RDS-TR-10-82

AN ADAPTIVE CONTROL STRATEGY FOR COMPUTER-BASED MANIPULATORS[1]

M. J. Chung
C. S. G. Lee

August 1982

CENTER FOR ROBOTICS AND INTEGRATED MANUFACTURING

Robot Systems Division

COLLEGE OF ENGINEERING

THE UNIVERSITY OF MICHIGAN

ANN ARBOR, MICHIGAN 48109

---

[1] This work was supported in part by the National Science Foundation Grant ECS-8106954 and the Robot Systems Division of the Center for Robotics and Integrated Manufacturing (CRIM) at The University of Michigan, Ann Arbor, MI. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the funding agencies.

Geo
G
C 78
1300

TABLE OF CONTENTS

Abstract

This report focuses on the study of an adaptive control method based on the perturbation equations in the vicinity of a desired trajectory. The highly coupled non-linear dynamic equations of a manipulator are expanded in the vicinity of a pre-planned joint trajectory to obtain the perturbation equations. These perturbation equations are then used to design a feedback control law about the desired trajectory. The torques for the joint actuators consist of nominal torques computed from the Newton-Euler equations of motion and the variational torques computed from the perturbation equations. Since the parameters in the perturbation equations are unknown and also slowly time-varying, a recursive least square identification scheme is used to perform on-line parameter identification. The parameters of the perturbation equations and the feedback gains of the controller are updated and adjusted in each sampling period successively to obtain the necessary control effort. This adaptive control strategy reduces the manipulator control problem from a nonlinear control to controlling a linear control system about a desired trajectory. Furthermore, a clear advantage of such formulation is that the nominal and variational torques can be computed separately and simultaneously. Computer simulation studies of a three-jointed PUMA robot arm are performed on a VAX-11/780 computer to illustrate the performance of this adaptive control strategy.

1. Introduction

Most automated manufacturing tasks are done by special purpose machines which are designed to perform their prespecified functions in a manufacturing process. The inflexibility of these machines makes the computer-controlled manipulators more attractive and cost effective in various manufacturing and assembly tasks. With the advancement of computer technology, a new generation of computer-based robots is emerging. This report deals with the control problem of these computer-based manipulators.

Given the equations of motion of a manipulator, the control problem is to find appropriate torques/forces to servo all the joints of the manipulator in real-time to track a desired trajectory as closely as possible. Several control methods are available in accomplishing this task. Most notable of these are in [1-15]. Current industrial practice employs conventional servomechanism to control present day manipulators. An n-jointed manipulator is being modeled as "n" separate joint manipulators. Each joint subsystem is controlled independently by simple servomechanism technique. However, the motion dynamics of an "n" degree-of-freedom manipulator is inherently nonlinear and can only be described by a set of "n" highly coupled nonlinear second order ordinary differential equations. Furthermore, the relationship between the workspace coordinates and the joint coordinates is given by complex trigonometric transformations. Hence the servomechanism approach models the varying dynamics of a manipulator inadequately and neglects the coupling effects of the joints. As a result, these manipulators move at slow speeds with unnecessary vibrations.

To maintain good performance over a wide range of motions and payloads, adaptive control methods may prove suitable. Among various adaptive methods, Model Referenced Adaptive Control (MRAC) is the most widely used and relatively easy to implement. In the MRAC method proposed by [10], a linear second-order time invariant differential equation was used as the reference model for each degree of freedom. The manipulator is controlled by adjusting position and velocity feedback gains to follow the reference model. To adjust position and velocity gains, the steepest descent method was used as the adaptive algorithm. In this case it is not easy to design a stable adaptive control law. Consequently stability analysis is critical. Unfortunately this stability analysis is very difficult because of the non-linearities and complexity of the dynamic equations of a manipulator.

In order to extend the capabilities of manipulators and improve their overall dynamic performance, there is a need to investigate and develop better adaptive control strategies that provide better control solutions to current control methods. The main goal of this report is to present the design and development of an adaptive control strategy which yields high performance over wide range of manipulator motions and payloads.

In the following sections, vectors are in boldface lower case alphabets while matrices are in boldface upper case alphabets.

2. Adaptive Control Formulation

In this section, we discuss the dynamic models of a manipulator that are useful for the purpose of control and briefly present the proposed adaptive control method. Major issues of the approach are discussed.

2.1. Dynamic Models of Manipulators

A priori information needed for control is a set of differential equations describing the dynamic behavior of a manipulator. Two main approaches are used by most researchers to systematically derive the dynamic model of a manipulator - the Lagrange-Euler (L-E) and the Newton-Euler (N-E) formulations. Bejczy [4] based on the Lagrangian formulation has shown that the dynamic equations of motion for a six-jointed manipulator (Stanford arm) are highly nonlinear and consists of inertia loading, coupling reaction forces between joints and gravity loading effects. However, the dynamic equations of motion as formulated by the Lagrange-Euler method have been shown to be computationally inefficient, and real-time control based on the "complete" dynamic model has been found difficult to achieve if not impossible [3,4,15]. In general, the Lagrange-Euler equations of motion for an n-jointed manipulator can be expressed in matrix vector notation as:

$$D(\vartheta)\,\ddot{\vartheta} + H(\vartheta,\dot{\vartheta}) + G(\vartheta) = \tau \qquad (1)$$

where τ is an $n \times 1$ external applied torques for joint actuators, ϑ is the joint angles, $\dot{\vartheta}$ is the joint velocities, $\ddot{\vartheta}$ is an $n \times 1$ joint acceleration vector, $G(\vartheta)$ is an $n \times 1$ gravitational force vector, $H(\vartheta,\dot{\vartheta})$ is an $n \times 1$ Coriolis and Centrifugal force vector and $D(\vartheta)$ is an $n \times n$ acceleration-related matrix.

The joint torques as computed from Eq. 1 are of order $O(n^3)$. For a six-jointed PUMA manipulator, it involves about 102,740 multiplications and 78,479 additions to compute the joint torques per trajectory set point [15].

To improve the speed of computation, simplified sets of equations have been used by other investigators. In general, these models simplify the underlying physics by neglecting the Coriolis and Centrifugal force terms. The resulting controlled manipulator has suboptimal dynamic performance restricting arm movement to slow speeds. At high speeds, the neglected terms become significant, making the accurate position control of the arm more difficult. For this reason, in most cases the performance specifications of computer-controlled manipulators have been relatively low so that relatively simple control methods are adequate.

An approach which has the advantage of both speed and accuracy was based on the N-E formulation [16]. This formulation yields a set of forward and backward recursive equations which can be applied to the robot links sequentially. The forward recursive equations compute the kinematics information (angular velocity, angular acceleration, linear acceleration, total force and moment) of each link from the base coordinate system to the end-effector while the backward recursive equations compute the necessary torque to be applied at each joint actuator from the end-effector to the base coordinate system.

Because of the nature of the N-E formulation and its method of systematically computing the joint torques, the computations are of order $O(n)$. These equations involve about 690 multiplications and 621 additions for a six-jointed PUMA manipulator per trajectory set point. The formulation takes about 3 ms to compute the feedback joint torques using a PDP 11/45 computer [15]. This may be fast enough for real-time control (depending on the arm's natural frequency) if one does not need to process other external sensor feedback signals.

Due to its recursive in nature, it is very difficult to obtain a set of closed form differential equations from the N-E formulation. Consequently it is difficult to design an optimum control system using the N-E equations of motion.

We shall use the N-E equations of motion to compute the nominal torques along a preplanned trajectory and the L-E equations of motion to derive the perturbation equations of motion in the next section.

2.2. Perturbation Equations of Motion

As mentioned earlier, most feedback control laws are based on simplified dynamic equations. However the approach works well only at slow speeds of movement. At high speeds of movement the Coriolis and Centrifugal forces are major components of the dynamic equations. The error in the computed torques can not be corrected with feedback because of excessive requirements on the required correction torques. If one uses the complete Lagrange-Euler equations of motion to obtain a nonlinear feedback control law, the computation of the control law may become increasingly plagued by the large quantity of "number crunching". A better control solution is to use the perturbation feedback control to control the manipulator in the vicinity of a desired trajectory.

In general the dynamic model of a manipulator can be described as in Eq. 1. Define $\mathbf{x} = [\vartheta, \dot{\vartheta}]^T$ and $\mathbf{u} = \tau$. Then the dynamic equations of a manipulator can be rewritten as:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t) \tag{2}$$

where $\mathbf{x}(t) \in R^{2n}$, $\mathbf{u}(t) \in R^n$, $t \in R^+$, $\mathbf{f} : R^{2n} \times R^n \times R^+ \to R^{2n}$ and continuously differentiable, and n is the number of degree of freedom of the manipulator.

With this formulation, the objective is to find a feedback control law $\mathbf{u}(t) = \mathbf{g}(\mathbf{x}(t))$ such that the closed loop control system $\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{g}(\mathbf{x}(t)), t)$ is asymptotically stable and tracks the desired trajectory as closely as possible with wide range of payloads.

Since the above equations of motion describe the complete robot arm dynamics, the desired torques for each trajectory set point can be computed (in open-loop fashion) "quite accurately" from the N-E equations of motion. These computed torques can be treated as the nominal torque values. Because of the existence of modeling errors and the disturbances in the system, an appropriate variable feedback gains adjustment algorithm must be devised.

If the dynamic model is relatively accurate, then the joint errors will be small and the dynamic equations of a manipulator can be expanded in the vicinity of a known nominal trajectory set points to obtain the associated perturbation equations.

Suppose that given a nominal trajectory from a trajectory planning system, the nominal torques $\mathbf{u}_n(t)$ can be computed rapidly from the N-E equations of motion using the nominal states $\mathbf{x}_n(t)$ from a planned trajectory, then $\mathbf{u}_n(t)$ and $\mathbf{x}_n(t)$ satisfy Eq. 2 or:

$$\dot{x}_n(t) = f(x_n(t), u_n(t), t) \tag{3}$$

Using the Taylor series expansion on Eq. 2 about the "nominal" trajectory and subtracting Eq. 3 from it, we obtain:

$$\delta\dot{x}(t) = \nabla_x f|_n \, \delta x(t) + \nabla_u f|_n \delta u(t) \tag{4}$$

where $\nabla_x f|_n$ and $\nabla_u f|_n$ are the gradients of $f(x,u,t)$ evaluating at x_n and u_n respectively, $\delta x(t) = x(t) - x_n(t)$, and $\delta u(t) = u(t) - u_n(t)$. Let $\nabla_x f|_n \equiv A(t)$ and $\nabla_u f|_n \equiv B(t)$, then we have the associated perturbation equations for this control system:

$$\delta\dot{x}(t) = A(t)\,\delta x(t) + B(t)\,\delta u(t) \tag{5}$$

As a result of this formulation, the control problem of a manipulator is reduced to determining $\delta u(t)$ which drives $\delta x(t)$ to zero. The torques for the joint actuators consist of nominal torques $u_n(t)$ computed from the N-E equations of motion and the variational torques $\delta u(t)$ computed from the feedback control law associated with the perturbation equations. The main advantage of this formulation is the reduction of a nonlinear control system problem to a linear control system problem about a nominal trajectory and $\delta u(t)$ is only responsible for providing control efforts which compensate the necessary correction torques for small deviation from the nominal trajectory. The proposed control block diagram is shown in Figure 1.

2.3. Parameter Identification of The Perturbation Equations

The design of a feedback control law, $\delta u(t) = h(\delta x(t)) = K(t)\delta x(t)$, about the nominal trajectory is still quite difficult because the unknown parameters in the perturbation equations are slowly time-varying. Thus parameter identification techniques and adaptive mechanism must be used to identify the unknown parameters $A(t)$ and $B(t)$ in Eq. 5.

The problem of identification can be formulated as an evaluation of a system model representing the essential aspects of an existing system and presenting knowledge of that system in a usable form. So it is not expected that an exact mathematical description of the physical system need to be obtained but a model "fitted" so that an adaptive control may be obtained.

