ROBUST TUNING OF ROBOT CONTROL SYSTEMS
(Grant # NAG 5 1550)

FINAL REPORT

I. Minis (P.I.), M. Uebel

Department of Mechanical Engineering
University of Maryland, Baltimore County Campus
Baltimore, MD 21228

Sponsored by: The Robotics Branch, NASA GSFC
Funding Period: May 1, 1991 - April 30, 1992

ABSTRACT

This study examines the computed torque control problem for a robot arm with flexible, geared, joint drive systems which are typical in many industrial robots. The standard computed torque algorithm is not directly applicable to this class of manipulators because of the dynamics introduced by the joint drive system. The proposed approach to computed torque control combines a computed torque algorithm with torque controllers at each joint. Three such control schemes are proposed. The first scheme utilizes the joint torque control system currently implemented on the robot arm and a novel form of the computed torque algorithm. The other two employ the standard computed torque algorithm and a novel model following torque control system based on model following techniques. Standard tasks and performance indices are used to evaluate the performance of the controllers. Both numerical simulations and experiments are used in the evaluation. The study shows that all three proposed systems lead to improved tracking performance over a conventional PD controller.

Table of Contents

List of Tables

List of Figures

Figure Page

Figure Page

Nomenclature

a	Observer pole
A	Denominator of plant transfer function
A_o	Observer polynomial
ADC	Analog-to-digital converter
$A_m(s)$ or $A_m(z)$	Denominator of model transfer function
B	Numerator of plant transfer function
B_{eq}	Equivalent damping
B_f	Material damping of flexible element
B_l	Viscous damping from link support bearings
B_{leq}	Equivalent link damping
B_m	Viscous damping from motor support bearings
$B_m(s)$ or $B_m(z)$	Numerator of model transfer function
B_{meq}	Equivalent motor damping
B^+	Stable plant zeros
B^-	Unstable or marginally stable plant zeros
CO	Cornering overshoot
DAC	Digital-to-analog converter
Δ	Uncertainty transfer function
ΔJ_l	Uncertainty in link inertia
ΔJ_m	Uncertainty in motor rotor and transmission inertia
e	Vector of joint errors (Θ_d-Θ_a)
$\dot{e}$	Vector of joint angular velocity errors
$\ddot{e}$	Vector of joint angular acceleration errors
G_c	Torque compensator transfer function
G_τ	Diagonal differential operator matrix
h_c	Angle and torque command update period
h_τ	Discrete torque servo update period
I	Identity matrix
J_{eq}	Equivalent inertia
J_l	Link inertia
J_{leq}	Equivalent link inertia

J_m	Motor rotor and transmission inertia
J_{meq}	Equivalent motor inertia
K_f	Spring constant of flexible element
K_c	Torque compensator gain
K_d	Diagonal derivative gain matrix
K_p	Diagonal proportional gain matrix
$\mathcal{L}_\infty(O)$	∞-Norm of orientation error
$\mathcal{L}_\infty(T)$	∞-Norm of translational error
$\mathcal{L}_2(O)$	2-Norm of orientation error
$\mathcal{L}_2(T)$	2-Norm of translational error
$M(\Theta_a)$ or M	Mass or inertia matrix
$\hat{M}(\Theta_a)$ or $\hat{M}$	Estimated mass or inertia matrix
N	Gear ratio
$Q(\Theta_a, \dot{\Theta}_a)$ or Q	Matrix of Coriolis, centrifugal, and friction torques
$\hat{Q}(\Theta_a, \dot{\Theta}_a)$ or $\hat{Q}$	Estimated matrix of Coriolis, centrifugal, and friction torques
R	Model following polynomial
R_1	Model following polynomial
S	Model following polynomial
T	Model following polynomial
T_1	Torque compensator numerator time constant
T_2	Torque compensator denominator time constant
Θ_a	Vector of actual joint angles
$\dot{\Theta}_a$	Vector of actual joint angular velocities
$\ddot{\Theta}_a$	Vector of actual joint angular accelerations
Θ_d	Vector of desired joint angles
$\dot{\Theta}_d$	Vector of desired joint angular velocities
$\ddot{\Theta}_d$	Vector of desired joint angle accelerations
Θ_g	Transmission gear angle
$\dot{\Theta}_g$	Transmission gear angular velocity
$\ddot{\Theta}_g$	Transmission gear angular acceleration

Θ_l	Link angle
$\dot{\Theta}_l$	Link angular velocity
$\ddot{\Theta}_l$	Link angular acceleration
Θ_m	Motor angle
$\dot{\Theta}_m$	Motor angular velocity
$\ddot{\Theta}_m$	Motor angular acceleration
τ_m	Torque applied to motor rotor
τ_c	Vector of torque commands
τ_d	Vector of torque commands
τ_s	Measured of sensed torque
τ_{app}	Net torque applied to link
τ_{dist}	Disturbance torque
ω_c	Vector of commanded joint angular velocities
ω_n	Natural frequency of model following torque loop
ζ	Damping of model following torque loop
ZOH	Zero-order-hold

Chapter 1: Introduction

A typical robotic manipulator consists of several links connected in series by revolute or prismatic joints. Due to their geometry, most robot manipulators are complex, nonlinear dynamical systems. Strong coupling exists between the dynamics of the robot joints, and it is most significant during high speed motions and periods of high acceleration. Gravity also plays a significant role in the robot dynamics by imposing configuration dependent disturbance loads. In addition, the dynamics of the robot can change dramatically with payload variations.

Motion control in most industrial robot arms is performed by simple Proportional-Derivative (PD) or Proportional-Integral-Derivative (PID) compensators that control each joint individually. Gravity compensation is accomplished, in many cases, by the addition of a constant torque command to the PD or PID control signal. These simple controllers are often tuned for a specific range of payloads, speeds, or configurations and may perform well for point to point tasks in which the connecting path is not critical. However, for applications in which payloads, speeds, and configurations vary in a wide range, or when precise trajectory following is essential, these simple control laws may not perform adequately.

Accurate trajectory following is a demanding task which often requires a more sophisticated control algorithm. Consequently, this area has been the focus of research efforts for many years. Most advanced control schemes for precise tracking incorporate some form of the inverse arm dynamics into the robot control system. The parameters of the dynamical equations may be based on analytical models of the arm or may be identified on-line through appropriate estimation schemes.

A promising approach to robot arm control is the "computed torque controller" which utilizes a complete dynamical model of the rigid link manipulator to compute the torques commanded to the joint motors given the desired trajectory. The drive system dynamical parameters, particularly the motor and gear inertias and the gear ratio, are typically incorporated into the model by assuming that the drive system components can also be treated as rigid bodies. Since the controller does not account for any non-rigid body dynamics, its performance is generally degraded by their presence.

