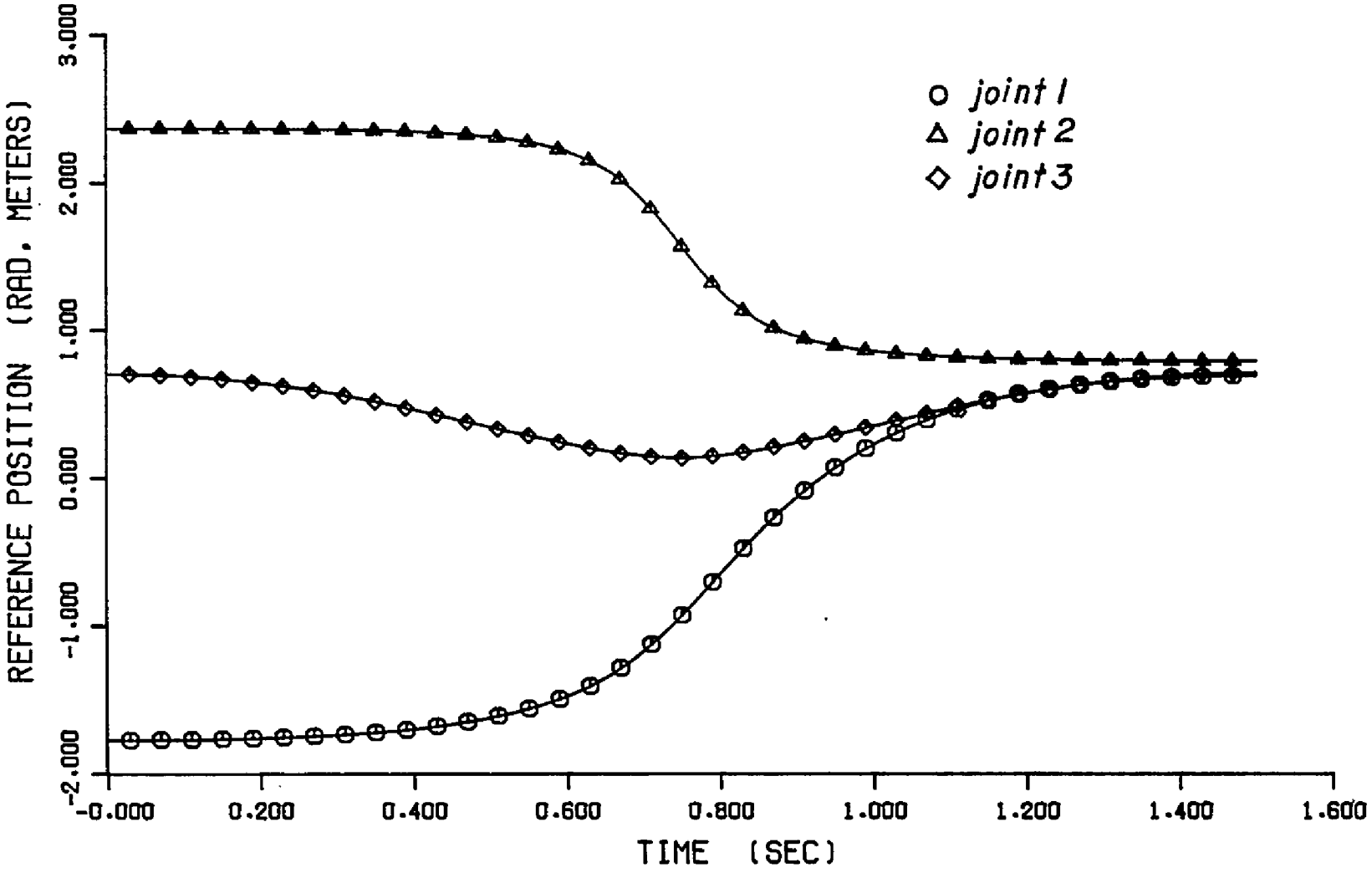

Figure 5.2. Joint Trajectories for a Straight Line Motion.

velocities for $0.70 < t < 0.90$ seconds. The effective inertias along the trajectory for joint 1 and joint 2 are shown in Figure (5.3). Note the large variations during the period, $0.6 < t < 0.9$ seconds. This is the period when most demand is placed on the control system.

The position error profiles for the controls based on models 1, 2, and 3 are shown in Figures (5.4), (5.5), and (5.6), respectively. After an initial learning period, all 3 algorithms yield satisfactory performance. Control 2, based on the least accurate model but with the least number of parameters to estimate, gives the best performance. This result is, in general, true for other straight line trajectories and various loading conditions. The underlying reasons for this paradox will be postulated in Section 5.5.

5.3 Manipulation of Unknown Masses

A common manipulator task is to approach an object, grasp the object, and then move to a new position. There is, of course, a discontinuity in inertia and gravity forces at the instant the object is grasped. The control must adapt to these new dynamic parameters as quickly as possible.

To simulate this situation, the unloaded manipulator is moved in a 0.65 second, straight line trajectory from (0.1, -0.5, -0.5) to (1.1, 0.1, 0.0) in the base coordinate system. The gripper is held at this position until $t = 0.85$ seconds. At the halfway point of this hover ($t = 0.75$ sec.) the mass and inertia parameters of link 3 are changed to represent the grasping of a 5.4 kg. cube. The trajectory is then completed as the arm is moved back to the original position.

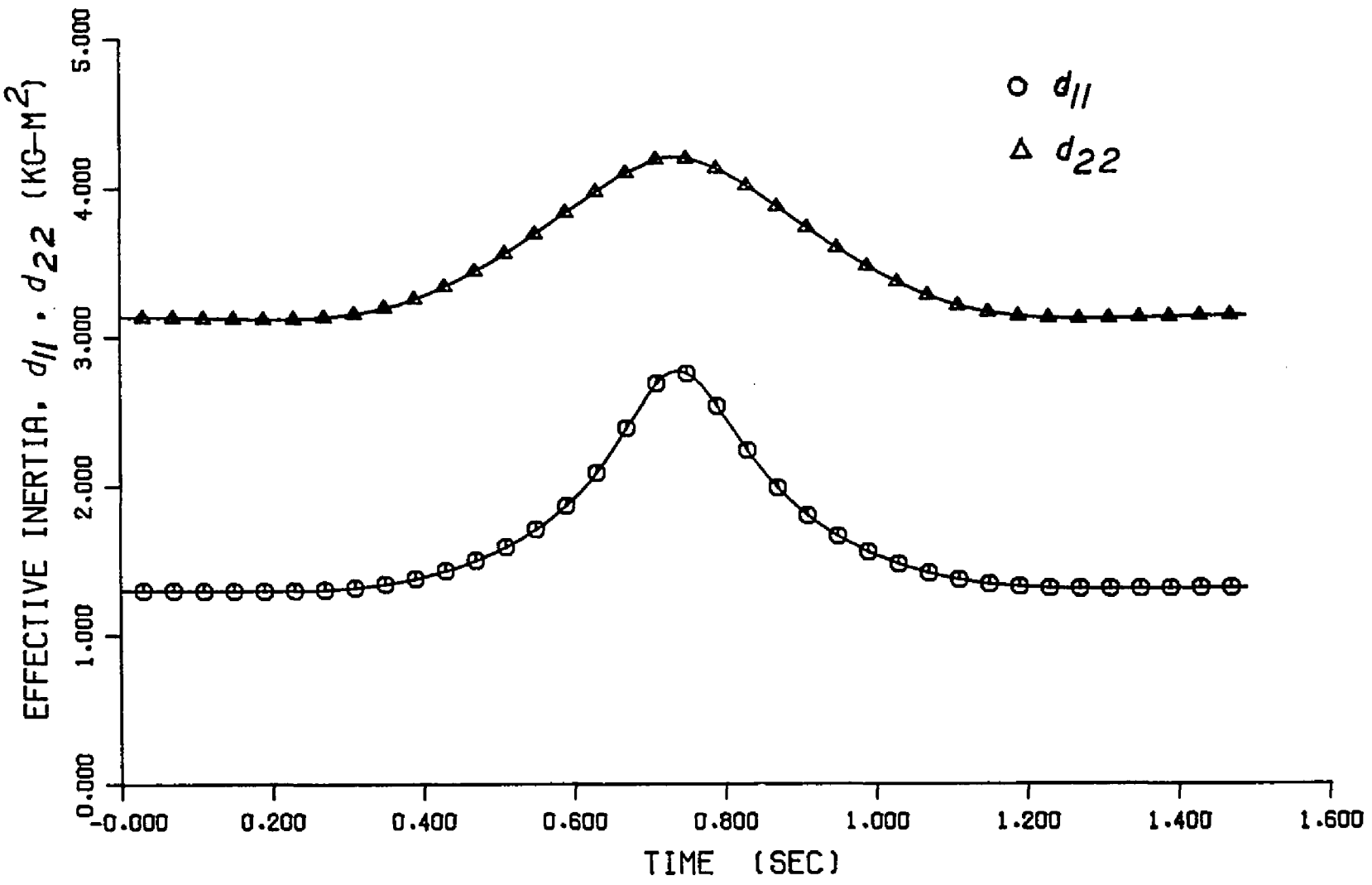

Figure 5.3. Effective Inertia Along the Straight Line Trajectory.