The mathematical approaches used in the identification scheme are either of the deterministic or stochastic types. Most commercial robots use incremental encoder to measure the angular positions and velocities. This greatly reduces the measurement noise. Because of this reason we choose the deterministic approach to make the problem simpler. In this case the noise present in the manipulator is assumed to be negligible and the uncertain parameters of the perturbation equations can be identified using simple methods.

The choice of model structure is one of the most important steps in the formulation of the identification problem. The choice will influence the identification characteristics such as the computational effort, the way in which the results of the identification can be used in subsequent operation. Due to the assumed digital computer application for identification and subsequent control, a linear discrete-time model must be obtained from $\delta\dot{x}(t)=A(t)\delta x(t)+B(t)\delta u(t)$.

Let $\delta x(t) \equiv x(t)$ and $\delta u(t) \equiv u(t)$. Using Euler transformation to discretize Eq. 5, we obtain the following discrete-time linear equations:

$$x(k+1)=A(k)\,x(k) + B(k)\,u(k) \tag{6}$$

$$y(k)=x(k) \tag{7}$$

$$A(k) = \begin{bmatrix} a_{11}(k) & \cdots & a_{1p}(k) \\ \vdots & & \vdots \\ a_{p1}(k) & \cdots & a_{pp}(k) \end{bmatrix} ; \quad B(k) = \begin{bmatrix} b_{11}(k) & \cdots & b_{1n}(k) \\ \vdots & & \vdots \\ b_{p1}(k) & \cdots & b_{pn}(k) \end{bmatrix} \tag{8}$$

where $p = 2n$, $x(k)$ and $y(k)$ are $2n\times1$ perturbed state and output vectors respectively, $u(k)$ is an $n\times1$ control input vector, $A(k)$ and $B(k)$ are $2n\times2n$ and $2n\times n$ matrices respectively, and n is the number of joint of the manipulator. With this model, $6n^2$ parameters need to be identified.

Among various identification methods applicable to the above model, we choose the recursive least square parameter identification method (RLS) because this method is conceptually simpler and relatively easy to implement with high speed of adaptation.

In this identification mechanism, we make the following assumptions:

(1) The parameters of the system is slowly time-varying but its variation speed is much slower than the adaptation speed.

(2) All the state variables $x = (\vartheta,\dot{\vartheta})^T$ of Eq. 6 are measurable.

(3) Measurement noise is negligible.

By expanding Eq. 6 term by term, we obtain the following scalar equations:

$$
\begin{aligned}
x_1(k+1) &= a_{11}(k)x_1(k) + a_{12}(k)x_2(k) + \cdots + a_{1p}(k)x_p(k) \\
&\quad + b_{11}(k)u_1(k) + b_{12}(k)u_2(k) + \cdots + b_{1n}(k)u_n(k) \\
&\quad\vdots \\
x_p(k+1) &= a_{p1}(k)x_1(k) + a_{p2}(k)x_2(k) + \cdots + a_{pp}(k)x_p(k) \\
&\quad + b_{p1}(k)u_1(k) + b_{p2}(k)u_2(k) + \cdots + b_{pn}(k)u_n(k)
\end{aligned} \tag{9}
$$

Define:

$$\vartheta_{ik} = [a_{i1}(k), a_{i2}(k), \cdots, a_{ip}(k), b_{i1}(k), \cdots, b_{in}(k)]^T \tag{10}$$

$$\varphi_k = [x_1(k), x_2(k), \cdots, x_p(k), u_1(k), \cdots, u_n(k)]^T \tag{11}$$

$$x(k) = [x_1(k), x_2(k), \cdots, x_p(k)]^T = [y_{1k}, y_{2k}, \cdots, y_{pk}]^T \tag{12}$$

Eq. 9 can be rewritten as follow:

$$y_{lk+1} = \varphi_k^T \vartheta_{lk} \quad where\ l = 1,2,\cdots p \text{ and } p = 2n \tag{13}$$

Based on the above input/output relationship, the recursive least square parameter identification algorithm is found to be:

$$\hat{\vartheta}_{lk+1} = \hat{\vartheta}_{lk} - P_k\varphi_k [\varphi_k^T P_k \varphi_k + r]^{-1} \ [\varphi_k^T \hat{\vartheta}_{lk} - y_{lk+1}] \tag{14}$$

$$P_{k+1} = [P_k - P_k\varphi_k [\varphi_k^T P_k \varphi_k + r]^{-1} \varphi_k^T P_k] \ r^{-1} \tag{15}$$

where $\hat{\vartheta}_{lk+1}$ is the estimated values of ϑ_{lk+1} and r is a weighting factor between 0 and 1. The use of r is common when tracking slowly time-varying system parameters.

In this RLS parameter identification method, both the model and the dynamic system have the same input **u**(k) and the estimated model parameters $\hat{\vartheta}_{lk+1}$ are adjusted as a function of the error between the output y_{lk+1} from the actual system as in Eq. 13 and the output $\varphi_k^T \hat{\vartheta}_{lk}$ from the reference model.

2.4. Controller Design for The Perturbation Equations of Motion

The parameters of the perturbation equations of motion obtained from the identification scheme can sometimes yield unstable system when applied to optimum control system design. The reason is that numerical solutions could be sensitive to deviations of the model parameter values from their true values. If the system parameter identification and the control design are performed in each sampling period successively, then the control system obtained will be stable asymptotically. Since the identification and control approaches for multi-input multi-output (MIMO) system require considerable amounts of computational time, we would like to keep all algorithms to moderate complexity.

For designing the controller for the perturbation equations, many suitable control schemes can be applied. Simple position-integral-derivative (PID) controller or pole-assignment can be used. Another method is the use of linear quadratic (LQ) method to generate control laws.

Consider the discrete time-varying linear system with the following well known performance criterion:

$$J_1 = \frac{1}{2}x(N)Sx(N) + \frac{1}{2}\sum_{k=0}^{N-1}\left[x^T(k)Qx(k) + u^T(k)Ru(k)\right] \tag{16}$$

The optimal control input $u(k)$ which minimizes the above criterion is given by:

$$u(k) = -K(k)x(k) \tag{17}$$

where

$$K(k) = \left[R + B^T(k)P(k+1)B(k)\right]^{-1}B^T(k)P(k+1)A(k) \tag{18}$$

$$P(k) = A^T(k)P(k+1)A(k) + Q - B^T(k)P(k+1)A(k)K(k) \tag{19}$$

and

$$P(N) = S \tag{20}$$

The optimal control law can be obtained only by solving the discrete Riccati equations in backward and in this particular situation (i.e., simultaneous identification and control), it is very difficult to use the above criterion because of computation time limitation.

To overcome this problem, we consider another performance criterion, namely one step control, subject to the above discrete time-varying system,

$$J_2(k) = \frac{1}{2}\left[x^T(k+1)Qx(k+1) + u^T(k)Ru(k)\right] \tag{21}$$

Then the optimal control input $u(k)$ based on the above criterion is given by:

$$u(k) = -K(k)x(k) \tag{22}$$

where

$$K(k) = \left[R + B^T(k)QB(k)\right]^{-1} B^T(k)QA(k) \tag{23}$$

In this case, we minimize the performance criterion $J_2(k)$ by finding the input u(k) which minimizes the immediate loss.

The optimal control is obtained under the assumptions that the system parameters have been identified quite accurately. The parameters used in Eqs. 22-23 are from the RLS identification scheme. It is known that this adaptive controller tunes the parameters of the system for an optimal control input.

3. Computer Simulation: A Three-jointed PUMA Robot Arm

In this section, a computer simulation study of a three-jointed PUMA manipulator was carried out on a VAX 11-780 computer to evaluate the validity of the use of the perturbation equations and the performance of the proposed adaptive control strategy.

In general, a Lagrange-Euler formulation for a manipulator can be written as $D(\vartheta)\,\ddot{\vartheta} + H(\dot{\vartheta},\vartheta) + G(\vartheta) = \tau$. If the Lagrange-Euler formulation of a three-jointed PUMA manipulator based on the 4x4 homogeneous transformation matrices [Pau81][Lee82] is expanded in general terms, the following equations are obtained:

$$D_{11}\ddot{\vartheta}_1 + D_{12}\ddot{\vartheta}_2 + D_{13}\ddot{\vartheta}_3 + H_{111}\dot{\vartheta}_1^2 + 2H_{112}\dot{\vartheta}_1\dot{\vartheta}_2 + 2H_{113}\dot{\vartheta}_1\dot{\vartheta}_3$$

$$+ H_{122}\dot{\vartheta}_2^2 + 2H_{123}\dot{\vartheta}_2\dot{\vartheta}_3 + H_{133}\dot{\vartheta}_3^2 + G_1 = \tau_1$$

$$D_{21}\ddot{\vartheta}_1 + D_{22}\ddot{\vartheta}_2 + D_{23}\ddot{\vartheta}_3 + H_{211}\dot{\vartheta}_1^2 + 2H_{212}\dot{\vartheta}_1\dot{\vartheta}_2 + 2H_{213}\dot{\vartheta}_1\dot{\vartheta}_3$$

$$+ H_{222}\dot{\vartheta}_2^2 + 2H_{223}\dot{\vartheta}_2\dot{\vartheta}_3 + H_{233}\dot{\vartheta}_3^2 + G_2 = \tau_2$$

$$D_{31}\ddot{\vartheta}_1 + D_{32}\ddot{\vartheta}_2 + D_{33}\ddot{\vartheta}_3 + H_{311}\dot{\vartheta}_1^2 + 2H_{312}\dot{\vartheta}_1\dot{\vartheta}_2 + 2H_{313}\dot{\vartheta}_1\dot{\vartheta}_3$$

$$+ H_{322}\dot{\vartheta}_2^2 + 2H_{323}\dot{\vartheta}_2\dot{\vartheta}_3 + H_{333}\dot{\vartheta}_3^2 + G_3 = \tau_3$$

The dynamic coefficients of the above equations (i.e. D 's, H 's, G 's) can be rewritten as follow:

(1) The coefficients of $D(\vartheta)$ are:

$$\begin{aligned} D_{11} = & J_{111} + J_{133} + J_{211}C_2^2 + 2J_{214}a_2C_2^2 + J_{222}S_2^2 + J_{233} + 2J_{234}d_2 \\ & + J_{244}(a_2^2C_2^2 + d_2^2) + J_{311}(S_3^2S_2^2 - 2S_3C_3S_2C_2 + C_3^2C_2^2) + J_{322} \\ & + J_{333}(S_3^2C_2^2 + 2s_3C_3S_2C_2 + C_3^2S_2^2) + J_{344}(a_2^2C_2^2 + d_2^2) \\ & + 2J_{334}(a_2S_3C_2^2 + a_2C_3S_2C_2) \end{aligned}$$

$$\begin{aligned} D_{22} = & J_{211} + J_{222} + 2J_{214}a_2 + J_{244}a_2^2 + J_{311} + J_{333} + J_{344}a_2^2 \\ & + 2J_{334}a_2S_3 \end{aligned}$$

$$D_{33} = J_{331} + J_{333}$$

$$\begin{aligned} D_{12} = D_{21} = & J_{213}S_2 + J_{214}d_2S_2 + J_{234}a_2S_2 + J_{244}a_2d_2S_2 + J_{344}a_2d_2S_2 \\ & + J_{334}d_2(S_3S_2 - C_3C_2) \end{aligned}$$

$$D_{13} = D_{31} = J_{334}d_2(S_3S_2 - C_3C_2)$$

$$D_{23} = D_{32} = J_{311} + J_{333} + J_{334}a_2S_3$$

(2) The coefficients of $H(\dot{\vartheta},\vartheta)$ are:

$$H_{111} = 0$$

$$\begin{aligned} H_{122} = & J_{213}C_2 + J_{214}d_2C_2 + J_{234}a_2C_2 + J_{244}a_2d_2C_2 \\ & + J_{344}a_2d_2C_2 + J_{334}d_2(S_3C_2 + C_3S_2) \end{aligned}$$

$$H_{133} = J_{334}d_2(S_3C_2 + C_3S_2)$$

$$H_{211} = J_{211}S_2C_2 - J_{222}S_2C_2 + 2J_{214}a_2S_2c_2 + J_{244}a_2^2S_2C_2$$
$$+ (J_{333}-J_{311})(S_3^2S_2C_2 + S_3C_3S_2^2 - S_3C_3C_2^2 - C_3^3S_2C_2)$$
$$+ J_{344}a_2^2S_2C_2 + J_{334}(2a_2S_3S_2C_2 + a_2C_3S_2^2 - a_2C_3C_2^2)$$

$$H_{222} = 0$$

$$H_{233} = J_{334}a_2C_3$$

$$H_{311} = (J_{333}-J_{311})(S_3^2S_2C_2 + S_3C_3S_2^2 - S_3C_3C_2^2 - C_3^2S_2C_2)$$
$$+ J_{334}(a_2S_3S_2C_2 - a_2C_3C_2^2)$$

$$H_{322} = -J_{334}a_2C_3$$

$$H_{333} = 0$$

$$H_{112} = -H_{211}$$

$$H_{113} = -H_{311}$$

$$H_{123} = H_{133}$$

$$H_{212} = 0$$

$$H_{213} = 0$$

$$H_{223} = -H_{322}$$

$$H_{312} = 0$$

$$H_{313} = 0$$

$$H_{323} = 0$$

(3) The coefficients of $G(\vartheta)$ are:

$G_1 = 0$

$G_2 = -(a_2 g_{34} C_2 + a_2 g_{24} C_2 + S_3 g_{33} C_2 + C_3 g_{33} S_2 + C_2 g_{21})$

$G_3 = -g_{33}(S_3 C_2 + C_3 S_2)$

where

$S_l = \sin(\vartheta_l)$

$C_l = \cos(\vartheta_l)$

$[g_{l1}\ \ g_{l2}\ \ g_{l3}]^T = [m_l g\bar{x}_l\ \ m_l g\bar{y}_l\ \ m_l g\bar{z}_l]^T$ and $g = 9.8062 m/\sec^2$

$$\mathbf{J}_l = \begin{bmatrix} J_{l11} & J_{l12} & J_{l13} & J_{l14} \\ J_{l12} & J_{l22} & J_{l23} & J_{l24} \\ J_{l13} & J_{l23} & J_{l33} & J_{l34} \\ J_{l14} & J_{l24} & J_{l34} & J_{l44} \end{bmatrix}$$

$$= \begin{bmatrix} \frac{-I_{lxx} + I_{lyy} + I_{lzz}}{2} + m_l\bar{x}_l^2 & m_l\bar{x}_l\bar{y}_l & m_l\bar{x}_l\bar{z}_l & m_l\bar{x}_l \\ m_l\bar{x}_l\bar{y}_l & \frac{I_{lxx} - I_{lyy} + I_{lzz}}{2} + m_l\bar{y}_l^2 & m_l\bar{y}_l\bar{z}_l & m_l\bar{y}_l \\ m_l\bar{x}_l\bar{z}_l & m_l\bar{y}_l\bar{z}_l & \frac{I_{lxx} + I_{lyy} - I_{lzz}}{2} + m_l\bar{z}_l^2 & m_l\bar{z}_l \\ m_l\bar{x}_l & m_l\bar{y}_l & m_l\bar{z}_l & m_l \end{bmatrix}$$

$$\mathbf{I}_l = \begin{bmatrix} I_{lxx} & 0 & 0 \\ 0 & I_{lyy} & 0 \\ 0 & 0 & I_{lzz} \end{bmatrix}$$; *we assume the products of inertia are negligible.*

$\bar{\mathbf{s}}_l = (\bar{x}_l, \bar{y}_l, \bar{z}_l)$ is the position vector of the center of mass of link l with respect to the l^{th} coordinate system.

The state equations of a three-jointed PUMA manipulator can be obtained by simple manipulations of Eq. 1. Define the inverse matrix of the acceleration-related matrix as follow:

$$D(\vartheta)^{-1} = \begin{bmatrix} D_{11} & D_{12} & D_{13} \\ D_{12} & D_{22} & D_{23} \\ D_{13} & D_{23} & D_{33} \end{bmatrix}^{-1} = \begin{bmatrix} \alpha_{11} & \alpha_{12} & \alpha_{13} \\ \alpha_{12} & \alpha_{22} & \alpha_{23} \\ \alpha_{13} & \alpha_{23} & \alpha_{33} \end{bmatrix}$$

then the dynamic equations of a PUMA manipulator become:

$$\begin{bmatrix} \ddot{\vartheta}_1 \\ \ddot{\vartheta}_2 \\ \ddot{\vartheta}_3 \end{bmatrix} = \begin{bmatrix} \alpha_{11} & \alpha_{12} & \alpha_{13} \\ \alpha_{12} & \alpha_{22} & \alpha_{23} \\ \alpha_{13} & \alpha_{23} & \alpha_{33} \end{bmatrix} \times$$

$$\left\{ - \begin{bmatrix} H_{122}\dot{\vartheta}_2^2 + H_{133}\dot{\vartheta}_3^2 + 2H_{112}\dot{\vartheta}_1\dot{\vartheta}_2 + 2H_{113}\dot{\vartheta}_1\dot{\vartheta}_3 + 2H_{123}\dot{\vartheta}_2\dot{\vartheta}_3 \\ H_{211}\dot{\vartheta}_1^2 + H_{233}\dot{\vartheta}_3^2 + 2H_{223}\dot{\vartheta}_2\dot{\vartheta}_3 + G_2 \\ H_{311}\dot{\vartheta}_1^2 + H_{322}\dot{\vartheta}_2^2 + G_3 \end{bmatrix} + \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} \right\}$$

$$\overset{\Delta}{=} \begin{bmatrix} \alpha_{11} & \alpha_{12} & \alpha_{13} \\ \alpha_{12} & \alpha_{22} & \alpha_{23} \\ \alpha_{13} & \alpha_{23} & \alpha_{33} \end{bmatrix} \begin{bmatrix} \beta_1+\tau_1 \\ \beta_2+\tau_2 \\ \beta_3+\tau_3 \end{bmatrix}$$

If the state variables and the inputs are defined as:

$$\begin{array}{lll} x_1 = \vartheta_1 & x_4 = \dot{\vartheta}_1 & u_1 = \tau_1 \\ x_2 = \vartheta_2 & x_5 = \dot{\vartheta}_2 & u_2 = \tau_2 \\ x_3 = \vartheta_3 & x_6 = \dot{\vartheta}_3 & u_3 = \tau_3 \end{array}$$

then the following state equations for a three-jointed PUMA manipulator are obtained:

$$\dot{x}_1 = f_1(x) = x_4$$

$$\dot{x}_2 = f_2(x) = x_5$$

$$\dot{x}_3 = f_3(x) = x_6$$

$$\dot{x}_4 = f_4(x) = \alpha_{11}(\beta_1+u_1) + \alpha_{12}(\beta_2+u_2) + \alpha_{13}(\beta_3+u_3)$$

$$\dot{x}_5 = f_5(x) = \alpha_{12}(\beta_1+u_1) + \alpha_{22}(\beta_2+u_2) + \alpha_{23}(\beta_3+u_3)$$

$$\dot{x}_6 = f_6(x) = \alpha_{13}(\beta_1+u_1) + \alpha_{23}(\beta_2+u_2) + \alpha_{33}(\beta_3+u_3)$$

Applying the Taylor series expansion to the above equations in the vicinity of a nominal (or desired) trajectory and neglecting the higher order terms, the perturbation equations $\delta\dot{x}(t) = A(t)\delta x(t) + B(t)\delta u(t)$ are obtained. For convenience, let $\delta x \equiv x$ and $\delta u \equiv u$, then the following equations are obtained:

$$\dot{x}(t) = A(t)x(t) + B(t)u(t)$$

where

$$x(t) = x_{actual}(t) - x_{nominal}(t)$$

$$u(t) = u_{actual}(t) - u_{nominal}(t)$$

$$A(t) = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & \frac{\partial f_4(x)}{\partial x_2} & \frac{\partial f_4(x)}{\partial x_3} & \frac{\partial f_4(x)}{\partial x_4} & \frac{\partial f_4(x)}{\partial x_5} & \frac{\partial f_4(x)}{\partial x_6} \\ 0 & \frac{\partial f_5(x)}{\partial x_2} & \frac{\partial f_5(x)}{\partial x_3} & \frac{\partial f_5(x)}{\partial x_4} & \frac{\partial f_5(x)}{\partial x_5} & \frac{\partial f_5(x)}{\partial x_6} \\ 0 & \frac{\partial f_6(x)}{\partial x_2} & \frac{\partial f_6(x)}{\partial x_3} & \frac{\partial f_6(x)}{\partial x_4} & \frac{\partial f_6(x)}{\partial x_5} & \frac{\partial f_6(x)}{\partial x_6} \end{bmatrix} \text{ at } x_n, u_n$$

$$B(t) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{\partial f_4(x)}{\partial u_1} & \frac{\partial f_4(x)}{\partial u_2} & \frac{\partial f_4(x)}{\partial u_3} \\ \frac{\partial f_5(x)}{\partial u_1} & \frac{\partial f_5(x)}{\partial u_2} & \frac{\partial f_5(x)}{\partial u_3} \\ \frac{\partial f_6(x)}{\partial u_1} & \frac{\partial f_6(x)}{\partial u_2} & \frac{\partial f_6(x)}{\partial u_3} \end{bmatrix} \text{ at } x_n, u_n$$

We simulated the proposed adaptive control strategy and compared it with the PD control method based on the computed torque technique [15] for various loading conditions for the given trajectory.

The numerical values used in this simulation are:

$diag\ Q = [\ 0.1, 0.1, 0.1, 0.1, 0.1, 0.1]$

$diag\ R = [\ 0.000001, 0.000001, 0.000001]$

and the initial values of the unknown parameters in the perturbation equations are chosen to be:

$diag\ A(0) = [\ 0.5, 0.5, 0.5, 0.5, 0.5, 0.5]$

$$B(0) = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}^T$$

$diag\ P(0) = [\ 1500, 1500, \cdot, \cdot, \cdot, 1500]_{9 \times 9}$

The off-diagonal terms of the above matrices are set to be zero and the weighting factor, r, in Eqs. 14-15 is set to 0.95. Also the control gains K_v and K_p of the PD controller are set to 20 and 100 respectively.

With reference to the robot arm dynamic equation, the physical geometric parameters used in this simulation are:

$d_1 = .664$ *meter*

$a_2 = .432$ *meter*

$d_2 = .1495$ *meter*

$m = [\ 2.27, 15.91, 11.36]$ *Kg*

$daig\ I_1 = [\ 0.0071, 0.0267, 0.0267]\ Kg-meter^2$

$daig\ I_2 = [\ 0.1000, 0.7300, 0.8025]\ Kg-meter^2$

$daig\ I_3 = [\ 0.0222, 0.2160, 0.2245]\ Kg-meter^2$

$p_1^* = [\ 0., -0.664, 0.]^T\ meter$

$p_2^* = [\ 0.432, 0., 0.1495]^T\ meter$

$p_3^* = [\ 0., 0., 0.]^T\ meter$

$\bar{s}_1 = [\ 0., 0, 0.073]^T\ meter$

$\bar{s}_2 = [\ -0.432, 0., 0.]^T\ meter$

$\bar{s}_3 = [\ 0., 0., 0.1]^T\ meter$

where p^*_i is the position vector of the origin of the $i-1^{th}$ frame with respect to the i^{th} coordinate system and $\bar{s}_i$ is the position vector of the center of mass of link i with respect to the i^{th} coordinate system.

In the N-E formulation, there are several geometric parameter values need to be measured from the arm's structural configuration. Among these values are the location of the center of mass and the inertia tensor matrix of each link. It is relatively difficult to get the exact values of the inertia tensor matrices because of the asymmetry of the links of a PUMA robot arm and the non-uniform distribution of its masses (mainly from the DC motors). In this simulation, we assume that the dynamic equations of a PUMA manipulator are known exactly except inertia tensor matrices. We also assume that the robot arm does not know the weight of the loads when it picks them up.