A geared joint drive system can be a significant source of unmodeled dynamics which may result from flexibility, backlash, and friction. Consequently, the most successful applications of computed torque controllers have been with direct drive robot arms in which the drive system dynamics are easily modelled. Most industrial robots, however, do not use direct drives due to their higher payload-to-weight ratio requirements. These requirements can only be provided by geared joint drive systems. Computed torque control in this class of robot arms has been less successful.

The objective of this thesis is to investigate the successful implementation of computed torque control on a robot arm with flexible, geared, joint drive systems which are typical in many industrial robot arms. Improved performance of the computed torque schemes is targeted by introducing a joint torque servo loop.

Three alternative control schemes are proposed. Their performance is evaluated through simulation using a set of standard tasks and appropriate performance criteria. The most promising controller is then implemented and evaluated on an actual 7 degree-of-freedom robot manipulator.

This thesis is structured as follows. In Chapter 2, the standard computed torque controller is presented, several previous studies are reviewed, and the proposed approach is outlined. Chapter 3 describes the robot arm considered in this study and presents the analysis and synthesis of the proposed control algorithms. In Chapter 4, the performance evaluation procedures and the results obtained from the simulation tests are discussed. Chapter 5 presents the implementation of the most promising controller and the experimental results obtained from performance tests using the actual robot arm. Chapter 6 provides the conclusions of this study and proposes directions for further research.

Chapter 2: Computed Torque Control

The robot control problem is presented in this chapter. The rigid body dynamical equations of motion for a robot manipulator and the associated computed torque control equations are reviewed. The complexities associated with computed torque control are illustrated, and an overview of previous work in non-linear robot control is given. Finally, the proposed approach to implement computed torque control schemes in industrial robots is outlined.

2.1 Background

A robot manipulator is typically modeled as a series of rigid links, connected by prismatic or revolute joints. If a geared transmission is present between each driving motor and the corresponding link, its components are also modeled as rigid bodies. The corresponding dynamical equations of motion for such a manipulator can be written in the following form:

$$\tau_a = M(\Theta_a)\,\ddot{\Theta}_a + Q(\Theta_a, \dot{\Theta}_a) \qquad (2.1)$$

where τ_a is the vector of torques applied to the links, Θ_a is the vector of joint angles, $M(\Theta_a)$ is the matrix of link inertia terms, and $Q(\Theta_a, \dot{\Theta}_a)$ is the vector of Coriolis, centrifugal, gravity, and friction torques. Although revolute joints are assumed in this work, the above equation also holds for prismatic joints. Several methods to obtain the rigid link robot dynamics have been developed [Uicker, 1965; Armstrong, 1979; Luh, 1980a; Paul, 1981].

Equation (2.1) represents a set of nonlinear, coupled, differential equations for the joint motions. The end-point trajectory following problem consists of finding the necessary torque commands for the joint motors, $\tau_c(t)$, such that the end-effector of the robot follows a prescribed path in Cartesian coordinates, $X_d(t)$. However, the intrinsic non-linearities and cross-coupling of joint motions present significant difficulties to this control problem, which has been the focus of much research in the last twenty years.

There exist two general approaches to compute the torque commands for trajectory following which differ in the error signal of the control algorithm. The first method computes the torque commands based on the position error expressed in Cartesian coordinates [Luh, 1980b; Tarn, 1990]. The second and most common method is to first compute a joint-space trajectory, $\Theta_d(t)$, using the desired Cartesian-space trajectory, $X_d(t)$, and then compute the appropriate torque commands to minimize the joint position errors [Craig, 1986].

2.1.1 The Fundamental Computed Torque Algorithm

A multitude of non-linear control laws which compute torque commands based on joint positional errors have been developed for robot manipulators [Craig, 1986, 1987; Asada, 1986; Fu, 1987; Slotine, 1987]. The principle objective of many of these control laws is to linearize and decouple the dynamical equations of motion so that each joint can be considered independently using linear control theory. This concept is utilized by several "computed torque controllers" [Paul, 1972; Craig, 1986; Tourassis, 1985] in which the joint torques are typically evaluated from:

$$\tau_c = \hat{M}(\Theta_a)\left\{\ddot{\Theta}_d + K_d(\dot{\Theta}_d - \dot{\Theta}_a) + K_p(\Theta_d - \Theta_a)\right\} + \hat{Q}(\Theta_a, \dot{\Theta}_a) \quad (2.2)$$

where τ_c is the vector of torque commands, Θ_a is the vector of actual joint angles, Θ_d is the vector of desired joint angles, $\hat{M}(\Theta_a)$ is the estimated matrix of inertia terms, K_d is a matrix of joint angular velocity gains, K_p is a matrix of joint angle gains, and $\hat{Q}(\Theta_a, \dot{\Theta}_a)$ is the estimated vector of Coriolis, centrifugal, gravity, and friction torques.

The computed torque controller given by Eq. (2.2) is similar to the robot dynamical equations of motion, Eq. (2.1). The error terms added to the desired acceleration command, $\ddot{\Theta}_d$, have been introduced in order to compensate for the joint angle and angular velocity errors. Under the assumption of a perfect model, i.e. $\hat{M} = M$ and $\hat{Q} = Q$[†], and using $\tau_c = \tau_a$, Eq. (2.2) may be substituted into Eq. (2.1) to yield:

$$\ddot{e} + K_d\,\dot{e} + K_p\,e = 0 \quad (2.3)$$

where: $e = (\Theta_d - \Theta_a)$

[†] For notational convenience, the dependence of the terms M, $\hat{M}$, Q, and $\hat{Q}$ on the joint angles and angular velocities will be omitted when possible.

If the matrices K_d and K_p are diagonal, Eq. (2.3) represents a set of linear, decoupled error equations. In this ideal case, the errors converge to zero if all gains are positive. Furthermore, the error dynamics may be shaped by appropriately selecting these gains. However, Eq. (2.3) cannot be realized in practice due to two principal reasons: the lack of accurate parameter estimates, and the presence of unmodeled dynamics, which are discussed below. Sweet and Good [1985] discuss several additional difficulties which may arise in robot motion control.

2.1.2 Complexities in Computed Torque Control

Inaccurate Parameter Estimates

The parameters of the robot dynamical model are estimated either analytically or experimentally. Given the complexity of the system, both approaches may result in inaccurate parameter values. For example, the viscous and Coulomb friction coefficients may change with temperature or configuration and their estimates are, therefore, inaccurate. Typically the estimates $\hat{M}$ and $\hat{Q}$ are not equal to the terms M and Q of the actual robot arm, and therefore, the error equation becomes:

$$\ddot{e} + K_d \dot{e} + K_p e = \hat{M}^{-1} (M - \hat{M}) \ddot{\Theta}_a + \hat{M}^{-1} (Q - \hat{Q}) \qquad (2.4)$$

In this case, the joint error dynamics remain non-linear and coupled. The extent of non-linearity and coupling depends on the deviation of the model from the actual dynamics. Equation (2.4) indicates that the tracking error dynamics will deviate from those described by Eq. (2.3).