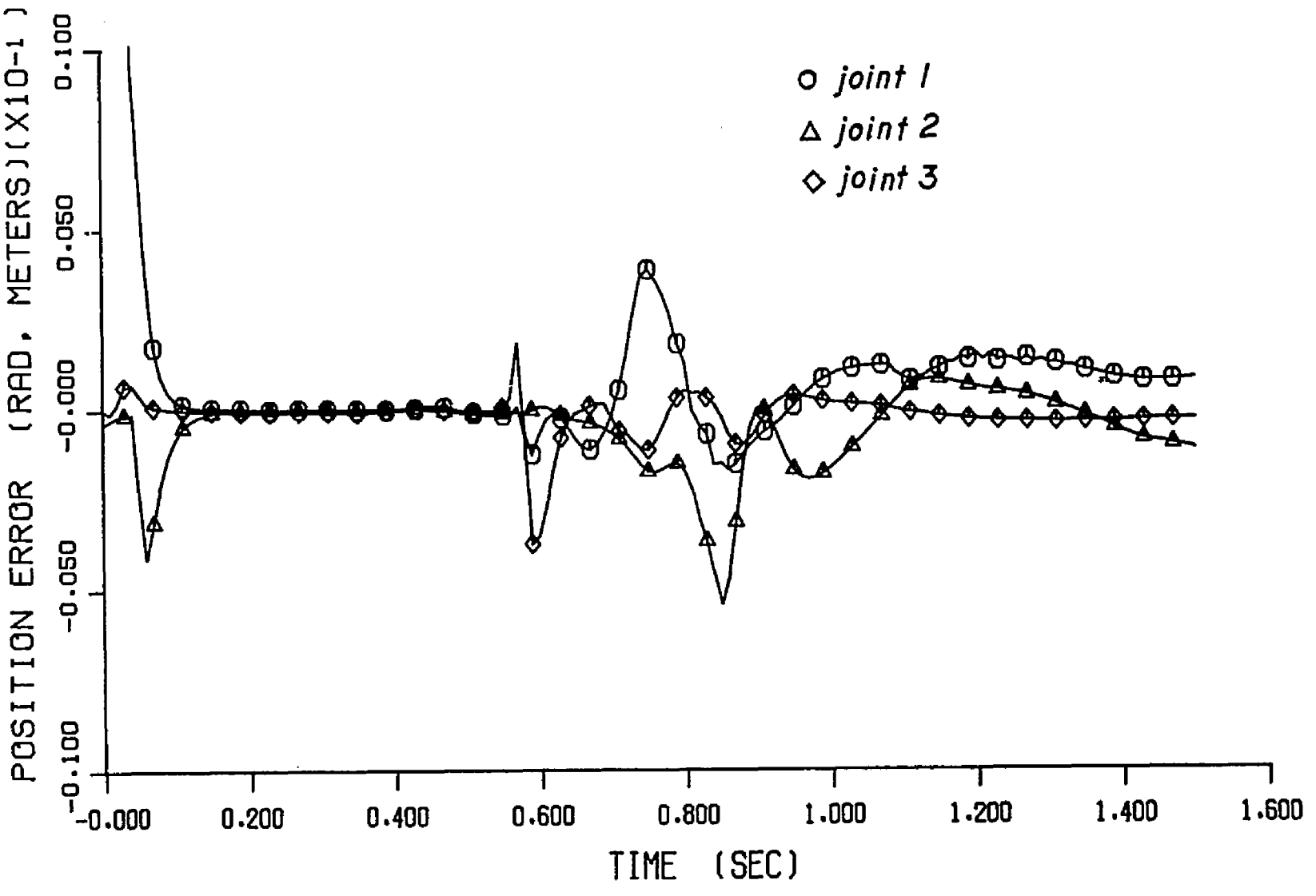

Figure 5.4. Deviations from Straight Line Trajectory Using Control 1.

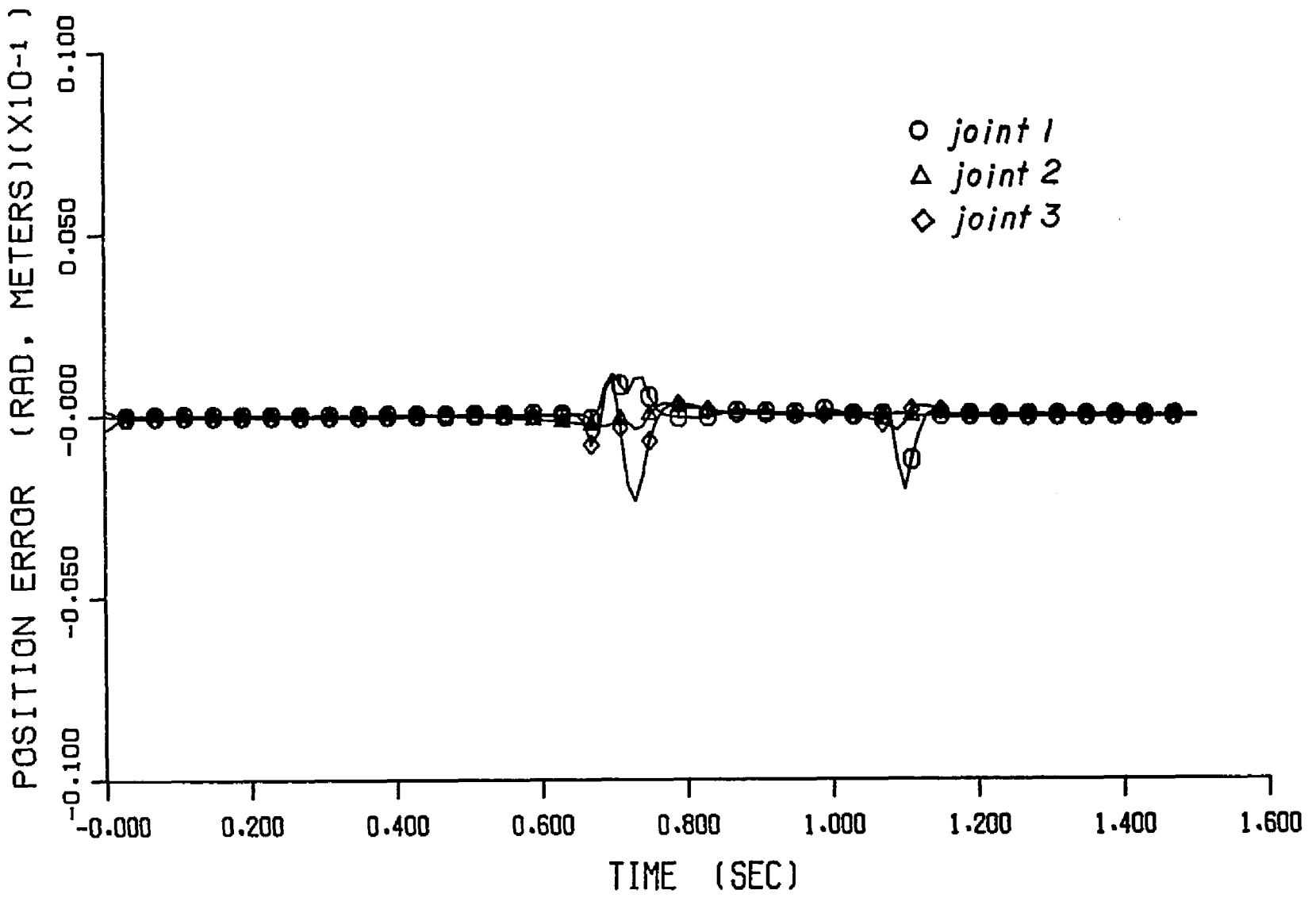

Figure 5.5. Deviations from Straight Line Trajectory Using Control 2.