In the simulation, the three-jointed manipulator moves from an initial joint angles $\vartheta_{initial}$ = (0°, 45°, 45°) to a final joint angles ϑ_{final} = (90°, −45°, 135°). The required time for this motion is 1 second. In this trajectory, the PUMA robot arm is fully stretched at 0.5 seconds. At this position, $(\vartheta_1, \vartheta_2, \vartheta_3)$ = (45°, 0°, 90°), the torques due to the gravity have the maximum values and the absolute values of joint velocity of the arm also becomes the maximum. The accelerations are sharply changed from the maximum values to the minimum values or vice versa.

To show the validity of the use of the perturbation equations, the nominal joint position values, velocity values and torques are perturbed with small error. When the manipulator picks up the maximum load ($\approx 2.3Kg$), the position and velocity errors from the proposed adaptive controller are less than one degree for all trajectory set points. For this reason, the perturbed error is set to be one degree for all joints. As shown in Figures 2-4, the contributions of the higher order terms resulted from the linearization are negligible for all the motion trajectories. Hence the perturbed equation of motion derived are valid for all trajectory.

Figure 5 shows the computer simulation program implementation flow chart. The simulation results are shown in Figures 6-11 and tabulated in Table 1. In Table 1, the performance of both controllers are compared for three different load conditions: (a) No-load and 10% error in inertia tensor matrix, (b) Half of maximum load and 10% error in inertia tensor matrix and (c) Maximum load and 10% error in inertia tensor matrix. In each case 10% error in inertia matrices means ±10 % error about its measured inertial values. For all the above cases, the adaptive controller shows better performance than the PD controller with constant feedback gains. The maximum error of each joint was calculated by assuming that link 2 and link 3 are each 0.5 meter long. Let the maximum absolute errors of joint one, joint two and joint three be $\mathbf{e}_1$, $\mathbf{e}_2$, and $\mathbf{e}_3$ respectively, then the resulting maximum error for the worst case can be found to be $\sqrt{\mathbf{e}_1^2 + (\mathbf{e}_2 + \mathbf{e}_3)^2}$. When the PUMA robot arm picks up maximum load, the magnitude of the applied torques are increased considerably, especially in joint two (see Figures 12-14). So we can expect relatively larger error in joint two as compared with other two joints (see Table 1). The comparison of the final position errors from these two controllers for the same trajectory is tabulated in Table 1.

Given A(0) and B(0), the performance of the adaptive controller is quite sensitive to the Q, R and P(0) values in Eqs. 14-15 and 21-23. In general, it is not easy to determine the proper values to achieve better performance. It is also difficult to figure out the correlations of these values with the overall performance of the controller especially in MIMO system because of the complexity of the dynamic equations of the PUMA robot arm.

4. Conclusion

An adaptive control based on the perturbation theory has been presented with computer simulation of a three-jointed PUMA robot arm. The proposed adaptive control was found to perform better for various loading conditions than the simple PD controller based on the computed torque technique. The torques for the joint actuators consist of nominal torques computed from the N-E equations of motion and the variational torques computed from the feedback control law associated with the perturbation equations. A clear advantage of such formulation is that the nominal torques and the variational torques can be computed separately and simultaneously.

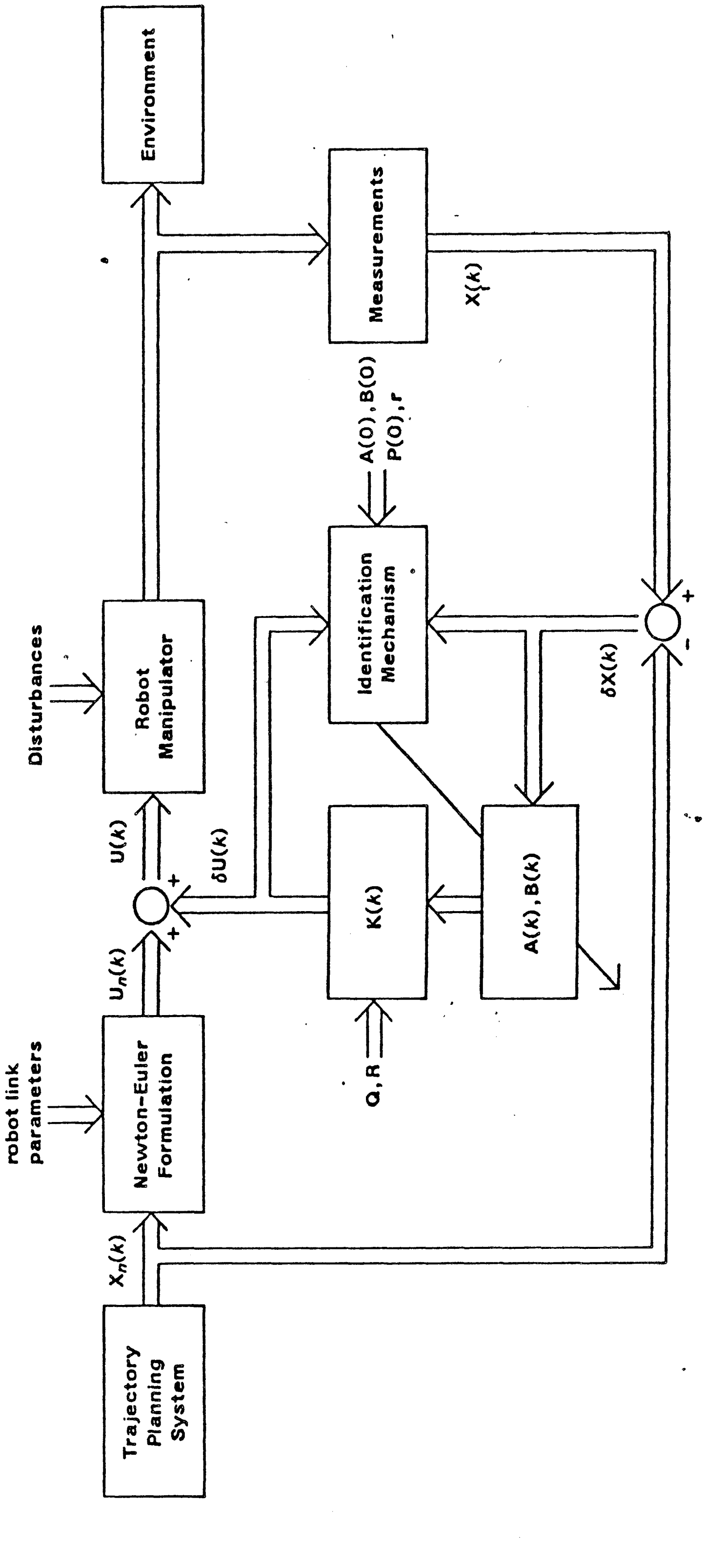

Figure 1 Control Block Diagram

(1) Set $k = 0$ where k is the k^{th} sampling period.

(2) Determine $\vartheta_d[k]$, $\dot{\vartheta}_d[k]$ and $\ddot{\vartheta}_d[k]$ from a preplanned trajectory. Also determine above values for the next sampling period.

(3) Compute the nominal torques $U_n(k)$ using the Newton-Euler equations of motion.

(4) Determine the applied torques $U_a(k) = U_n(k) + \delta U(k)$.

(5) Compute the coefficients of the dynamic equations of a PUMA manipulator using the Lagrange-Euler equations of motion as in Eq. 1.

(6) Integrate the dynamic equations of a PUMA manipulator derived from the Lagrange-Euler equations of motion using the 4^{th} order Runge-Kutta method. The outputs are $\vartheta_a[k+1]$ and $\dot{\vartheta}_a[k+1]$.

(7) Compute position and velocity errors.

$$\delta X[k+1] = [\ \vartheta_a[k+1] - \vartheta_d[k+1]\ ,\ \dot{\vartheta}_a[K+1] - \dot{\vartheta}_d[k+1]\ ]^T$$

(8) Using $\delta X[k+1]$ and $\delta U[k]$ and the identification algorithm as in Eqs. 14-15, update the parameters of the system as in Eq. 5.

(9) Using the updated parameters, compute the control gains as in Eq. 23 and the perturbed inputs $\delta U(k+1)$.

(10) Set $k = k + 1$.

(11) Is $k = N$? (Total of N sampling periods). If yes, Stop. Else go to step 2.

Figure 2 Flow-Chart of Computer Simulation for the Proposed Controller

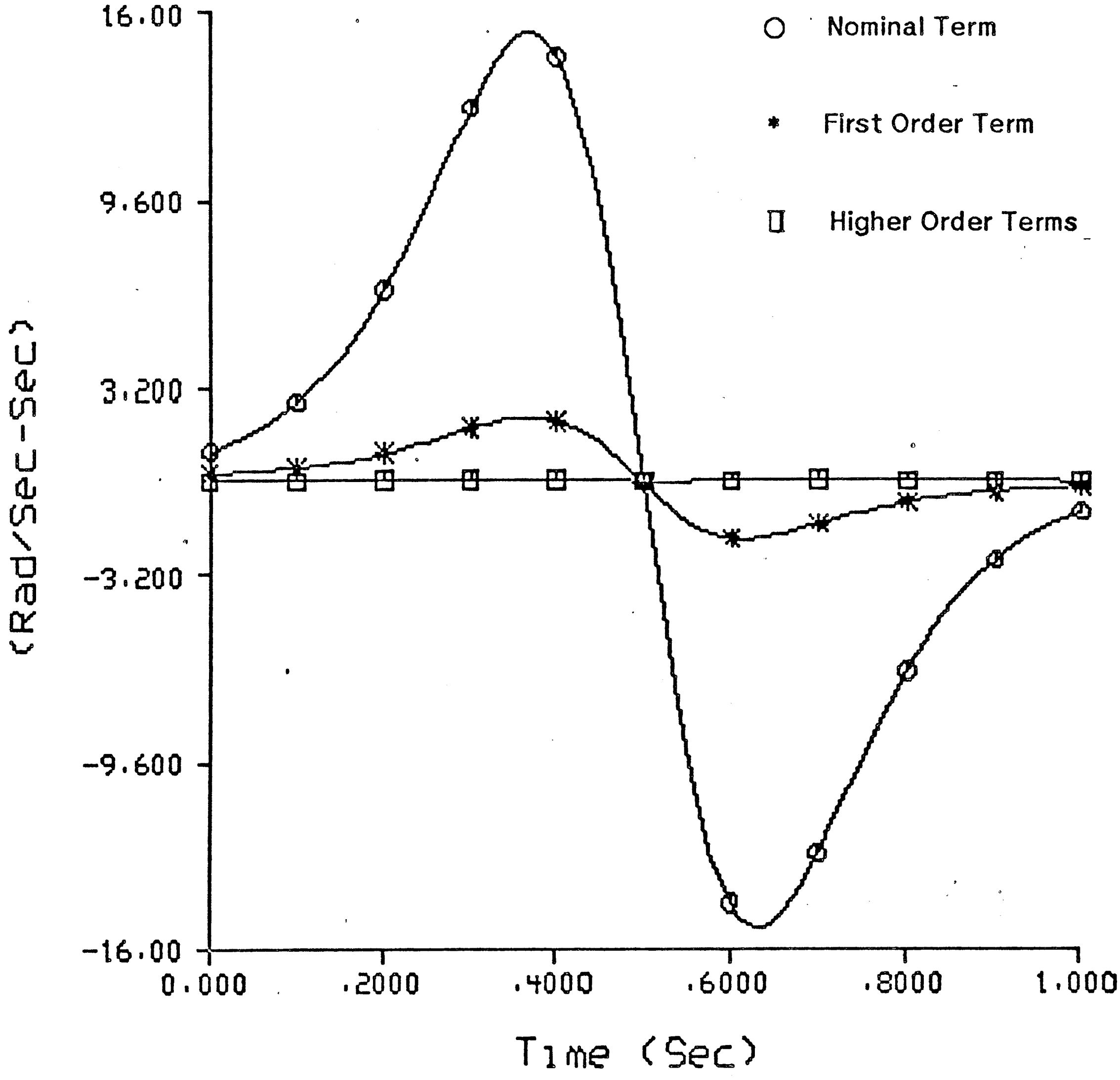

Figure 3 Contributions of the First Order Term and the Higher Order Terms in the Linearization of the Dynamic Model (Joint 1)