Unmodeled Dynamics

Degraded performance of the computed torque controller may also arise from unmodeled dynamics in the robot arm. For example, in deriving the error dynamics given by Eq. (2.3), it is assumed that the motor output torque accurately tracks the commanded torque. This assumption is reasonable for those robot arms which incorporate a servo loop to control the current in the motor windings, as in Khosla [1989]. In the absence of such a current loop, controlling the motor output torque may be a difficult task. Furthermore, even if the motor torque accurately tracks the commanded torque, other unmodeled dynamics, such as the flexibility of the drive system, may effect the torque applied to the link itself.

The deviation of the applied torque, τ_a, from the commanded torque, τ_c, due to the unmodeled dynamics may be represented as:

$$\tau_a = G_\tau \tau_c \tag{2.5}$$

where G_τ is a diagonal matrix of time domain differential operators. In this case, even under the assumption of perfect $\hat{M}$ and $\hat{Q}$ estimates, the error equation becomes:

$$\ddot{e} + K_d \dot{e} + K_p e = (M^{-1} G_\tau^{-1} M - I) \ddot{\Theta}_a + M^{-1} (G_\tau^{-1} - I) Q \tag{2.6}$$

In this case, neither K_p nor K_d can compensate for the matrix G_τ, and clearly, the dynamics of the individual joints are coupled.

2.2 Related Research

The most successful applications of computed torque control have been with direct drive robot arms which incorporate motor current servo loops. Direct drive systems may be accurately modeled and controlled due to several distinct characteristics. First, since the motor rotor is directly coupled to the corresponding link, direct drive arms are free of backlash and much of the friction which accompanies geared drive systems. Furthermore, by utilizing a high-gain, motor-current servo loop, the back emf effects in the motor are compensated. Consequently, the drive system appears as a simple gain between the motor current command and the output torque [Sweet, 1985].

Direct drive arms have been constructed for research purposes at Carnegie Mellon University [Kanade, 1984; Schmitz, 1985], MIT [An, 1988a,b], and Yale University [Buhler, 1990; Levin, 1989]. Computed torque schemes have been implemented on each of these arms and have shown superior performance with respect to conventional PD and PID controllers [Whitcomb, 1991; An, 1988b; Khosla, 1989].

Using direct drive arms, several researchers have investigated the effects of incomplete dynamical models and inaccurate parameter estimates on the performance of computed torque controllers. Whitcomb [1991] investigated arm performance in the presence of known and unknown payloads. He found that the root mean square of the tracking error ($\mathcal{L}_2$-norm of tracking error) degraded by approximately 50% when the payload inertia was not included in the model. Khosla [1988] excluded the Coriolis and centrifugal terms from the dynamical equations and utilized only the diagonal inertia matrix terms in the computed torque controller. His results indicated that neglecting the Coriolis and centrifugal terms introduces significant trajectory tracking errors even at low joint velocities of approximately 1 rad/sec. Deterioration in performance was also shown when the inertia matrix was simplified to include only diagonal terms. Asada [1983] studied the system performance as the inertia, damping, and gravity terms were removed from the computed torque controller, to finally yield a simple PD controller. His results showed a continuous decline in performance as each of these terms was omitted. These and other experiments have shown that the inertia, Coriolis, centrifugal, gravity, and friction terms may all have significant contributions to the robot arm dynamics given by Eq. (2.1). Furthermore, the impact of any single term depends on the robot arm design and the task it is performing.

Although precise control of the torque applied to each joint allows computed torque controllers to be successfully implemented on direct drive arms, most industrial robots utilize geared drive systems which complicate their dynamics. Gear backlash and flexibility are usually not modeled in the robot dynamical equations, but may have significant effects in the arm dynamics. In particular, highly undesirable drive system dynamics result from the flexibility in harmonic drives, which have become popular transmission systems for robot arms due to their compactness and high gear ratios. The flexible element of the harmonic drive, the flexspline, leads to underdamped, low frequency oscillations of the drive system [Karlen 1990]. The drive system dynamics degrade the performance of the computed torque controller and have been the principle reason that most applications of computed torque controllers on industrial robot arms have not lead to improved performance.

Leahy [1989] found this to be the case when he implemented a computed torque scheme on a PUMA-600 industrial robot arm. His experiments showed that the actuator plays a significant role in the dynamics of the entire manipulator. He also concluded that even though tracking performance is degraded by the drive system dynamics, the robustness of the computed torque controller is enhanced by the high gear ratio. However, he found the overall performance of the computed torque controller to be unacceptable.

Several researchers have investigated control schemes for trajectory following with robot arms which contain flexible joints. Spong [1987] showed that feedback linearization based on inverse plant dynamics, an algorithm similar to the computed torque controller, can be used to control such a manipulator. However, in this case, the joint acceleration and jerk must be available for feedback to ensure robustness with respect to modeling uncertainties. DeLuca [1988] utilized non-linear feedback to achieve both joint decoupling and linearization. However, his method required full state feedback, i.e., it utilized both the motor and joint positions and velocities. Ghorbel [1989] presented an adaptive algorithm for control of flexible joint manipulators which eliminates the need for jerk and acceleration feedback, but still required the full measurement of the motor and joint positions and velocities.

In an effort to reduce the dynamical effects of the drive system, several researchers and at least one robot manufacturer [Karlen, 1990] have implemented joint torque control through direct feedback of the torque measured between the transmission and the link. Luh [1983] was the first to implement a joint torque controller on two joints of the Stanford Arm and achieved a 95% reduction in effective frictional torques of the joints. A similar torque loop was implemented on a PUMA 500 Arm by Pfeffer [1989]. His results also showed significant reductions (97%) in effective frictional torque and substantial improvement in fine motion control. For robot arms with harmonic drives, the torque loop is necessary to limit the low frequency oscillations of the open loop drive system, as previously discussed.

2.3 Proposed Approach to Computed Torque

Previous research in computed torque control has shown that: i) Computed torque algorithms have the potential to provide superior performance over conventional PD or PID controllers, although their performance is directly related to the accuracy and completeness of the robot dynamical model; ii) Direct drive robot arms can be accurately described by the rigid body model of Eq. (2.1) and, therefore, are amenable to computed torque control; iii) Geared drive systems, common to industrial robots, introduce complex dynamics, in particular, flexibility in the robot joints. Most industrial robots do not use direct drives since they require high payload-to-weight ratios offered only by geared drive systems. Consequently, there has been little success in implementing computed torque controllers on this class of manipulators.