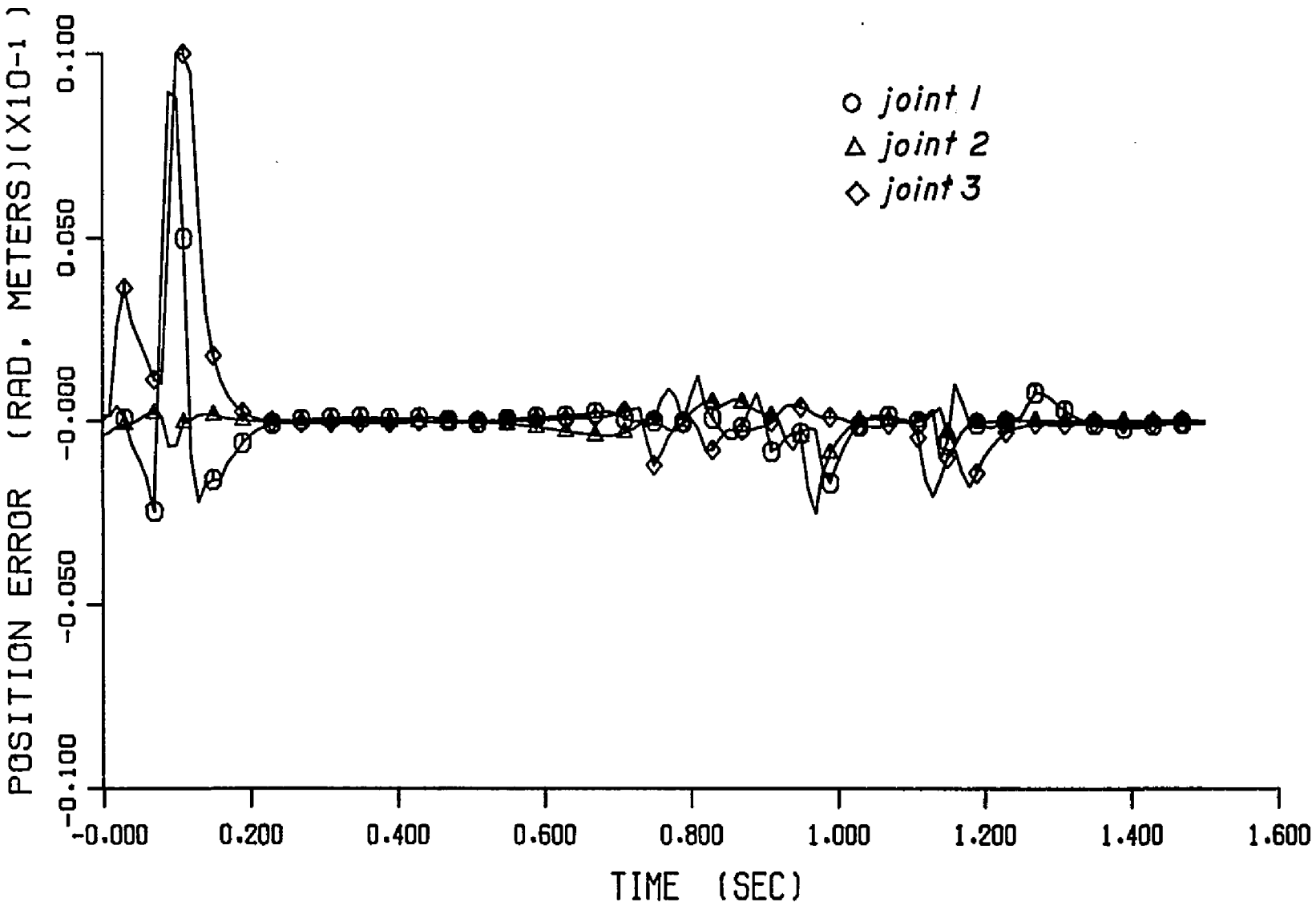

Figure 5.6. Deviations from Straight Line Trajectory Using Control 3.

The effective inertia histories of joint 1 (d_{11}), joint 2 (d_{22}), and joint 3 (d_{33}) are shown in Figure (5.7). Using control 2, since it is most effective in straight line motion, yields the position errors of Figure (5.8). Note that although link 2 initially droops due to the instantaneous increase in gravity load, it recovers very quickly. It also appears that the control is adapting to the new inertia parameters when the return motion begins at t = 0.85 seconds.

5.4 Comparison of Controls 1, 2, and 3

In order to analyze the effectiveness of each model to compensate for dynamic interactions between the joints, trajectories are selected which emphasize one phenomenon at a time. Compensation of coupling inertia torques and tracking of rapidly varying effective inertias are the two cases studied here.

5.4.1 Coupling Inertia Compensation

Referring to equation (2.1.1), consider the dynamics of joint i,

$$d_{ii}\ddot{q}_i + \sum_{\substack{j=1 \\ i\neq j}}^{N} d_{ij}\ddot{q}_j + \ldots = \tau_i \qquad (5.4.1)$$

which may also be expressed as,

$$d_{ii}(\ddot{q}_i + \sum_{\substack{j=1 \\ i\neq j}}^{N} (d_{ij}/d_{ii})\ddot{q}_j) + \ldots = \tau_i \qquad (5.4.2)$$

Thus, large d_{ij}/d_{ii} ratios and rapidly varying $\ddot{q}_j$ place a large demand on the adaptive feedforward component of the control.

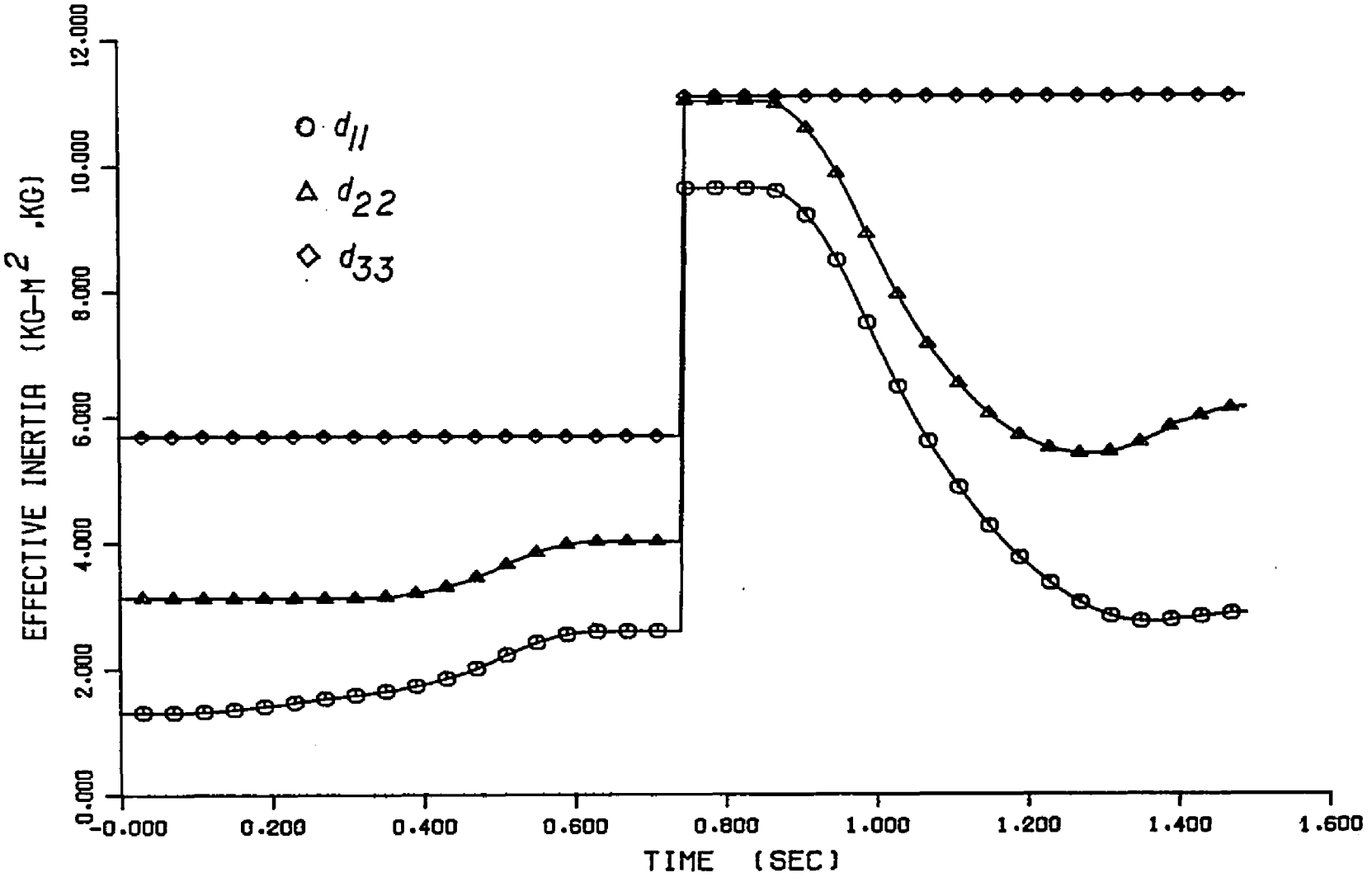

Figure 5.7. Effective Inertia Profiles During Payload Manipulation.