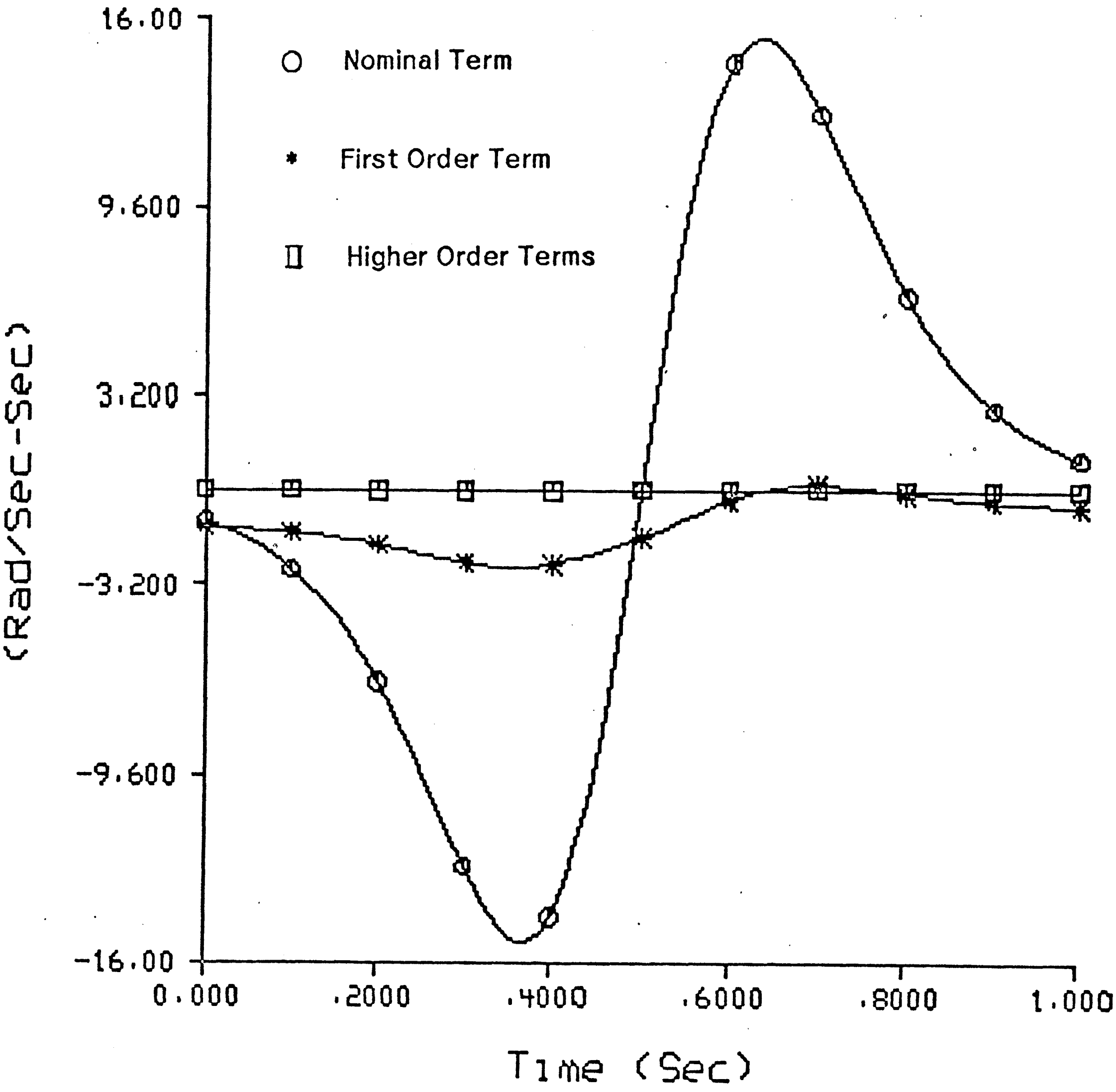

Figure 4 Contributions of the First Order Term and the Higher Order Terms in the Linearization of the Dynamic Model (Joint 2)

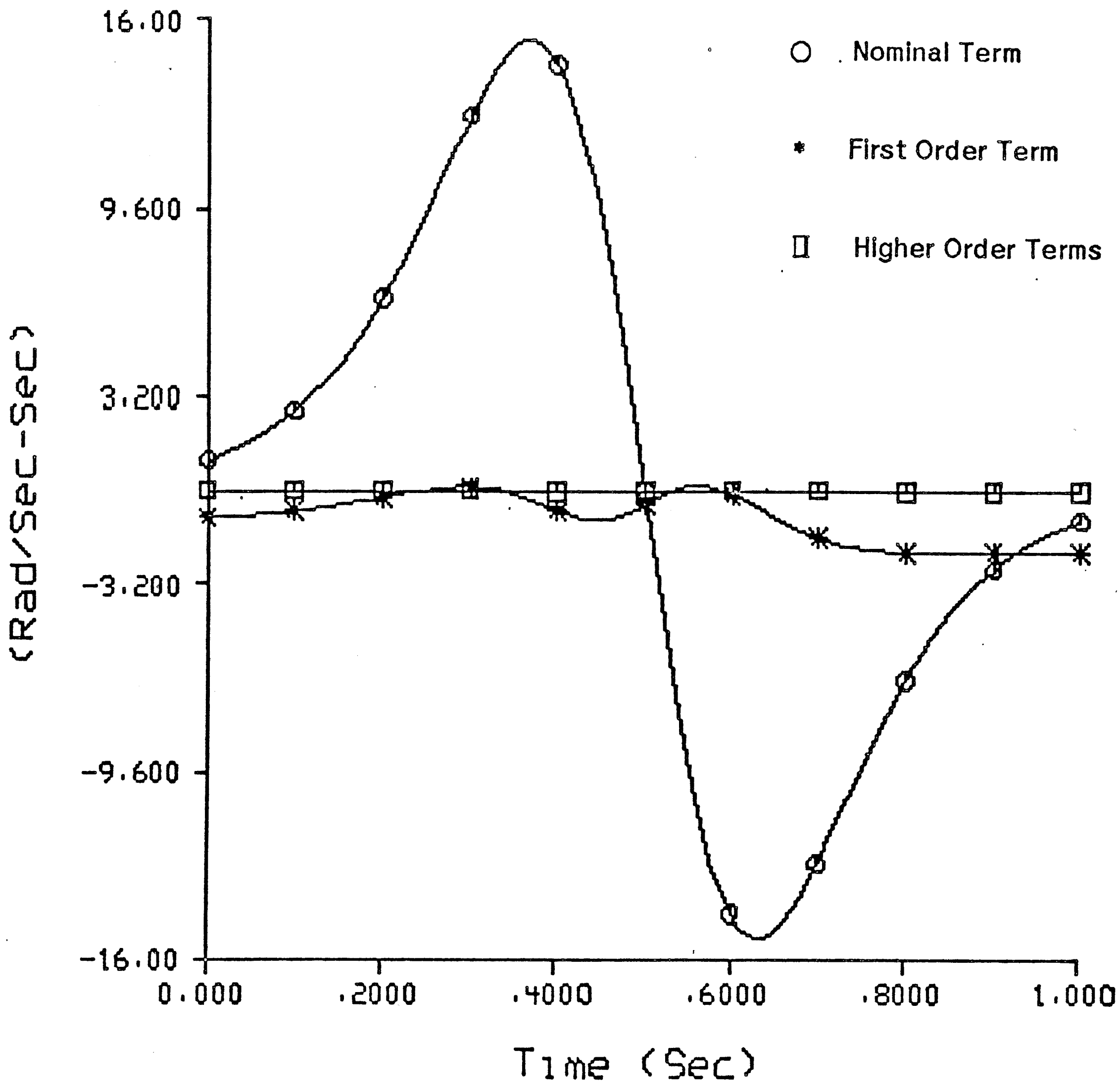

Figure 5 Contributions of the First Order Term and the Higher Order Terms in the Linearization of the Dynamic Model (Joint 3)

Various Loading Conditions	Joint	PD Controller: Trajectory Tracking Max. Error (degree)	PD Controller: Trajectory Tracking Max. Error (mm)	PD Controller: Final Position Error (mm)	Adaptive Controller: Trajectory Tracking Max. Error (degree)	Adaptive Controller: Trajectory Tracking Max. Error (mm)	Adaptive Controller: Final Position Error (mm)
No load and 10% error in inertia tensor	1	0.1858	3.24	1.59	0.1436	2.51	2.39
	2	0.3234	5.64	0.45	0.0810	1.41	0.99
	3	0.5214	4.55	0.49	0.0765	0.67	0.58
1/2 max. load and 10% error in inertia tensor	1	0.2286	3.99	2.17	0.0460	0.80	0.12
	2	0.4351	7.59	0.24	0.2818	4.92	4.64
	3	0.7774	6.78	1.75	0.1076	0.94	0.31
Max. load and 10% error in inertia tensor	1	0.2719	4.75	2.49	0.0771	1.35	0.75
	2	0.5042	8.80	1.02	0.3988	6.96	6.32
	3	0.9008	7.86	2.69	0.0663	0.58	0.07

Table 1 Comparison of Controllers

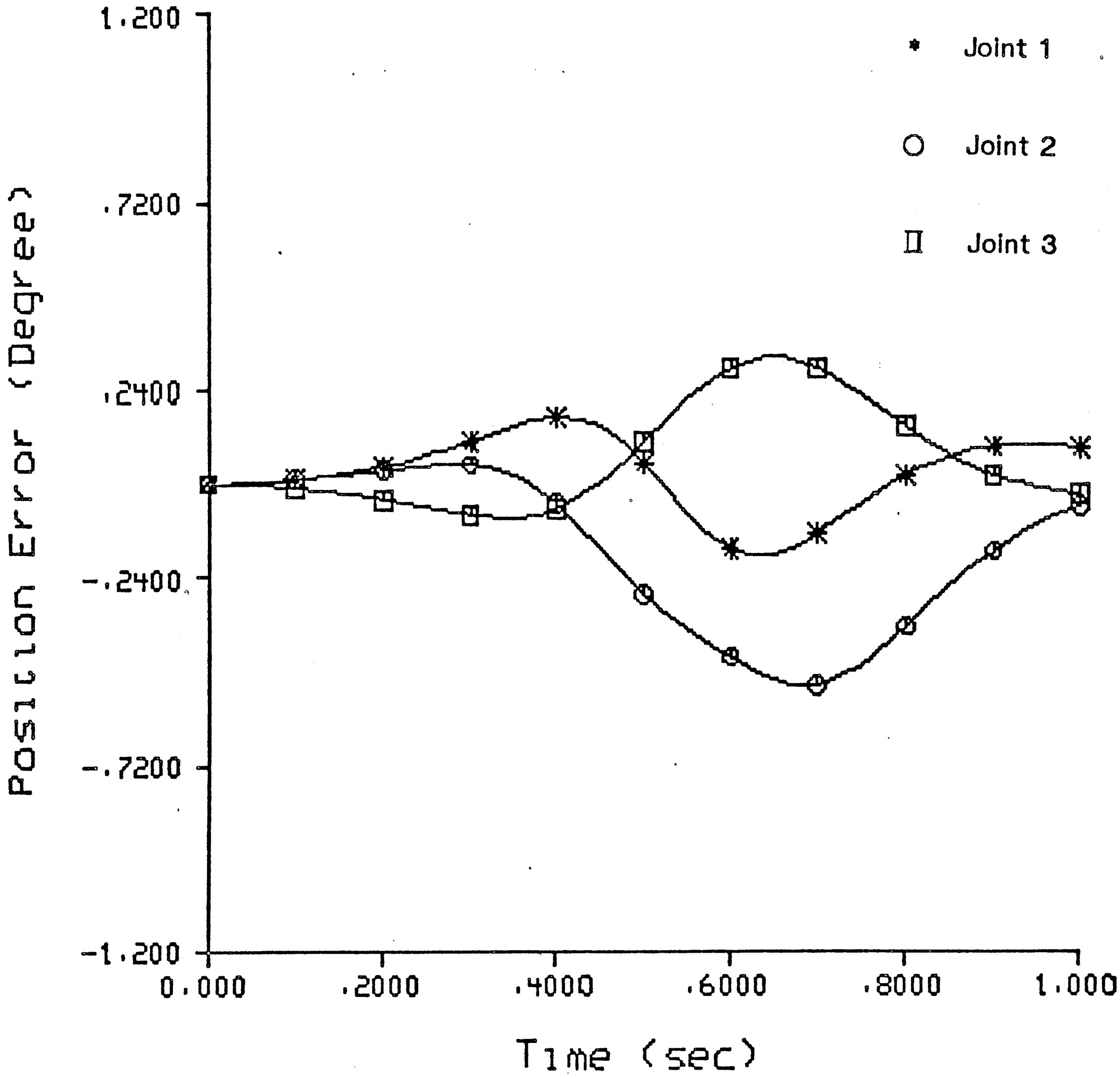

Figure 6 PD Controller: Case (a)