This thesis examines the problem of computed torque control for geared, flexible joint manipulators with joint torque servo loops. Three novel motion control systems are proposed, each consisting of two parts: i) a computed torque algorithm which provides the torque commands to all joints based on a desired joint-space trajectory and a dynamical model of the arm; ii) a torque control system at each joint which regulates the actual torque output of the drive system. The first control scheme utilizes the joint torque controller currently implemented on the robot arm considered and a novel form of the computed torque algorithm. The two others utilizes the standard computed torque algorithm and a novel model following torque control system based on model following techniques.

It should be emphasized that the present work investigates the computed torque control problem for a seven degree of freedom (7-DOF) robot arm, both theoretically and experimentally. Much of the previous work on computed torque controllers, as well as other non-linear control schemes, has been limited to 3-DOF robot arms. Since most industrial robots are 5, 6, or 7-DOF manipulators, this work offers significant insights on the application of these controllers in practical robot systems.

Chapter 3: Control System Analysis and Synthesis

The robot arm considered in this study and the existing PD motion control system are briefly described in this chapter. The open loop joint drive system is analyzed to illustrate the undesirable dynamics which result from its flexibility, thus necessitating the introduction of a torque control system. Subsequently, the analysis and design of the three proposed motion control systems are presented.

Each of the proposed motion control schemes consists of two levels: a computed torque algorithm which provides the torque commands to all joints, and a local torque control system at each joint which regulates the actual torque output of the drive system. The first scheme utilizes the existing torque control loop and a novel computed torque algorithm. The other two schemes utilize the standard computed torque algorithm, Eq (2.2), and a novel torque control system which is based on model following techniques. Two versions of the model following controller are presented, one continuous and the other discrete.

3.1 RRC 1607HP Robot Manipulator

The robot arm considered in this study is the Robotics Research Corporation (RRC) 1607HP Dexterous Manipulator, shown in Fig. 3.1, and described in detail in Karlen [1990]. It is a 7-DOF manipulator with revolute joints. The drive system of each joint, shown in Fig. 3.2, consists of a DC motor connected to the wave

ORIGINAL PAGE IS
OF POOR QUALITY

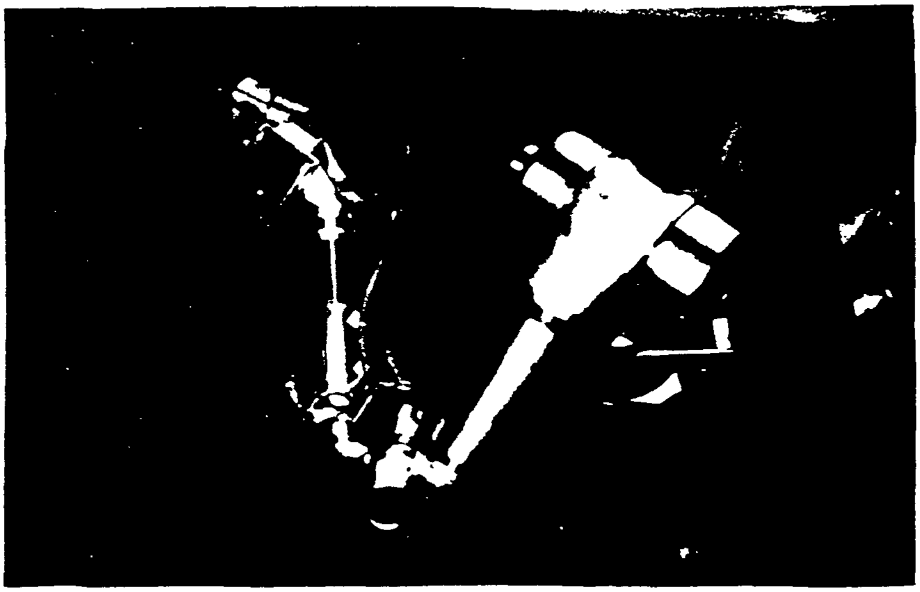

Figure 3.1: RRC 1607HP dexterous manipulator

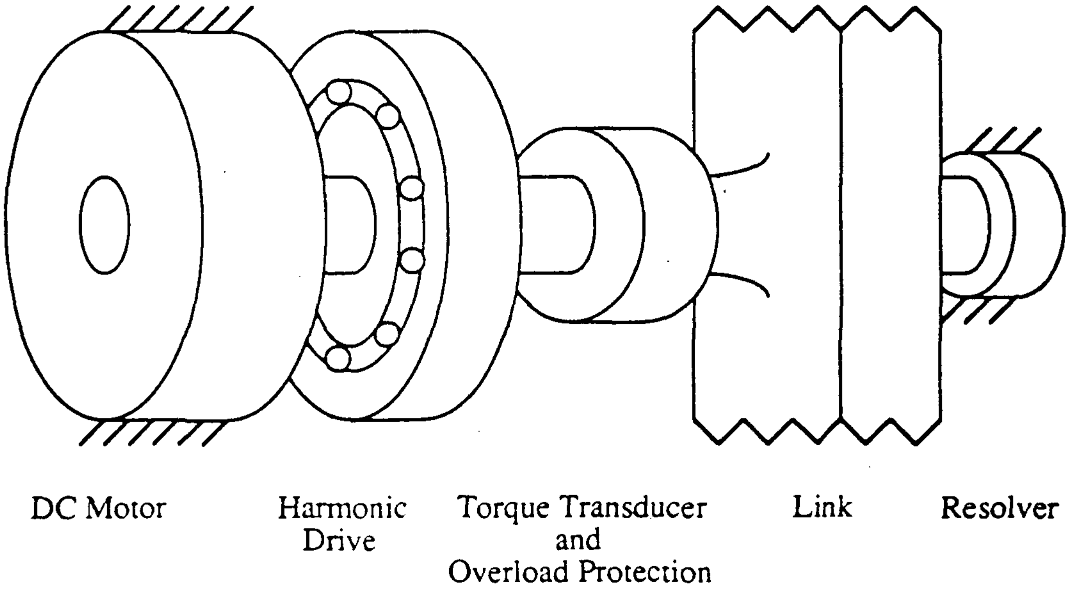

Figure 3.2: Joint drive system

generator of a harmonic drive. The flexspline of the harmonic drive is connected to the link structure through an overload protection device and torque transducer. The latter allows measurement of the torque applied to the link at a point just prior to the support bearings. Joint position and velocity measurements are provided by a resolver, mounted directly to the joint housing. It is emphasized that the resolver provides the joint, and not the motor, angle and angular velocity information. The dynamics of the joint drive system are analyzed in detail in the next section.

The existing control system of each joint, Fig. 3.3, functions as a PD motion controller and contains three analog servo loops: a motor current loop, a torque loop, and a joint velocity loop. A single joint position loop provides the velocity commands to all joints at discrete time periods, h_c, according to the equation:

$$\omega_c = K_p (\Theta_d - \Theta_a) \tag{3.1}$$

where ω_c is the vector of commanded joint angular velocities, and K_p is a diagonal matrix of proportional gains. The velocity commands are introduced to the analog hardware through a Digital-to-Analog convertor (DAC) and a zero-order-hold (ZOH). It is noted that the original control system of the RRC arm has been modified as described in Miller [1991] so that the parameters K_p and h_c can be specified by the user.