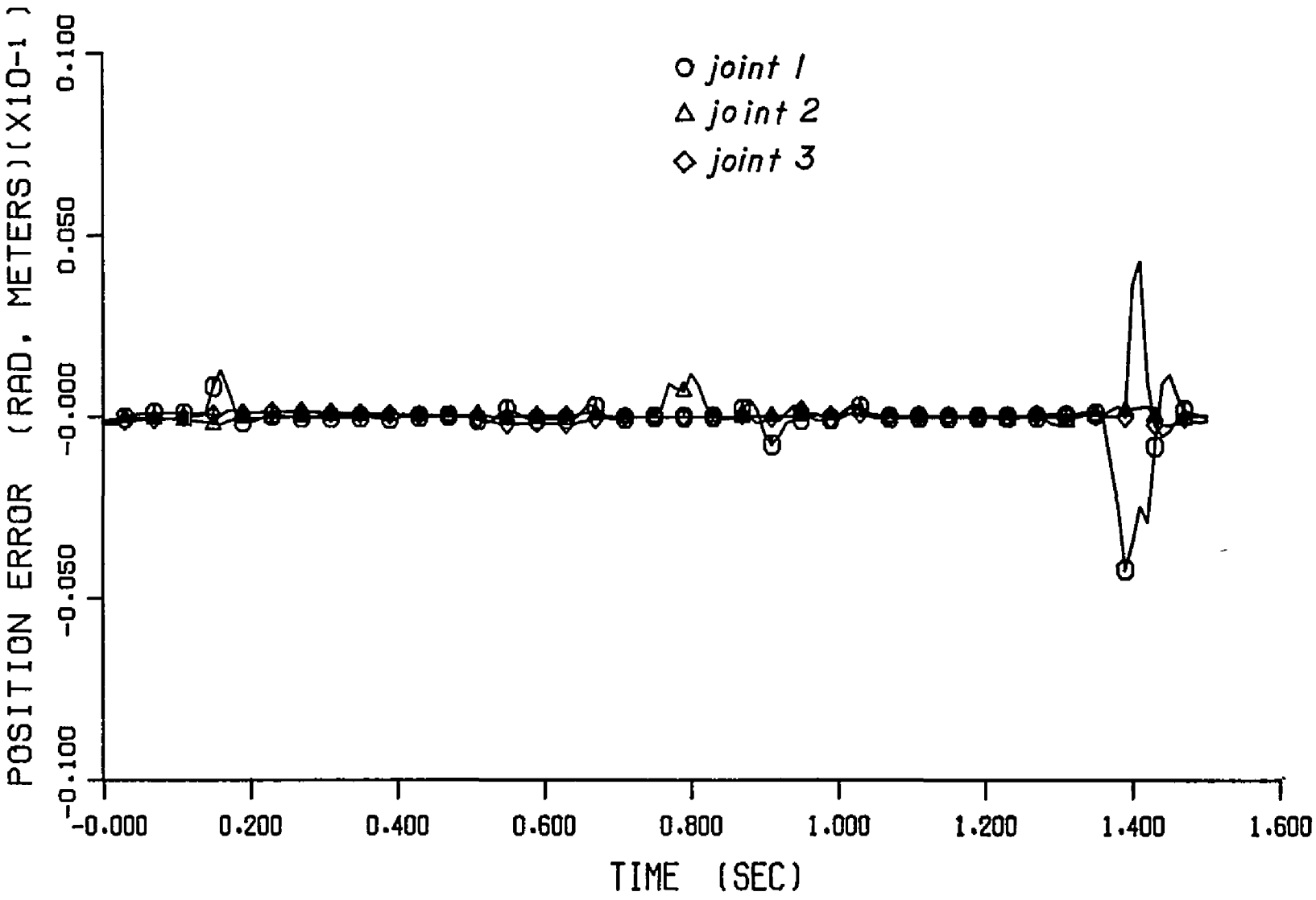

Figure 5.8. Position Error Profiles During Payload Manipulation.

Such is the case for the 0.6 second trajectories of Figure (5.9) where the regulation of joint 3 in the presence of a significant coupling inertia disturbance from joint 1 is examined. For $0.0 < t < 0.2$ seconds a small learning signal is applied to the system to enable the adaptation algorithm to drive the parameter vector estimate, $\hat{\theta}$, from an initial guess, $\hat{\theta}(0)$, to within a small neighborhood of the true parameter vector, θ. The actual trajectory of interest is during the period $0.2 \leq t \leq 0.6$. The joint 1 trajectory is specifically designed to create large coupling inertia torques ($= d_{31}\ddot{q}_1$) on joint 3.

The acceleration profile of joint 1,

$$\ddot{q}_1 = 39.3 \text{ rad/s}^2 \quad \text{for} \quad 0.2 \leq t < 0.4$$

$$\ddot{q}_1 = -39.3 \text{ rad/s}^2 \quad \text{for} \quad 0.4 \leq t \leq 0.6$$

can be expected to disturb joint 3 tracking, particularly during the discontinuity of $\ddot{q}_1$ at $t = 0.4$ seconds. During the entire trajectory, $d_{31} = -0.428$ kg-m. Since $q_2 = 90°$, there are no gravity forces acting on joint 3. The coupling inertia torque will therefore be the dominant factor of the joint 3 dynamics since $\ddot{q}_{r_2} = \ddot{q}_{r_3} = 0.0$.

Joint 3 position error results for control 1 are shown in Figure (5.10). Two different simulation runs using control 1 are shown. In run A, the initial parameter estimate for joint 3, $\hat{\theta}_3(0)$ is only slightly perturbed from the actual parameter vector, $\theta_3(0)$. In run B, $\hat{\theta}_3(0)$ is skewed significantly from $\theta_3(0)$. Although disturbance rejection is acceptable at $t = 0.4$, the overall performance is poor due to the slow parameter convergence during the period $0 < t < 0.3$ seconds. Position

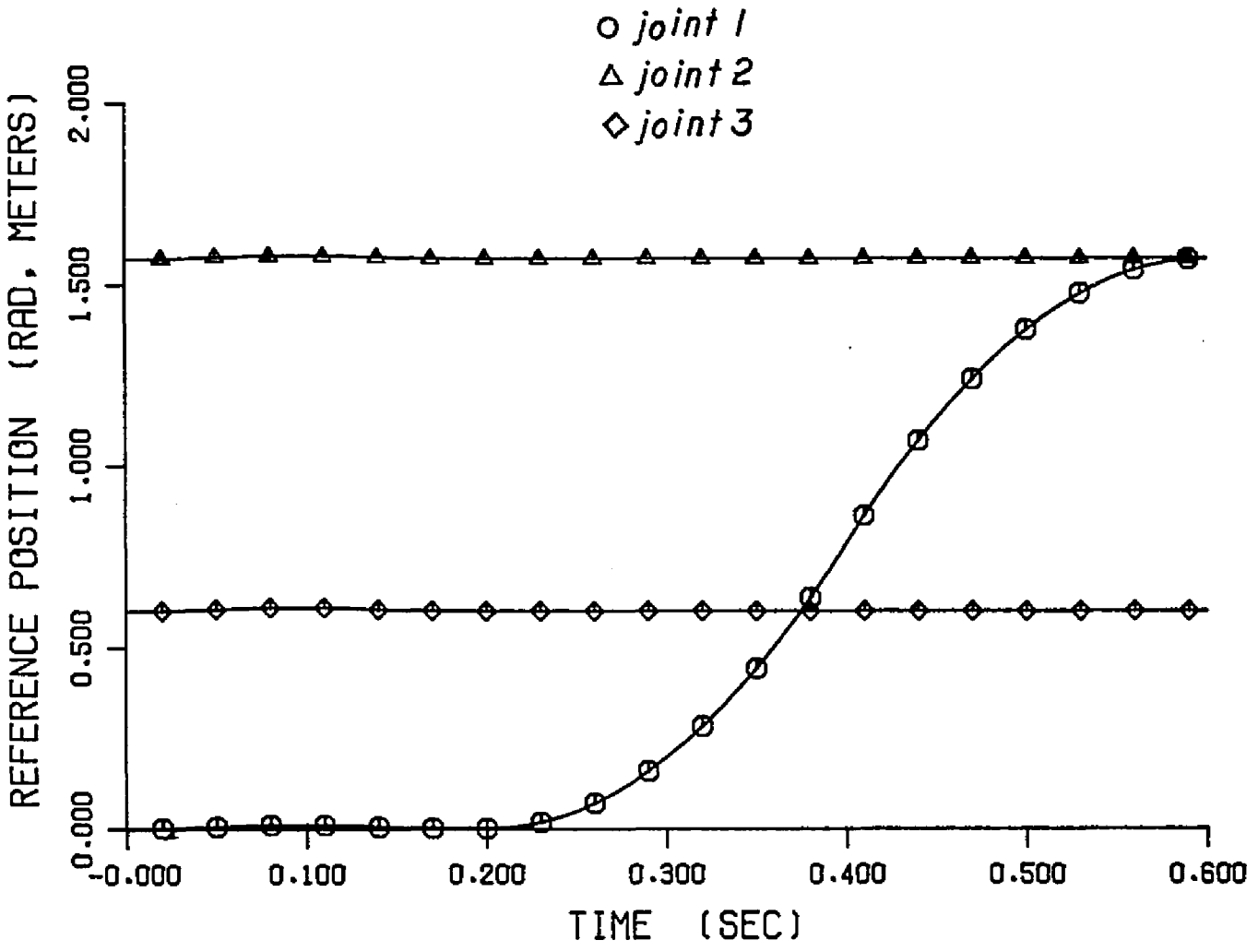

Figure 5.9. Reference Trajectories, Test of Coupling Inertia Disturbance Rejection.