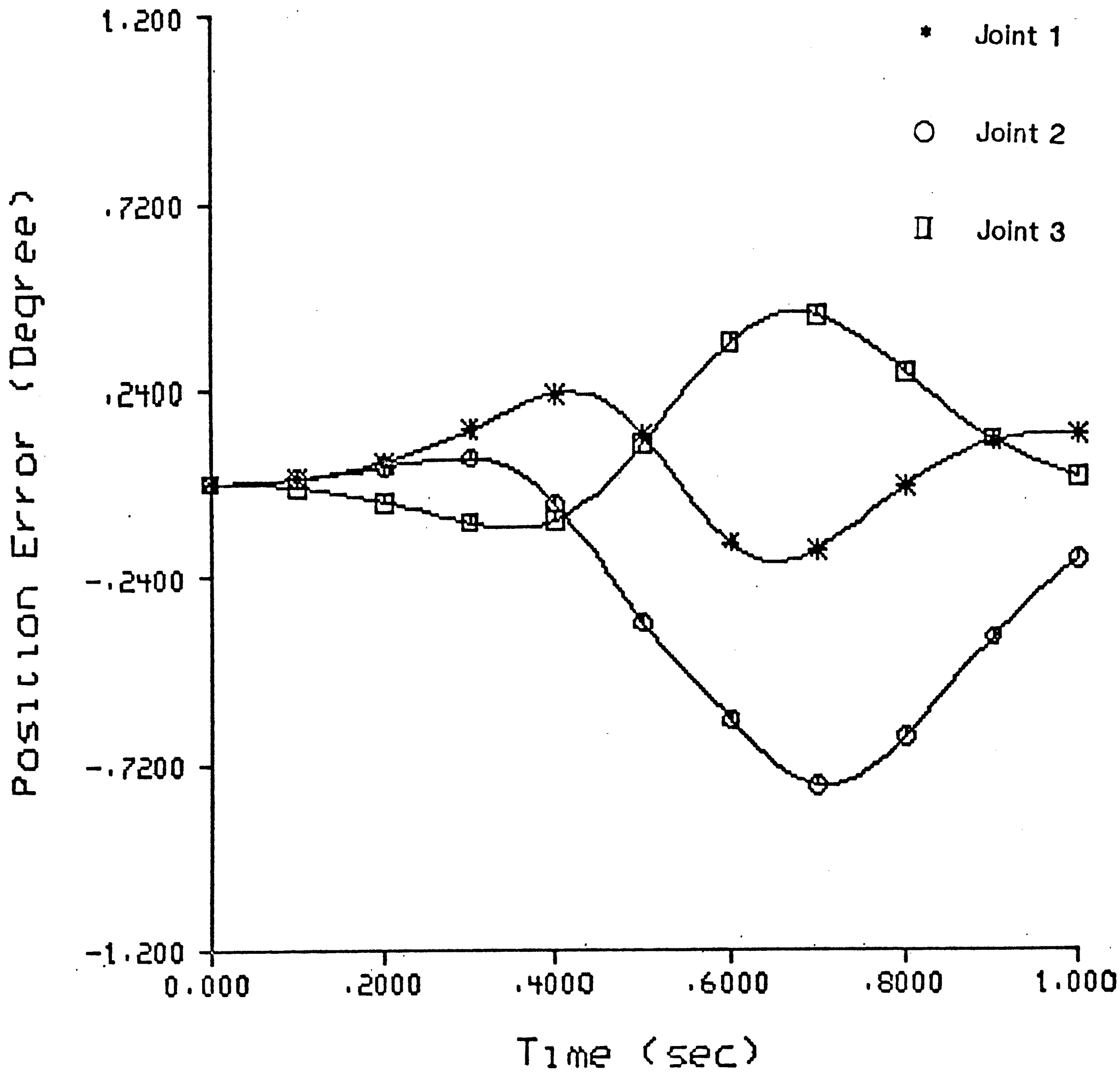

Figure 7 PD Controller: Case (b)

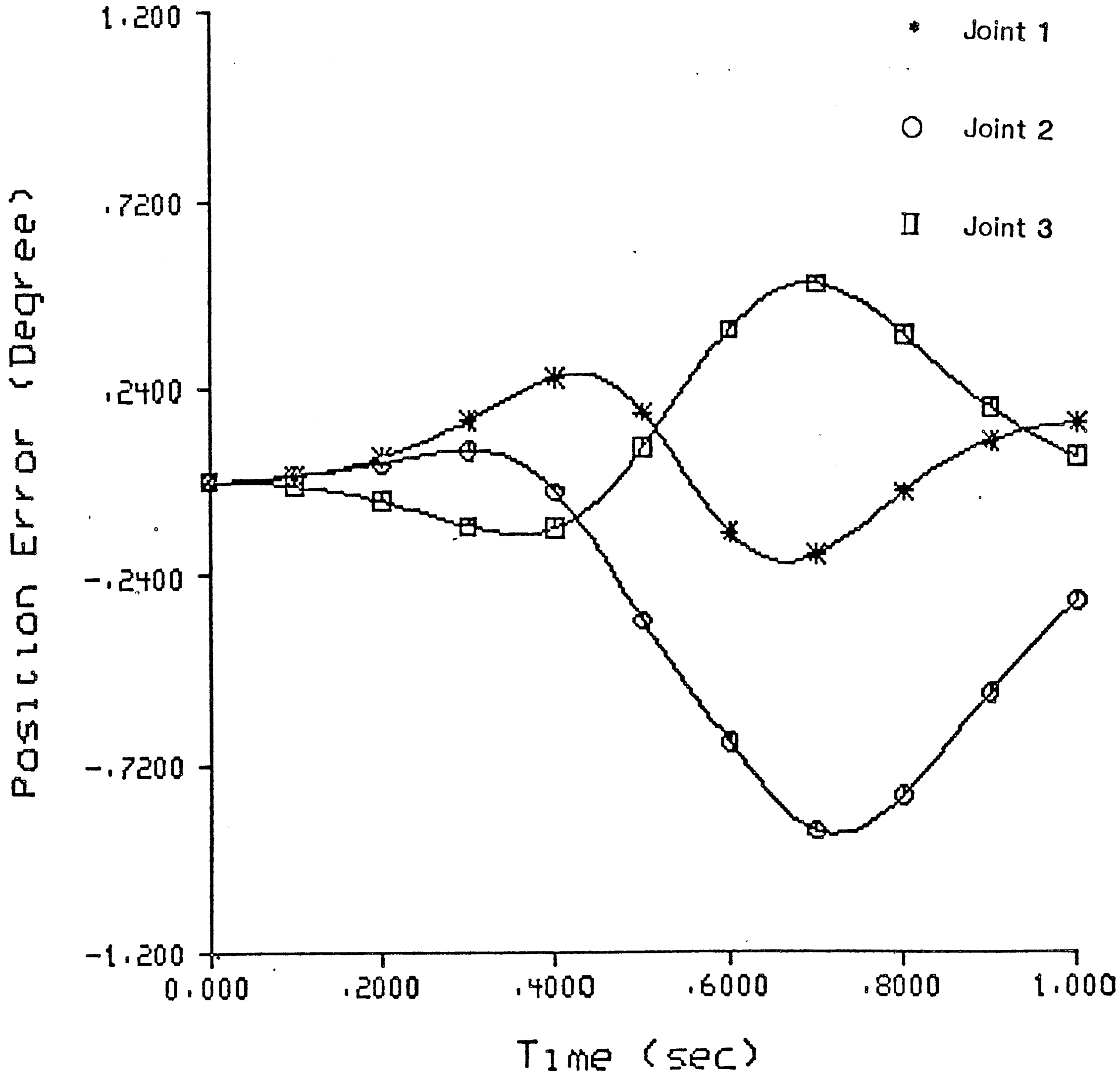

Figure 8 PD Controller: Case (c)

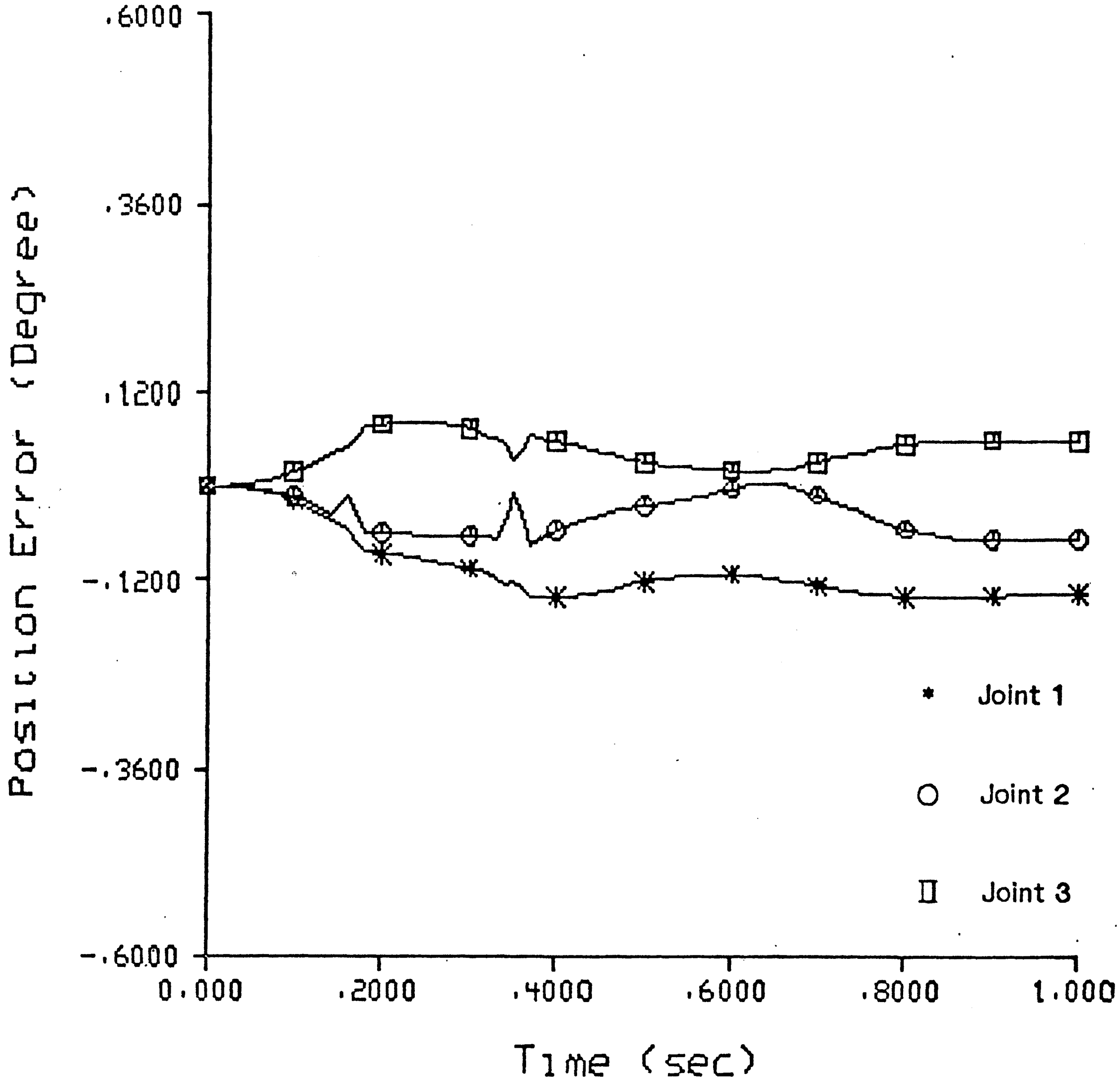

Figure 9 Adaptive Controller: Case (a)