The velocity loop operates on the joint velocity errors and provides torque commands to the torque loop. The objective of the torque loop is to minimize the oscillations in the drive system which result from the compliance of the harmonic

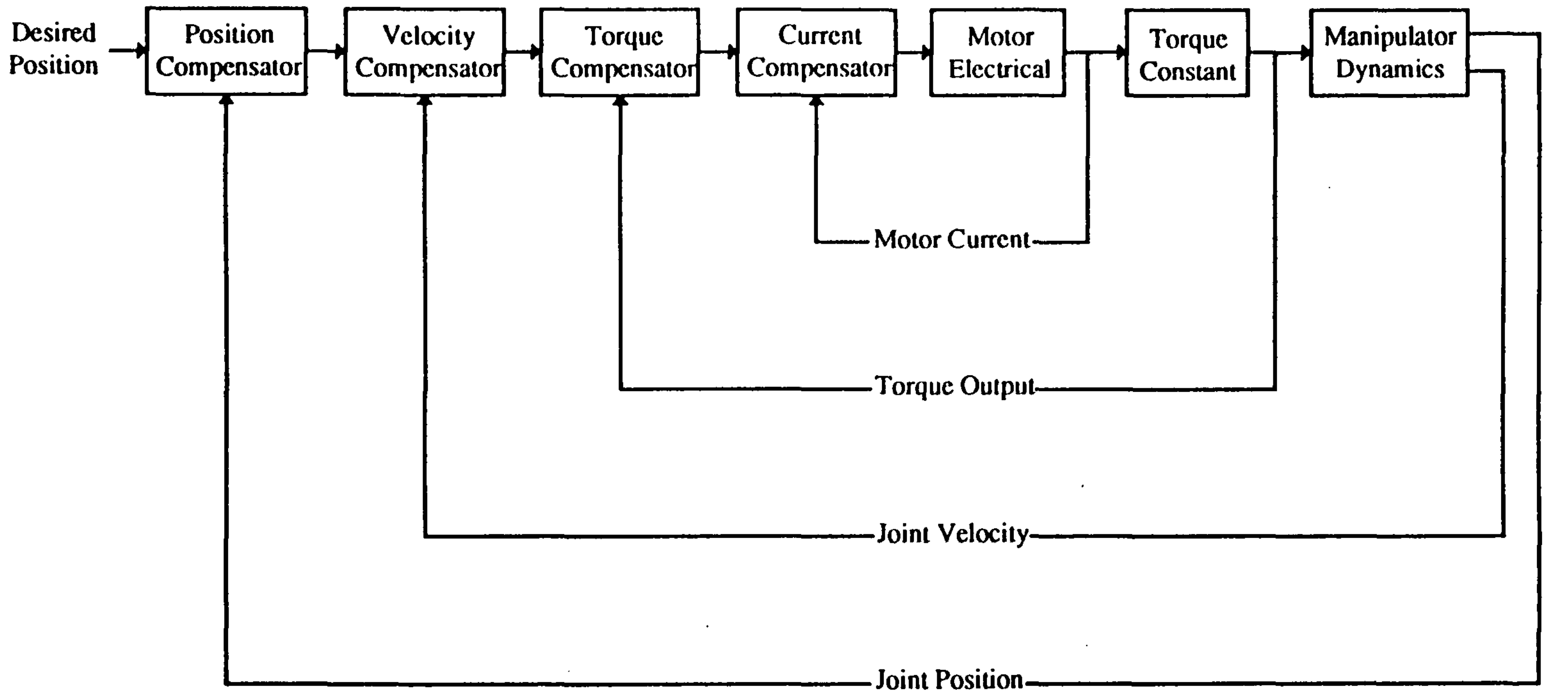

Figure 3.3: RRC control system configuration

drive (see Section 3.2) and to provide the appropriate current commands to the current loop. The current loop regulates the current in the motor windings and results in a closed loop system with a flat frequency response over a high bandwidth.[†] Since the motor torque is proportional to the motor current, the motor/current loop combination may be modeled as a simple gain between the input current command and the motor output torque (see Section 2.3). This gain is equal to the motor torque constant.

The modular architecture of this control system allows the use of external position, velocity, or torque command inputs. For the computed torque controllers, only torque inputs are required, thus the position and velocity loops are not used with these controllers. The PD motion control system is used only for comparison purposes.

3.2 Dynamics of the Joint Drive System

Figure 3.4 shows a schematic representation of the single joint drive system. The block diagram for this system is shown in Fig. 3.5. τ_m is the electro-magnetic torque, motor torque, applied to the motor rotor. J_m and B_m are the inertia and damping of the motor rotor, respectively. The harmonic drive is modeled as a pair of gears with ratio, N, coupled to a flexible element with spring constant, K_f, and

[†] Up to 1KHz as per discussion with Paul Eismann of Robotics Research Corporation.