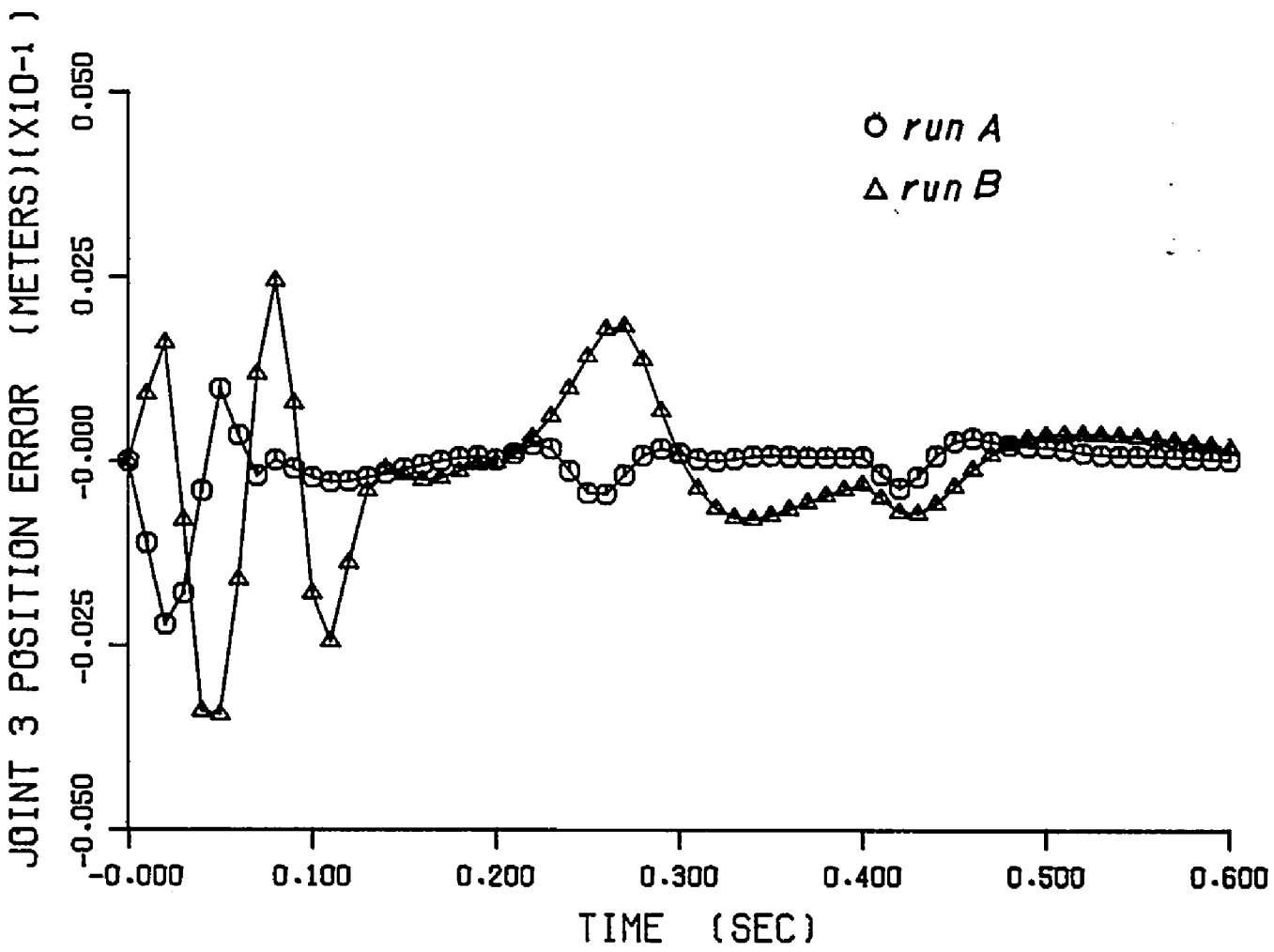

Figure 5.10. Joint 3 Position Errors Using Control 1, Effect of $\hat{\theta}(0)$.

error results for controls 1 (run A), 2 and 3 are shown in Figure (5.11). Control 2 deviates sharply from the desired trajectory due to the disturbance at t = 0.4. Note, however, the longer learning times of the more complex controllers (1 and 3) compared to control 2. These results are consistent with other similar simulation runs.

5.4.2 Effective Inertia Tracking

Another criterion of interest is the ability to compensate for rapidly varying effective inertia, d_{ii}. The effective inertia of joint 1, d_{11}, will vary from 1.8 to 2.7 kg-m^2 as joint 2 is swept from 45° to 90°, with joint 3 fully extended. This situation is simulated in a 1.2 second motion. A learning signal is applied during the first 0.2 seconds of the trajectory. This enables the adaptation algorithm to converge to the proper parameters before the trajectory of interest starts at t = 0.2 seconds. It also gives an indication of the relative convergence rates of the three controllers. The joint of interest is joint 1 as it is swept from 0° to 60° during the period $0.2 \leq t \leq 1.2$ seconds. The effective inertia of joint 1 during this motion is illustrated in Figure (5.12).

The joint 1 position errors for controllers 1, 2, and 3 are shown in Figure (5.13). Note that controls 2 and 3 give superior performance over control 1. Also note that consistent with Figure (5.11), the learning time of control 1 is longer than that of control 2.

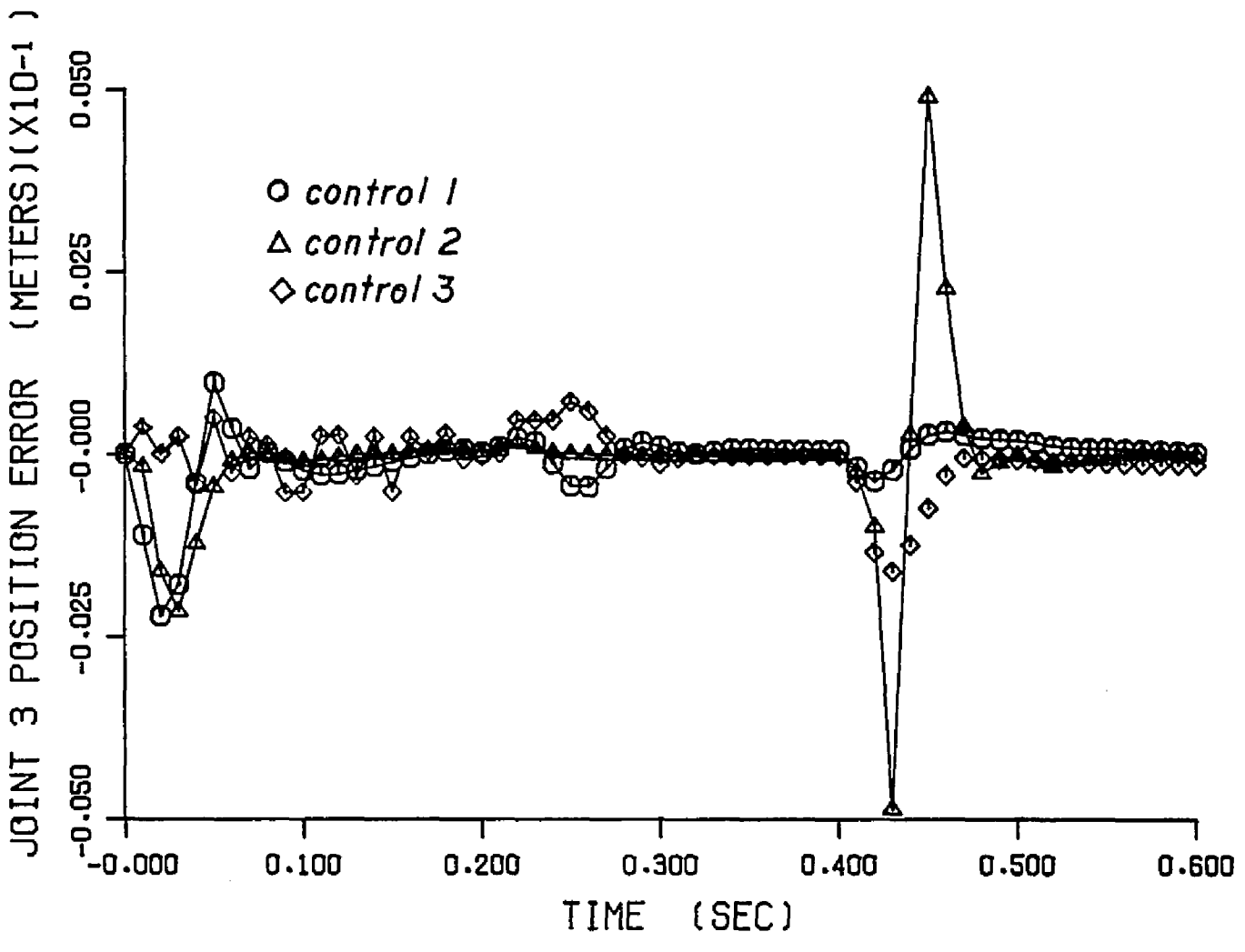

Figure 5.11. Comparison of Position Errors Using Controls 1, 2, and 3.