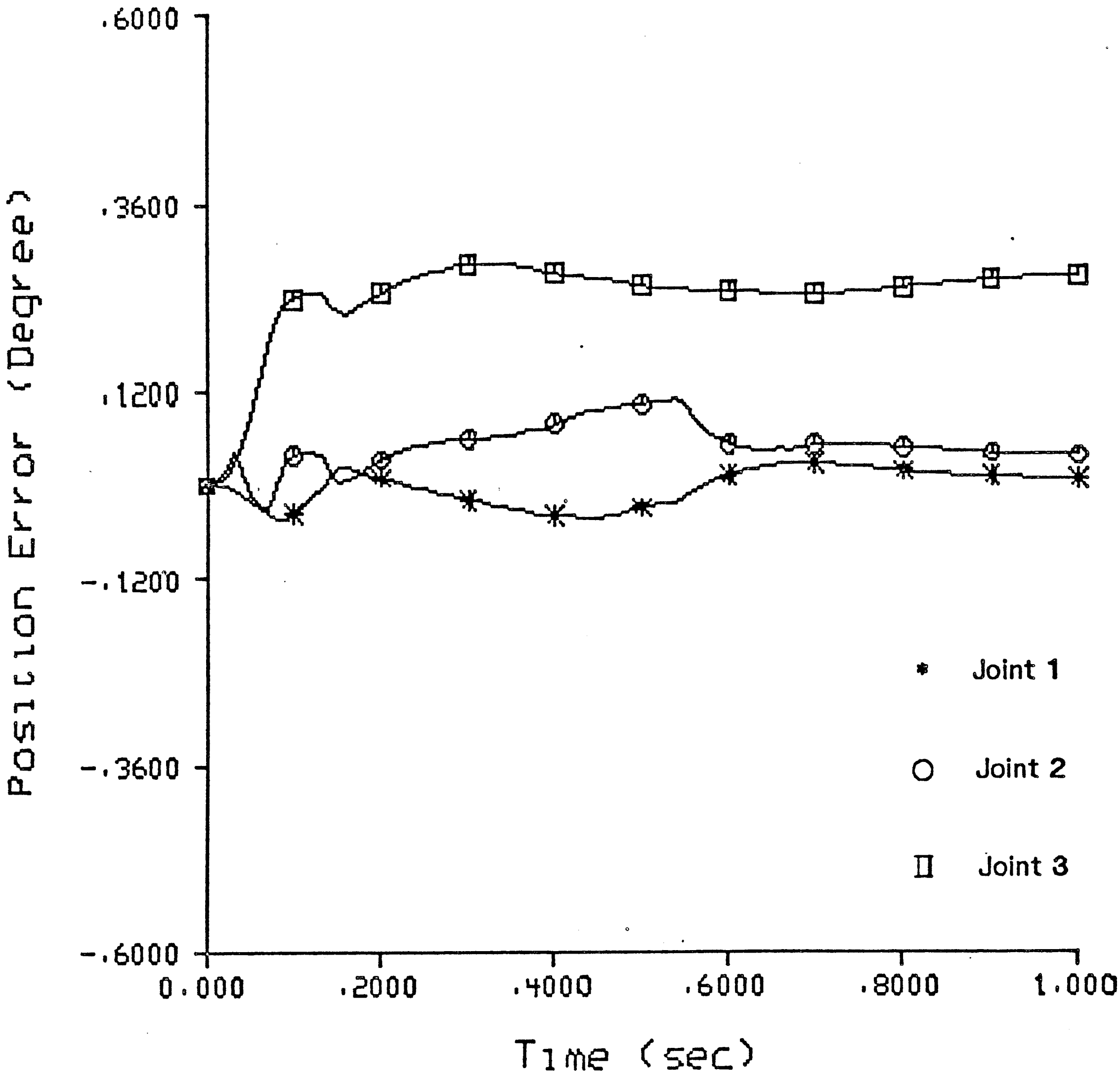

Figure 10 Adaptive Controller: Case (b)

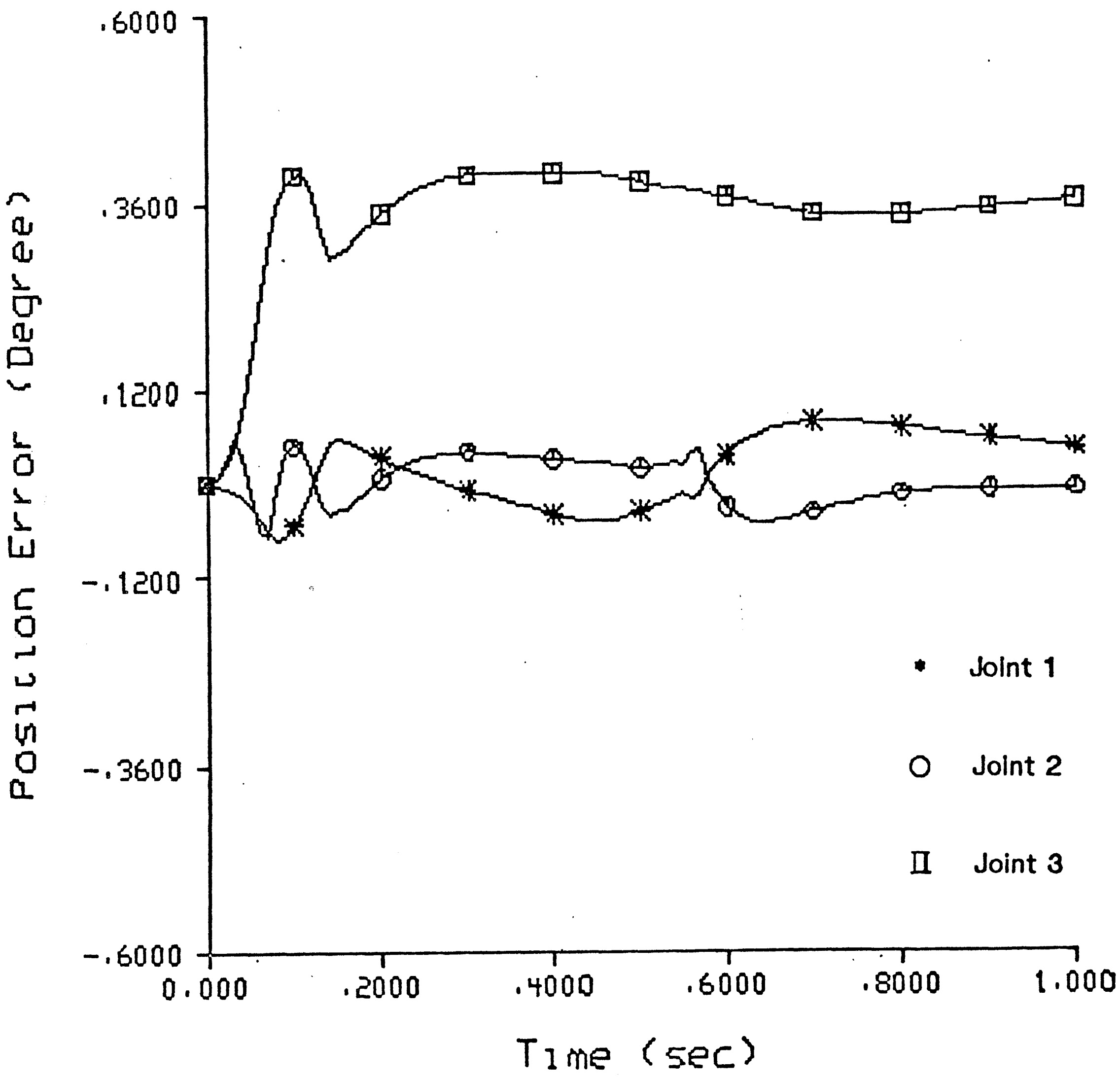

Figure 11 Adaptive Controller: Case (c)

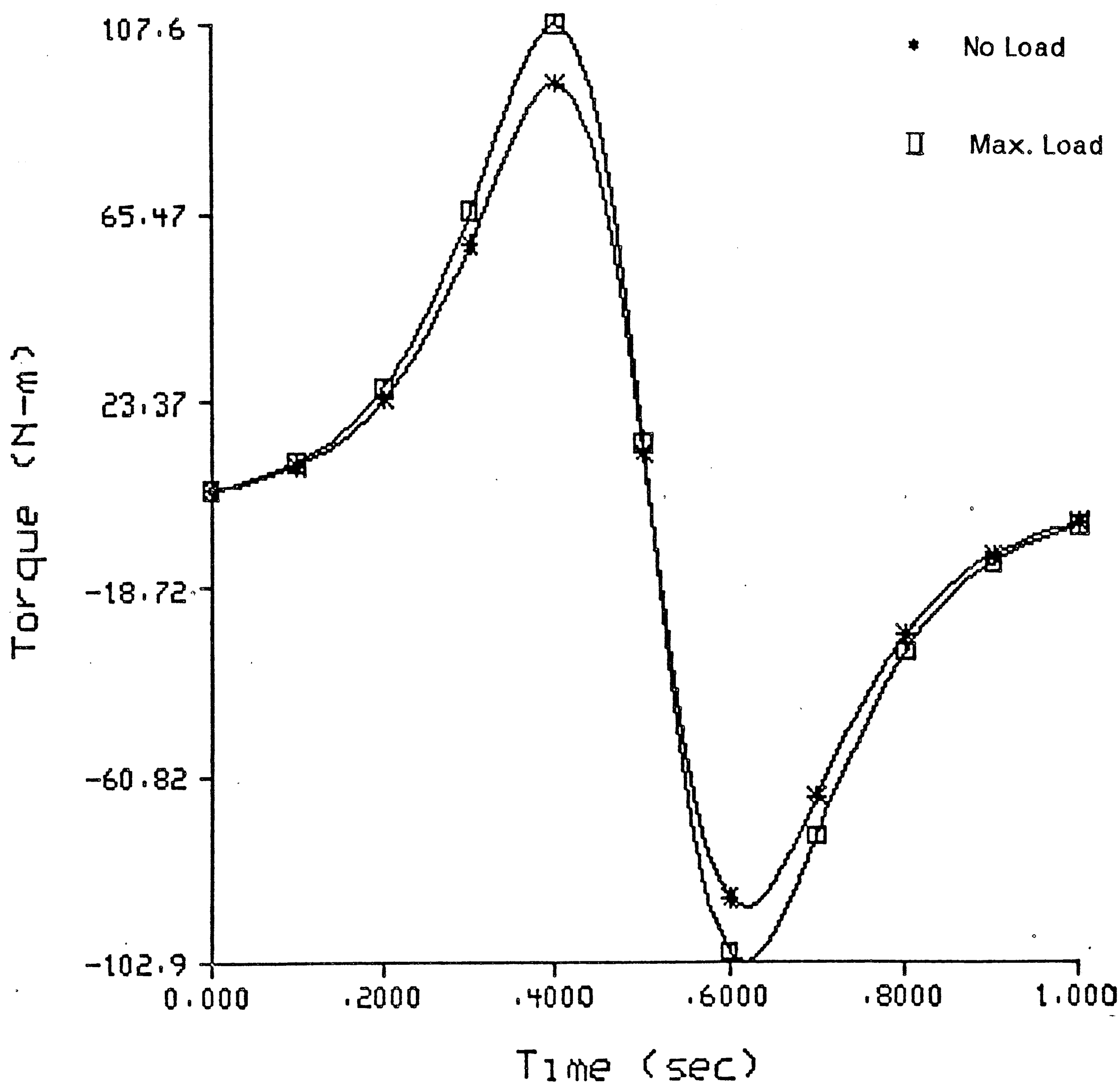

Figure 12 Variation of the Applied Torque (Joint 1)