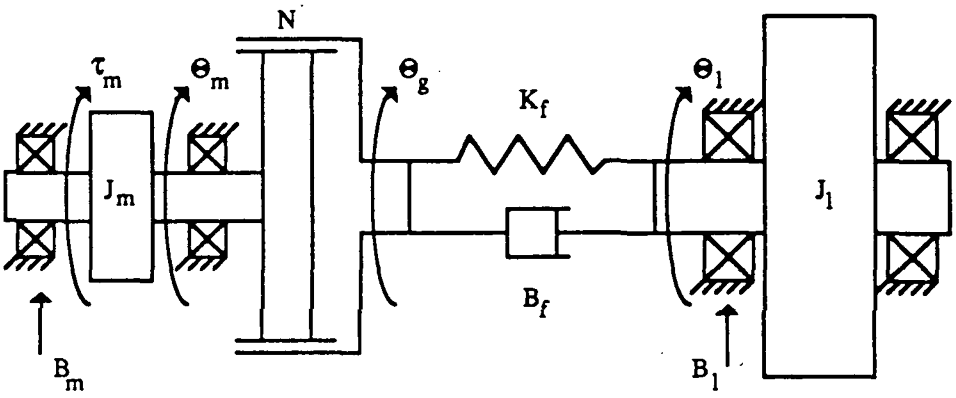

Figure 3.4: Schematic of the joint drive system

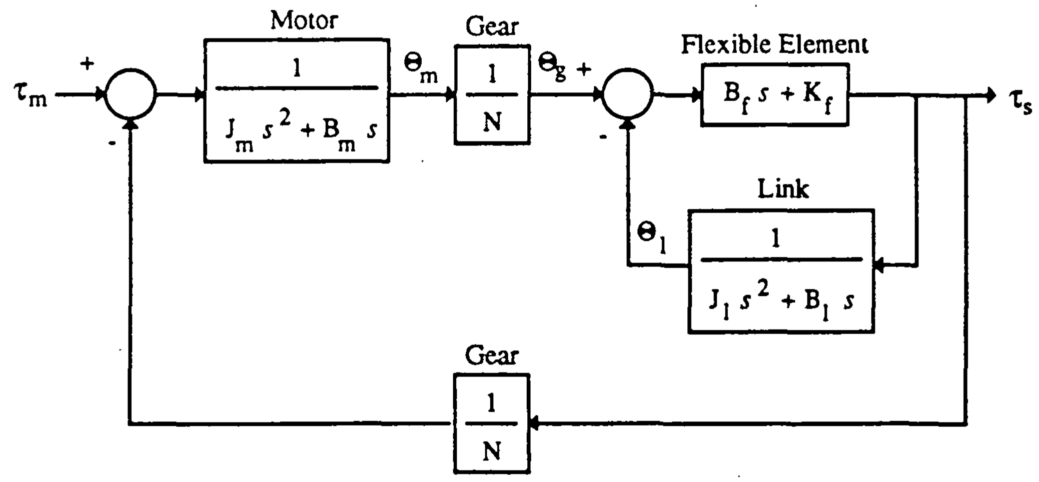

Figure 3.5: Block diagram joint drive system

material damping, B_f. B_l represents the damping at the link side of the flexible element. The motor and link damping terms, B_m and B_l, result from the viscous damping of the support bearings. J_l is the effective inertia at the link side of the flexible element. It represents the inertia of that portion of the robot arm which moves with the joint. Therefore, it includes the inertia of all the links between the joint under consideration and the end effector. J_l is equivalent to the corresponding diagonal element of the inertia matrix, M, of Eq. (2.1), when the motor rotor and transmission inertias are not considered.

If the backlash in the transmission is negligible, the drive system has two degrees of freedom which can be selected as the joint angle, Θ_l, and the gear angle, Θ_g, or equivalently, the motor angle, Θ_m, (see Fig. 3.4). The equations of motion of the joint drive system are:

$$J_l \ddot{\Theta}_l = K_f (\Theta_g - \Theta_l) + B_f (\dot{\Theta}_g - \dot{\Theta}_l) - B_l \dot{\Theta}_g \qquad (3.2)$$

$$J_m \ddot{\Theta}_m = - \frac{K_f}{N} (\Theta_g - \Theta_l) - \frac{B_f}{N} (\dot{\Theta}_g - \dot{\Theta}_l) - B_m \dot{\Theta}_m + \tau_m \qquad (3.3)$$

The gear angle and the motor angle are related by the gear ratio, N, i.e.:

$$\Theta_m = N \Theta_g \qquad (3.4)$$

Equation (3.3) can be written in terms of the joint angle and the gear angle only, using Eq. (3.4) to obtain:

$$N^2 J_m \ddot{\Theta}_g = - K_f (\Theta_g - \Theta_l) - B_f (\dot{\Theta}_g - \dot{\Theta}_l) - B_m N^2 \dot{\Theta}_g + N \tau_m \tag{3.5}$$

Taking the Laplace transform of Eqs. (3.2) and (3.5), and assuming zero initial conditions, the equations of motion can be written in matrix form as :

$$\begin{bmatrix} N^2 J_m s^2 + (N^2 B_m + B_f)s + K_f & - B_f s - K_f \\ - B_f s - K_f & J_l s^2 + (B_l + B_f)s + K_f \end{bmatrix} \begin{bmatrix} \Theta_g \\ \Theta_l \end{bmatrix} = \begin{bmatrix} N \tau_m \\ 0 \end{bmatrix} \tag{3.6}$$

This system is solved for Θ_g and Θ_l to obtain:

$$\frac{\Theta_g}{\tau_m} = \frac{N \{J_l s^2 + (B_l + B_f)s + K_f\}}{\{N^2 J_m s^2 + (N^2 B_m + B_f)s + K_f\} \{J_l s^2 + (B_l + B_f)s + K_f\} - (B_f s + K_f)^2} \tag{3.7}$$

and

$$\frac{\Theta_l}{\tau_m} = \frac{N (B_f s + K_f)}{\{N^2 J_m s^2 + (N^2 B_m + B_f)s + K_f\} \{J_l s^2 + (B_l + B_f)s + K_f\} - (B_f s + K_f)^2} \tag{3.8}$$

The torque transmitted through the harmonic drive is measured in the torque transducer with strain gauges. This sensed torque, τ_s, is equal to the sum of the torque applied to the link and the friction losses of the link support bearings. It is given by:

$$\tau_s = (B_f s + K_f)(\Theta_g - \Theta_l) \tag{3.9}$$

Using Eqs. (3.7) - (3.9), the transfer function between the motor torque, τ_m, and the sensed torque, τ_s, is obtained from:

$$\frac{\tau_s}{\tau_m} = \frac{N\, s\, (J_l s + B_l)\,(B_f s + K_f)}{\left\{N^2 J_m s^2 + (N^2 B_m + B_f) s + K_f\right\}\left\{J_l s^2 + (B_l + B_f) s + K_f\right\} - (B_f s + K_f)^2} \tag{3.10}$$

Equation (3.10) is critical in this study since it represents the dynamics of the joint drive system to be controlled by the joint torque controllers.

The zeros of the open loop transfer function given by Eq. (3.10) are located at 0, $-K_f/B_f$, and $-B_l/J_l$. Since the spring constant, K_f, is much greater than the material damping of the flexible element [Chen 1990], the zero at $-K_f/B_f$ will be far from the imaginary axis. Furthermore, since J_l varies with payload and configuration, the location of the zero given by $-B_l/J_l$ may also vary significantly at different operating conditions. For example, the ratio $-B_l/J_l$ for joint 1 of the robot arm under consideration varies from -0.06 to -0.6. These variations can be even more significant if different payloads are carried by the end effector.

One pole of the open loop drive system, Eq. (3.10), is located at the origin and, therefore, will be cancelled by the corresponding zero, resulting in a third order transfer function. The characteristic equation of this reduced transfer function can be written as:

$$1 + J_l \frac{N^2 s \left\{ J_m s^2 + (N^2 B_m + B_f)s + K_f \right\}}{N^2 \left\{ (B_l + B_f)s + K_f \right\} (J_m s + B_m) + B_l s \, (B_f s + K_f)} = 0 \tag{3.11}$$

Figure 3.6 shows a representative root locus of the drive system poles for varying J_l. If J_l increases, one pole approaches the origin while the other two poles approach the roots of (see Eq. (3.11)):

$$N^2 J_m s^2 + (N^2 B_m + B_f) s + K_f = 0 \tag{3.12}$$

Since the material damping, B_f, and the motor damping, B_m, are typically small, these roots are complex and located close to the imaginary axis. Typical values for joint 1 of the robot arm under consideration are -12 ± j 154.