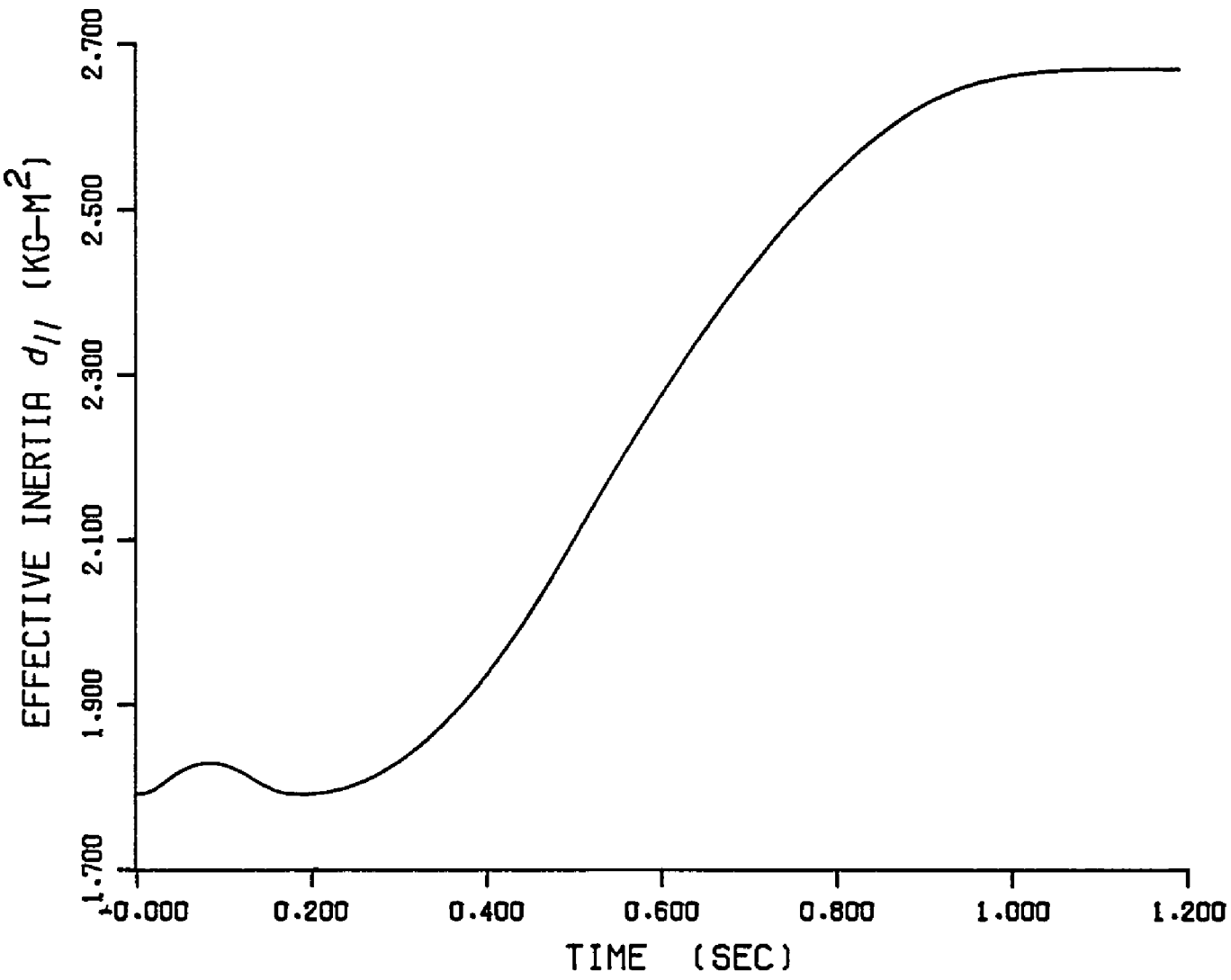

Figure 5.12. Effective Inertia Profile of Joint 1.

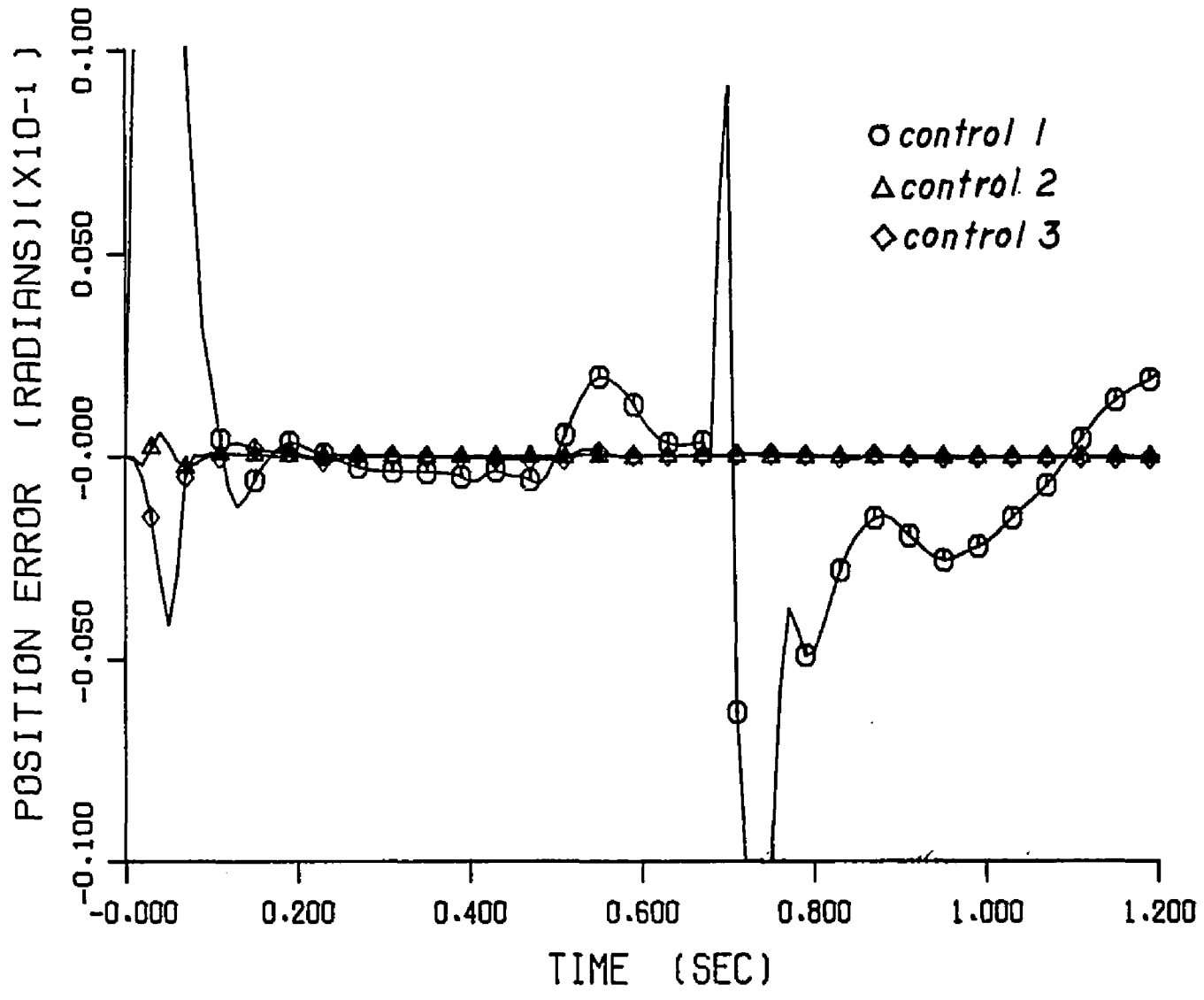

Figure 5.13. Effective Inertia Tracking, Comparison of Controls 1, 2, and 3.

5.5 Interpretation of Results

The simulation results show that selection of the model upon which the control algorithm is based has a dramatic effect on the closed loop performance of the system. The results also show that no one model is superior for all motions. A quick review of models 1, 2, and 3 (equations (4.2.8), (4.2.10), and (4.2.11), respectively) will shed some light on the performance trade-off. Model 1, although offering the most complex and accurate representation of the coupling disturbances, requires the control to identify 11 parameters and consequently requires a longer learning time. Model 2 is the simplest model, lumping all coupling disturbances into one term, h_i. It requires the control to identify only 3 parameters, however. Model 3 is a compromise between the complexity of model 1 and the simplicity of model 2. Lumping only the gravity terms into the parameter h_i, it requires identification of 7 parameters.

The trade-off is between the convergence rate of the parameter adaptation algorithm (PAA) and the accuracy in representing the disturbance torques. As discussed in Chapter 3, the success of the control system is dependent upon the ability of the PAA to track time-varying system parameters. Simulation results indicate that the convergence rates are slower when there are more parameters to estimate. This can be seen by comparing the initial learning times of the three models in Figures (5.11) and (5.13). Thus, for motions in which there are rapid variations in the dynamic coefficients, model 2 is more robust and yields superior performance. This is the case for the motions simulated in Sections 5.2 and 5.4.2.

The very feature that makes model 2 so simple and thus converge so quickly, namely lumping all the disturbance torques into one parameter, can be a liability in some situations. The problem will manifest itself during periods of dominant coupling with discontinuous or rapidly varying joint accelerations.

The failure of model 2 to perform well in such cases (e.g., Section 5.4.1, where models 1 and 3 yield superior disturbance rejection) is explained as follows. Consider the task of controlling joint 1 in the presence of a significant coupling inertia disturbance from joint 2. In the model 2 representation, this disturbance is lumped into h_1, i.e.,

$$h_1 = d_{12}\ddot{y}_2 \tag{5.5.1}$$

If the trajectory of joint 2 includes a discontinuity in acceleration, as in Figure (5.14), then the parameter, h_1, will also be discontinuous. During the time that it takes the PAA to converge to this new value of h_1, performance will be degraded. Thus, lumping disturbances into the term can sacrifice performance.