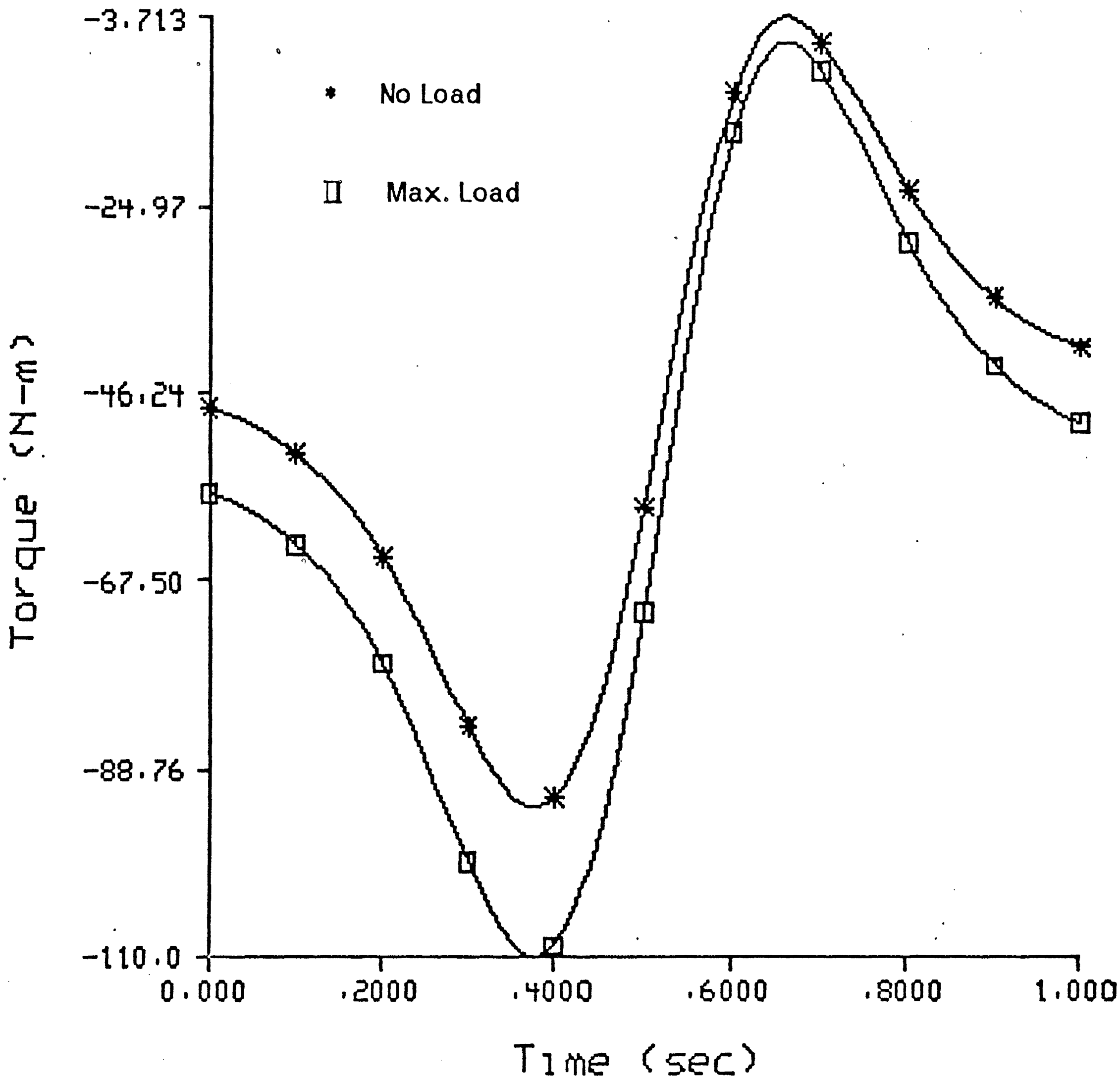

Figure 13 Variation of the Applied Torque (Joint 2)

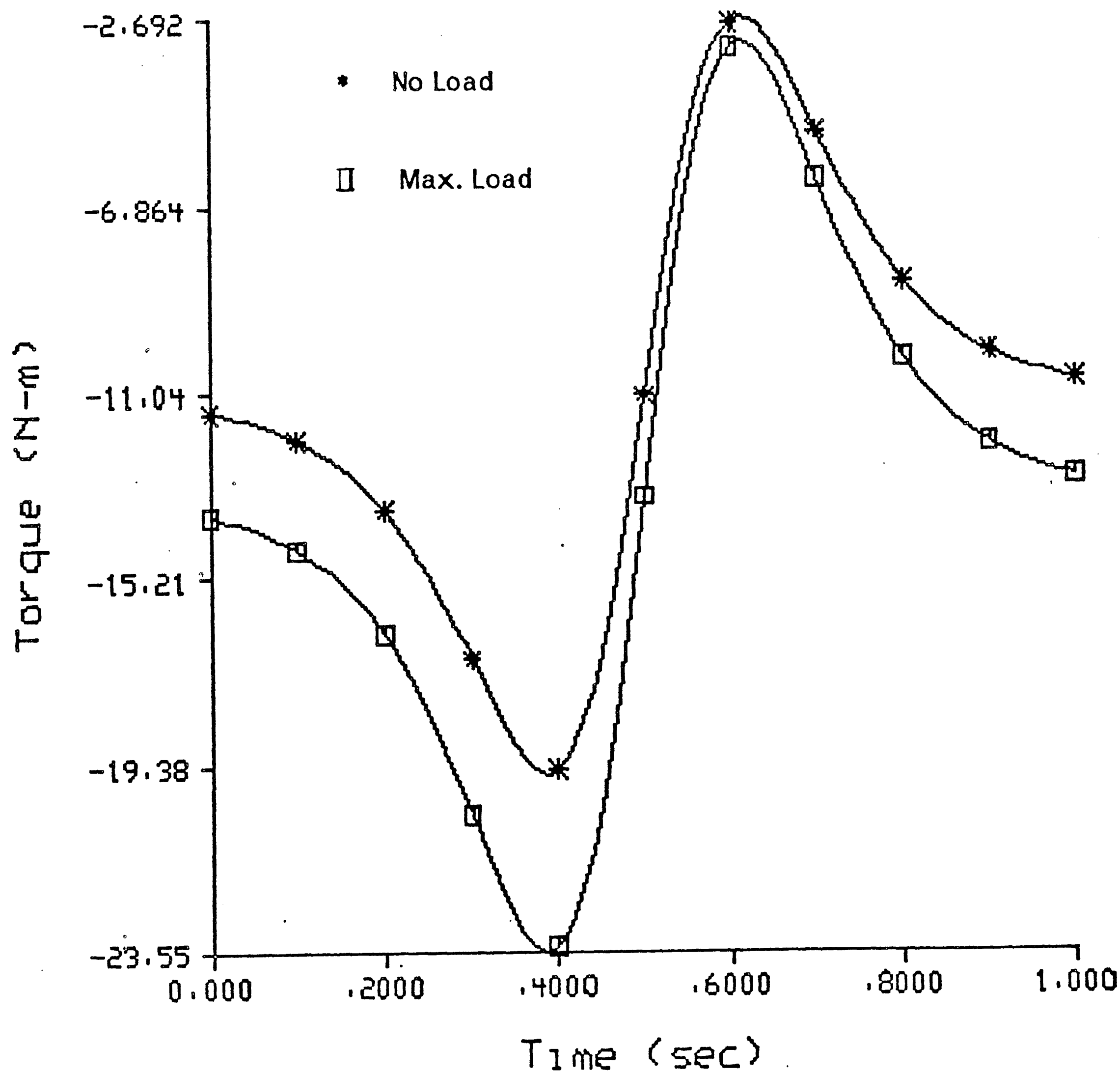

Figure 14 Variation of the Applied Torque (Joint 3)

5. References

1. Whitney, D. E., "Resolved Motion Rate Control of Manipulators and Human Prostheses," IEEE Transactions on Man-Machine System, Vol. MMS-10, No. 2, June 1969, pp 47-53.

2. Kahn, M. E., Roth, B., "The Near-Minimum-Time Control of Open-Loop Articulated Kinematic Chains," Transaction of the ASME, Journal of Dynamic Systems, Measurement and Control, Vol. 93, Sep. 1971, pp 164-172.

3. Paul, R., "Modeling, Trajectory Calculation and Servoing of a Computer Controlled Arm," Stanford Artificial Intelligence Laboratory, A.I. Memo 177, Sep. 1972.

4. Bejczy, A. K., "Robot Arm Dynamics and Control," Technical Memorandum 33-669, Jet Propulsion Laboratory, Feb. 1974.

5. Albus, J. S., "A New Approach to Manipulator Control: The Cerebellar Model Articulation Controller (CMAC)," Transaction of the ASME, Journal of Dynamic Systems, Measurement and Control, Vol. 97, Sep. 1975, pp 220-227.

6. Hemani, H., Camana, P. C., "Nonlinear Feedback in Simple Locomotion System," IEEE Transactions on Automatic Control, Vol. AC-19, Dec. 1976, pp 855-860.

7. Saridis, G. N., Stephanou, H. E., "A Hierarchical Approach to the Control of a Prosthetic Arm," IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-7, No. 6, 1977, pp. 407-420.

8. Young, K. K. D., "Controller Design for a Manipulator Using Theory of Variable Structure Systems," IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-8, No. 2, Feb. 1978, pp. 101-109.

9. Saridis, G. N., Lee, C. S. G., "An Approximation Theory of Optimal Control for Trainable Manipulators," IEEE Transactions on Systems, Man and Cybernetics,

Vol. SMC-9, no. 3, March 1979, pp 152-159.

10. Dubowsky, S., DesForges, D. T., "The Application of Model Referenced Adaptive Control to Robotic Manipulators," Transaction of the ASME, Journal of Dynamic Systems, Measurement and Control, Vol. 101, Sep. 1979, pp 193- 200.

11. Luh, J. Y. S., Walker, M. W., Paul, R., "Resolved-Acceleration Control of Mechanical Manipulators", IEEE Transactions on Automatic Control, Vol. AC-25, No. 3, June 1980, pp 468-474.

12. Horowitz, R., Tomizuku, M., "An Adaptive Control Scheme for Mechanical Manipulator - Compensation of Nonlinearity and Decoupling Control," The Dynamic systems and Control Division Paper of the ASME Winter Annual Meeting at Chicago, Nov. 16-21, 1980.

13. Turney, J. L., Mudge, T. N, Lee, C. S. G., "Equivalence of Two Formulation for Robot Arm Dynamics," SEL Report No. 142, Dec. 1980.

14. Koivo, A. J., Guo, T. H., "Control of Robotic Manipulator with Adaptive Controller," The 20th IEEE Conference on Decision and Control, San Diego, CA., Dec. 16-18 1981, pp. 271-276.

15. Lee, C. S. G., Chung, M. J., Turney, J. L., Mudge, T. N., "On the Control of Mechanical Manipulators," Proceedings of the Sixth IFAC Conference in Estimation and Parameter Identification, Washington D.C., June 1982, pp. 1454-1459.

16. Luh, J. Y. S., Walker, M. W., Paul, R., "On-line Computational Schemes for Mechanical Manipulators," Transaction of the ASME, Journal of Dynamic systems, Measurement and Control, Vol. 102, June 1980, pp 67-76.

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

18. Landau, Y. D., "Adaptive Control - the Model Reference Approach," Marcel-Dekker, Inc, 1979.

19. Isermann, R., (Ed.) "Identification and System Parameter Estimation," Proceedings of the Fifth IFAC Symposium, Damstadt, Federal Republic of Germany, Vol. 1, Sep. 24-28 1979.

20. Lee, C. S. G., Lee, B. H., Nigam, R., " An Efficient Formulation of Robot Arm Dynamics for Control and Computer Simulation," CRIM Technical Report RSD-TR-8-82, The University of Michigan, August, 1982.

21. Paul, R. P., "Robot Manipulators: Mathematics, Programs and Control," M. I. T. Press, 1981.

22. Lee, C. S. G., "Fundamental of Robotics," to be published by Addison-Wiley Pub. Co..