In summary, model 2 yields the fastest convergence, and is by far the most robust algorithm. It excels in high speed tracking of trajectories with smooth acceleration profiles. Model 1 is less robust than models 2 and 3, but is superior in compensating for coupling inertia disturbances with rapidly varying or discontinuous joint accelerations (such trajectories will result from minimum time, bang-bang control). Of course, the algorithm resulting from model 1 requires many more floating point operations per sampling period, and thus any performance

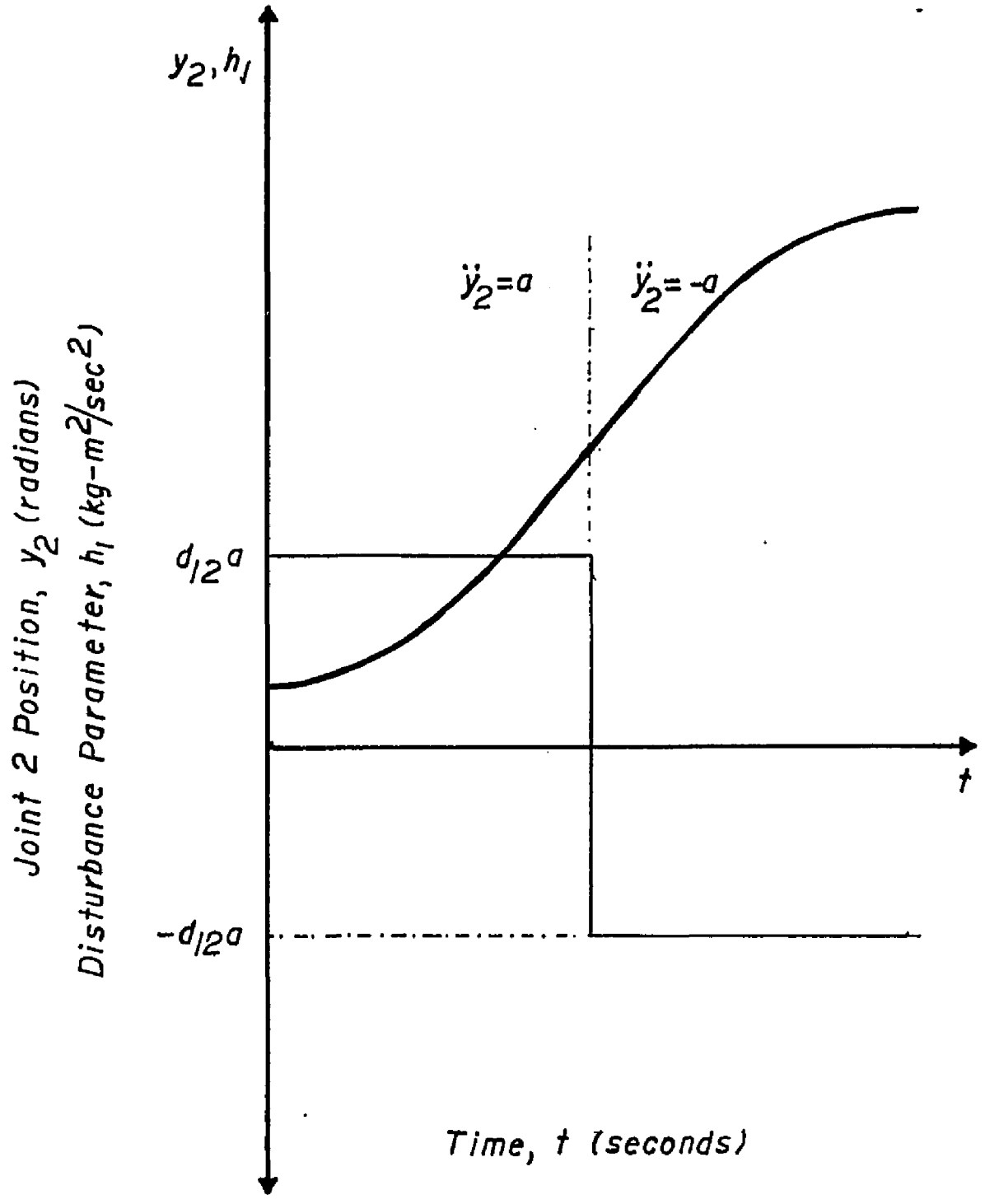

Figure 5.14. The Parameter h_1 (Model 1) for a Minimum Time Trajectory.

gains would have to be weighed against implementation cost. The performance of model 3 is somewhere in between these two extremes, giving good performance for a wide variety of trajectories.

CHAPTER 6

CONCLUSIONS

The major contributions of this work may be summarized as follows.

1. The successful application of adaptive control to demanding manipulator tracking problems has been demonstrated. These high speed motion simulations have involved large joint displacements, large variations in the joint dynamic coefficients, and significant dynamic coupling between the joints. The adaptive control is used here as a stand-alone, full-authority control. That is to say, it is not used as a perturbation control in conjunction with a nominal control.

2. The nature of the independent joint control objectives and the manipulator dynamics promote a decentralized control scheme. There is an autonomous controller dedicated to each joint. In general, the controllers do not ignore the states of the other joints, as this information is necessary to take advantage of feedforward compensation. The resulting decentralized SISO controllers have distinct implementation advantages over a single centralized controller in terms of algorithm complexity.

3. The effect of subsystem model structure on the quality of feedforward compensation has been investigated. No one controller yields superior performance for all trajectories, as there is a trade-off between adaptation rate and accuracy in representing the coupling torques.

Each model's strengths and weaknesses in tracking certain kinds of trajectories have been discussed.

How does this adaptive scheme compare with the most popular precision tracking controller in terms of performance and complexity? The computed torque technique (described in Chapter 1), although undoubtedly a very robust algorithm, is susceptible to tracking errors unless all system parameters are known very accurately. Inertia parameters of the links, actuator parameters, friction coefficients, and even the mass and inertia properties of the objects to be manipulated must be known a priori. The adaptive control with its on-line learning capability is more flexible in this respect.

Unlike the adaptive scheme, the computed torque technique is a centralized controller which does not lend itself to distributed processing. The computed torque controller hardware consists of one processing element, while the proposed adaptive controller hareware consists of N processing elements (for an N jointed manipulator) operating in parallel. The adaptive control scheme based on model 3 requires roughly the same number of operations per processing element as the computed torque control. The adaptive control algorithm based on model 1 requires many more, while the algorithm based on model 2 requires fewer operations per processing element than the computed torque control.

The decentralized adaptive control scheme developed in this thesis has compared favorably with other adaptive control schemes applied to the manipulator problem. Its strengths include: (1) efficiency, as the resulting algorithm shows potential for easy implementation by

distributed processing techniques; (2) insensitivity to motor parameter drift as the motor dynamics are included within the adaptation loop; (3) performance, as the benefits of proper modelling and compensation of the dynamic coupling between joints have been demonstrated. None of the earlier developed adaptive schemes may claim all of these features.

Although this work has answered many questions on the applicability of adaptive techniques to the decentralized control of manipulator dynamics, many issues remain open to further investigation. Theoretical studies which might confirm generalizations drawn from the simulation results include relating convergence rate of the PAA to tracking performance and robustness. The simulation could be upgraded to determine the effects of computational delay, nonlinear joint friction, and actuator saturation on closed loop performance. The control schemes could be applied to a manipulator with lower gear ratios, in which case the effects of coupling torques would be more dramatic and the quality of feedforward compensation more critical. Despite the theoretical and practical questions that remain, adaptive control has been shown to be a promising solution to the control problem.

APPENDIX A

DISCRETIZATION OF SUBSYSTEM DYNAMICS

From equation (4.2.2), the dynamics of subsystem i are given by

$$J_i\ddot{y}_i(t) + [B_{v_i} + (K_{t_i}/R_{m_i})K_{B_i}]\dot{y}_i =$$

$$[K_{t_i}/(n_iR_{m_i})]u_i(t) - (1/n_i^2)P_i \qquad \text{(A-1)}$$

where

$$J_i = J_{m_i} + (1/n_i^2)d_{ii}(y) \qquad \text{(A.2)}$$

$$P_i = \sum_{\substack{j=1 \\ j\neq i}}^{N} d_{ij}(y)\ddot{y}_j + f_i(y,\dot{y}) + g_i(y) \qquad \text{(A.3)}$$

For the sake of brevity, the i subscripts will be excluded from the discussion except where necessary.

As a crude linearization of equations (A.1) through (A.3), the nonlinear terms $d_{ii}(y)$, $d_{ij}(y)$, $f(y,\dot{y})$, and $g(y)$ are assumed constant, i.e.,

$$d_{ii}(y) = d_{ii} \qquad \text{(A.4)}$$

$$d_{ij}(y) = d_{ij} \qquad \text{(A.5)}$$

$$f(y,\dot{y}) = f \qquad \text{(A.6)}$$

$$g(y) = g \qquad \text{(A.7)}$$

First consider the undisturbed subsystem,

$$J_i\ddot{y}_i(t) + [B_v + (K_t/R_m)K_B]\dot{y}_i(t) = [K_t/(nR_m)]u_i(t) \qquad \text{(A.8)}$$

which may be rewritten as,

$$J_i\frac{d}{dt}(\dot{y}_i(t)) + [B_v + (K_t/R_m)K_B]\dot{y}_i(t) = [K_t/(nR_m)]u_i(t) \qquad \text{(A.9)}$$

Taking the Laplace transform of both sides of equation (A.9), the continuous time transfer function is

$$G_i(s) = \frac{\dot{Y}_i(s)}{U_i(s)} = \frac{K_t/(nJ_iR_m)}{s + (1/J_i)[B_v + (K_t/R_m)K_B]} \tag{A.10}$$

$$= \frac{\alpha}{s + \beta} \tag{A.11}$$

where

$$\alpha = K_t/(nJ_iR_m) \tag{A.12}$$

$$\beta = (1/J)[B_v + (K_t/R_m)K_B] \tag{A.13}$$

The discrete time transfer function, $G_i(z)$, is related to $G_i(s)$ by the following expression (Kuo 1980),

$$G_i(a) = (1 - z^{-1})Z[G_i(s)/s] \tag{A.14}$$

where $Z[G_i(s)/s]$ denotes the z transform of

$$\{\mathcal{L}^{-1}[G_i(s)/s]\}\Big|_{t=kT} \tag{A.15}$$

Thus,

$$G_i(z) = (1 - z^{-1})Z[\frac{\alpha}{s(s+\beta)}] \tag{A.16}$$

$$= (1 - z^{-1})][\frac{\alpha}{\beta}(\frac{z}{z-1} - \frac{z}{z - e^{-\beta T}})] \tag{A.17}$$

Combining the terms, the discrete time transfer function is found to be,

$$G_i(z) = \frac{\dot{Y}_i(z)}{U_i(z)} = \frac{\alpha}{\beta}\ [\frac{1 - e^{-\beta T}}{z - e^{-\beta T}}] \tag{A.18}$$

Taking the inverse Z transform, the difference equation is found to be,

$$\dot{y}_i(k+1) = e^{-\beta T}\dot{y}_i(i) + \frac{\alpha}{\beta}(1 - e^{-\beta T})u_i(k) \tag{A.19}$$

or

$$\dot{y}_i(k+1) = -a_1\dot{y}_i(k) + b_0 u_i(k) \tag{A.20}$$

where,

$$a_1 = -e^{-\beta T} \tag{A.21}$$

$$b_o = (\alpha/\beta)(1 - e^{-\beta T}) \tag{A.22}$$

Now consider the subsystem dynamics under the influence of a coupling inertia disturbance, i.e.,

$$J_i\ddot{y}_i(t) + [B_v + (K_t/R_m)K_B]\dot{y}_i(t) = -(1/n^2)d_{ij}\ddot{y}_j(t) \tag{A.23}$$

The discrete time difference equation relating $\dot{y}_i(k)$ to $\dot{y}_j(k)$ is found by applying equation (A.14):

$$\dot{y}_i(k+1) = e^{-\beta T}\dot{y}_i(k) - [1/(J_i n^2)]d_{ij}[\dot{y}_j(k+1) - \dot{y}_j(k)] \tag{A.24}$$

where β is defined by equation (A.13). Thus the coupling inertia disturbances can be represented by a moving average of the other joint velocities.

By similar reasoning, it can be shown that the gravity disturbance, $g_i(y)$, can be linearized and represented by a moving average of the joint positions. In fact, both the coupling inertia and gravity disturbances can be represented by a moving average of the joint positions if three terms are used in the series, i.e.,

$$\dot{y}_i(k+1) = e^{-\beta T}\dot{y}_i(k) + c_0 y_j(k) + c_1 y_j(k-1) + c_2 y_j(k-2) \tag{A.25}$$

By superposition then, the complete linear, discretized representation of the subsystem dynamics is given by,

$$\dot{y}_i(k+1) = -a_1\dot{y}_i(i) + b_0u_i(k) + \sum_{j=1}^{N} c_j(z^{-1})v_j(k) \qquad (A.26)$$

where $v_j(k) = y_j(k)$ or $v_j(k) = \dot{y}_j(k)$.

APPENDIX B

DYNAMIC PARAMETERS OF THE STANFORD MANIPULATOR

Mechanical Parameters	Link 1	Link 2	Link 3
Link Inertias (kg-m^2)			
I_{xx}	0.276	0.108	2.51
I_{yy}	0.255	0.018	2.51
I_{zz}	0.071	0.100	0.006
Link Center of Mass (m)			
$\bar{x}$	0.0	0.0	0.0
$\bar{y}$	0.0175	-0.1054	0.0
$\bar{z}$	-0.1105	0.0	-0.64
Link Mass (kg)			
m	9.29	5.01	4.25
Motor Inertia (kg-m^2)			
J_m	$5.65 \cdot 10^{-5}$	$2.33 \cdot 10^{-4}$	$2.05 \cdot 10^{-4}$
Dynamic Friction (N-m-sec/rad)			
B_v	$8.09 \cdot 10^{-5}$	$3.03 \cdot 10^{-4}$	$3.11 \cdot 10^{-2}$
Gear Ratio			
n	100:1	100:1	82.9:1.0 (rad/m)

Note: Link Inertias and Link Center of Masses are referred to the link coordinate systems of Figure 5.1.

Electrical Parameters	Link 1	Link 2	Link 3
Torque Constant (N-m/amp)			
K_t	$4.3 \cdot 10^{-2}$	$1.02 \cdot 10^{-1}$	$2.83 \cdot 10^{-1}$
EMF Constant (volt-sec/rad)			
K_B	$4.2 \cdot 10^{-2}$	$1.01 \cdot 10^{-1}$	$3.03 \cdot 10^{-1}$

LIST OF REFERENCES

Asada, H., Kanade, T., and Takeyama, I. "Control of a Direct-Drive Arm," Proc. of the Winter Annual Meeting of the ASME, pp. 63-72, Phoenix, AZ, Nov. 1982.

Astrom, K., Borrison, U., Ljung, L., and Wittenmark, B. "Theory and Applications of Self-Tuning Regulators," Automatica, Vol. 13, pp. 457-476, 1977.

Astrom, K. J., and Eykhoff, P. "System Identification--A Survey," Automatica, Vol. 7, pp. 123-162, 1971.

Bejczy, A. K. "Robot Arm Dynamics and Control," Tech. Memo 33-669, Jet Propulsion Laboratory, Feb. 1974.

Borrison, U. "Self-Tuning Regulators for a Class of Multivariable Systems," Automatica, Vol. 15, pp. 209-215, 1979.

Clarke, D. W. "Introduction to Self-Tuning Controllers," Self-Tuning and Adaptive Control: Theory and Applications, C. J. Harris and S. A. Billings, Eds. New York: Peregrinus, pp. 36-71, 1981.

Horowitz, R., and Tomizuka, M. "An Adaptive Control Scheme for Mechanical Manipulators--Compensation of Nonlinearity and Decoupling Control," Proc. of the Winter Annual Meeting of the ASME, Paper 80-WA/DSC-6, Chicago, IL, 1980.

Horowitz, R., and Tomizuka, M. "Model Reference Adaptive Control of Mechanical Manipulators," Proc. of the IFAC Workshop on Adaptive Systems in Control and Signal Processing, San Francisco, CA, June 1983.

Isermann, R. "Parameter Adaptive Control Algorithms--A Tutorial," Proc. of the 6th IFAC/IFIP Conf., pp. 139-154, Duesseldorf, Germany, Oct. 1982.

Koivo, K. J., and Guo, T. "Adaptive Linear Controller for Robotic Manipulators," IEEE Trans. Automatic Control, Vol. AC-28, pp. 162-171, Feb. 1983.

Kuo, B. C. Digital Control Systems. New York: Holt, Rinehart, and Winston, 1980.

Landau, I. D. "Model Reference Adaptive Control and Stochastic Self-Tuning Regulators--A Unified Approach," Trans. ASME, J. Dynamic Systems, Measurement, and Control, Vol. 103, No. 4, pp. 404-416, Dec. 1981.

Lee, C. S. G., and Chung, M. J. "An Adaptive Control Strategy for Computer-Based Manipulators," Proc. of 21st IEEE Conf. Decision and Control, pp. 95-100, Dec. 1982.

Lozano, R. and Landau, I. D. "Redesign of Explicit and Implicit Discrete Time Model Reference Adaptive Control Schemes," Int. J. Control, Vol. 33, No. 2, pp. 247-268, 1981.

Luh, J. Y. S., Walker, M. W., and Paul, R. P. C. "On-Line Computational Scheme for Mechanical Manipulators," Trans. ASME, J. Dynamic Systems, Measurement, and Control, Vol. 102, pp. 69-76, June 1980.

Markiewicz, B. R. "Analysis of the Computed Torque Drive Method and Comparison with Conventional Position Servo for a Computer-Controlled Manipulator," Tech. Memo 33-601, Jet Propulsion Laboratory, March 1973.

Paul, R. P. Robot Manipulators: Mathematics, Programming and Control. Cambridge, MA: MIT Press, 1981.

Vukobratovic, K. M. and Stokic, D. M. "One Engineering Concept of Dynamic Control of Manipulators," Trans. ASME, J. Dynamic Systems, Measurement, and Control, Vol. 104, pp. 108-118, 1981.

Walker, M. W. and Orin, D. E. "Efficient Dynamic Computer Simulation of Robotic Mechanisms," Trans. ASME, J. Dynamic Systems, Measurement, and Control, Vol. 104, pp. 205-211, Sept. 1982.