Based on the observations of the pole and zero locations, it is apparent that for those joints with relatively high inertia, J_l, the time response will show oscillations due to the lightly damped pair of complex poles. Figure 3.7 shows the simulated step response of the drive system for joint 1 of the robot arm. The underdamped oscillations shown in this figure are typical for this type of drive system. Since they are unacceptable in most practical robots, an appropriate controller must be used to improve these dynamics. For this purpose, a joint torque servo, or torque loop, is presently used in the RRC arm and is discussed in the next section. An alternative approach to torque control based on model following techniques is also presented in that section.

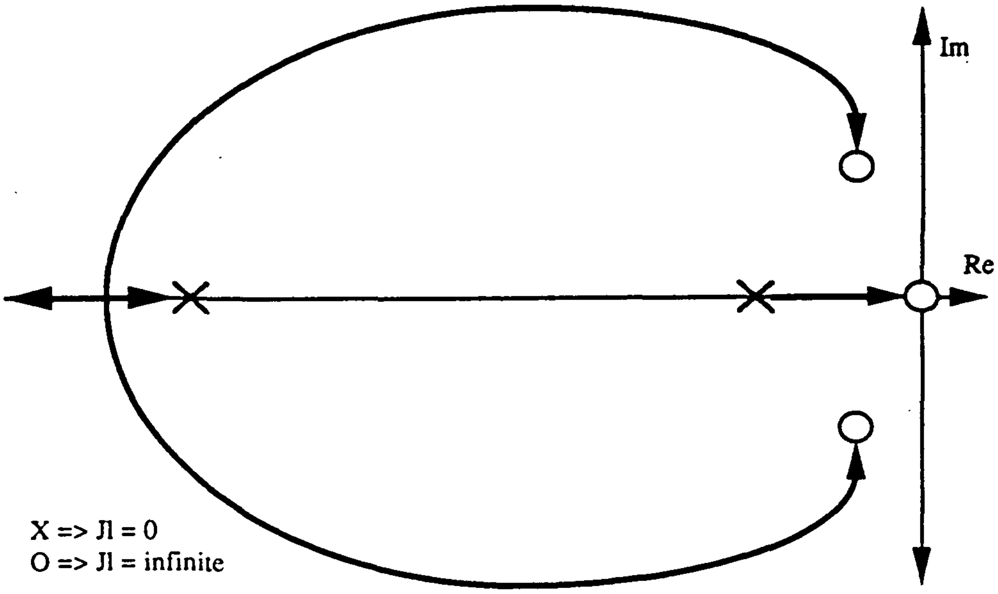

Figure 3.6: Root locus of the joint drive system for varying link Inertia

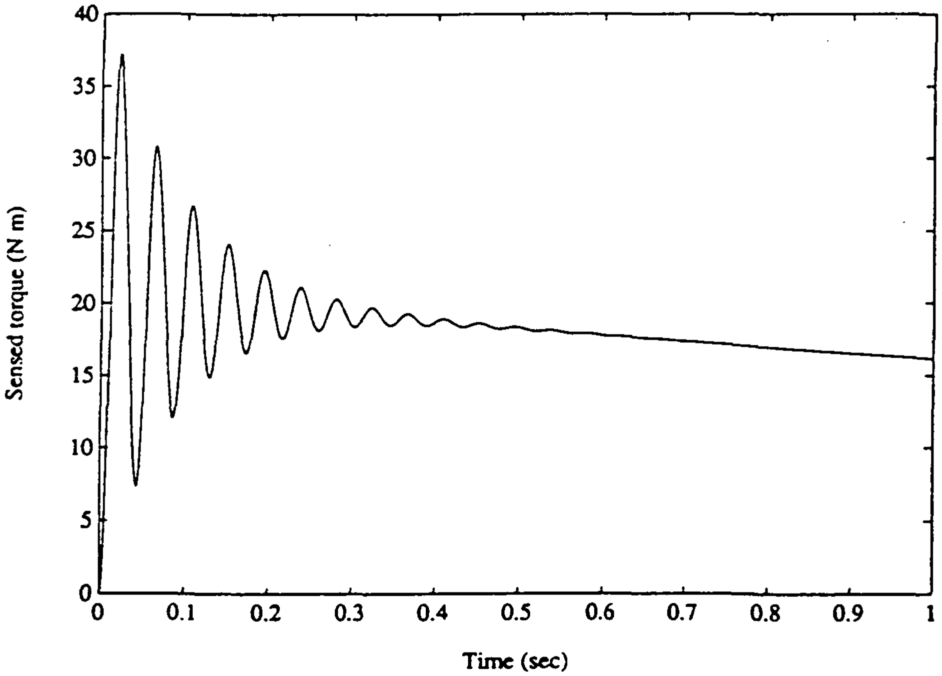

Figure 3.7: Step response of the joint drive system

3.3 Joint Torque Control

3.3.1 RRC Joint Torque Control

The existing RRC control system uses the torque servo loop shown in Fig. 3.8 to improve the dynamics of the joint drive system. τ_s is the torque measured by the torque transducer located just after the harmonic drive. τ_c is the torque commanded to the torque loop. Based on the torque error, $\tau_c - \tau_s$, the torque compensator of the actual system provides a current command signal to the current loop, as described in Section 3.1. For this model, the dynamics of both the motor and current loop are lumped into the torque compensator transfer function, and therefore, the output of the torque compensator is the motor torque, τ_m. As mentioned in Section 3.1, the bandwidth of the motor/current loop combination is significantly higher than the bandwidth of the torque loop, and therefore, it can be modelled as a simple gain, equal to the torque constant of the motor.

The RRC torque controller is similar to the one described in Luh [1983], and it incorporates a phase lead compensator to add damping to the open loop drive system. The general form of the lead compensator is:

$$Gc = \frac{K_c\,(T_1\,s + 1)}{(T_2\,s + 1)} \tag{3.13}$$

where: $T_1 > T_2$

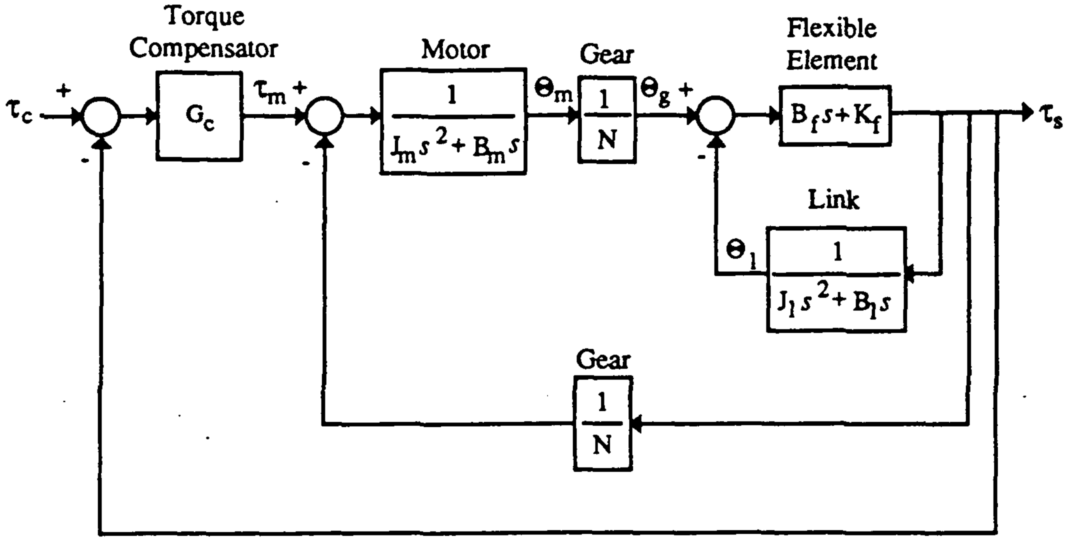

Figure 3.8: Block diagram of the RRC torque loop

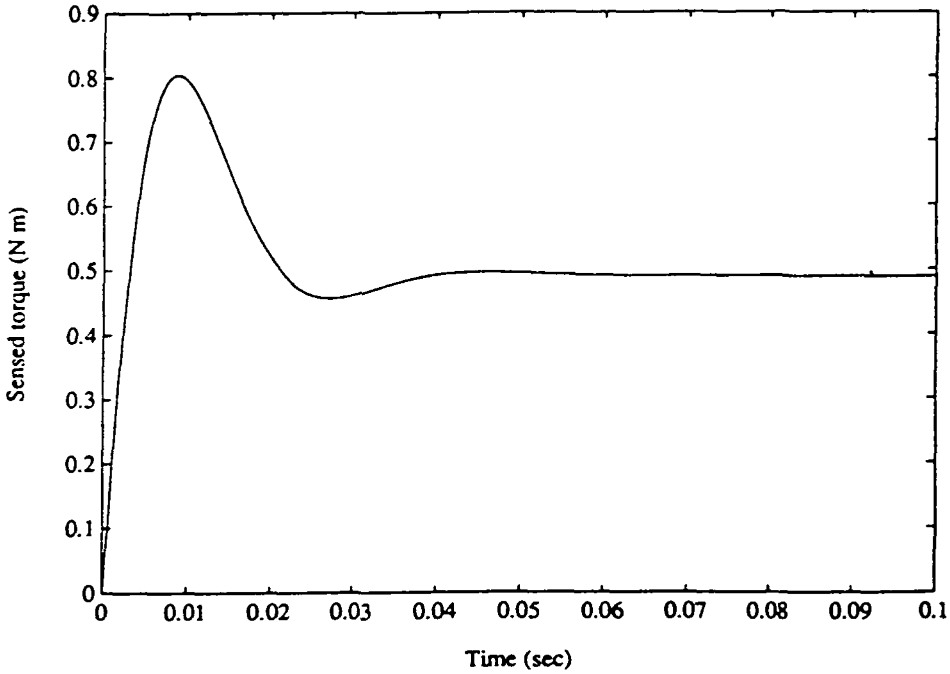

Figure 3.9: Step Response of the RRC torque loop

If the time constants T_1 and T_2 are chosen properly, the lead compensator will increase both the phase margin of the closed loop system and its damping.† Figure 3.9 shows the simulated step response of the RRC torque loop for joint 1. A comparison between Figs. 3.7 and 3.9 indicates that the RRC torque loop significantly improves the drive system dynamics.

Since the dynamics of the drive system and, therefore, the dynamics of the torque loop depend on the effective link inertia, which in turn changes with configuration, it is of interest to examine the robustness of the RRC torque loop for various link inertias. The Routh-Hurwitz stability criterion was used for this purpose and it was found that the closed loop system is stable for all values of effective link inertia, J_l. To obtain indications on robust performance, simulations were performed with several possible inertia values. Figure 3.10 shows the simulated step response of the RRC torque loop for joint 1 under maximum, minimum, and median values of J_l. It clearly illustrates that the performance of the torque loop is highly dependent on the effective link inertia, and therefore, on configuration. Similar results are indicated by Fig. 3.11 which shows the simulated step response of the torque loop for joint 7. Since the effective link inertia of this joint does not change with configuration, the joint responses with no payload, the maximum rated payload, and a median payload are shown. This figure illustrates that large differences in performance may be observed in the outermost joints when a payload is present.

† The actual values of the RRC arm parameters are proprietary.

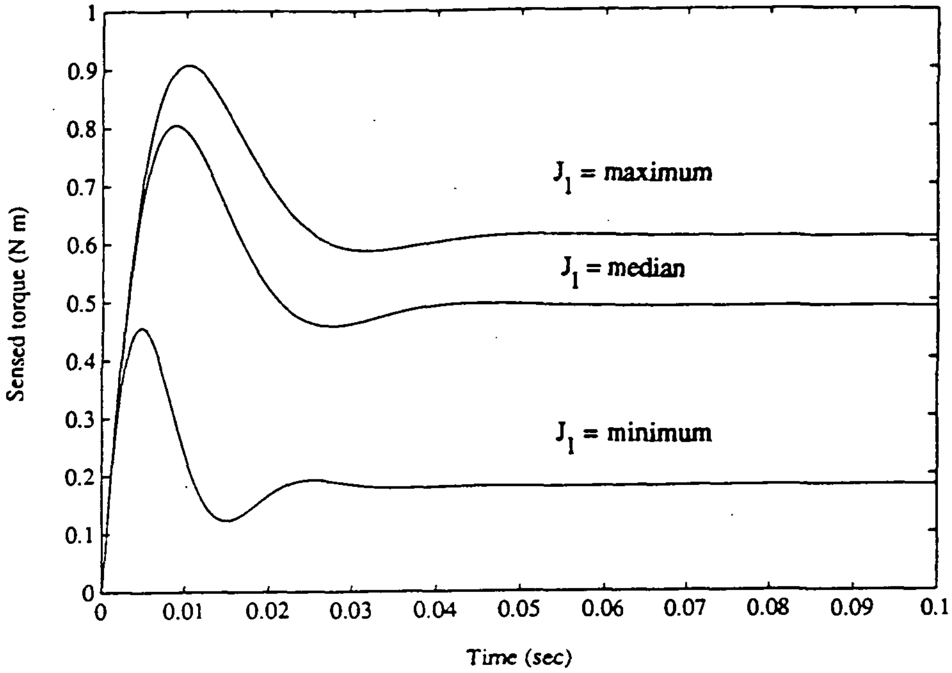

Figure 3.10: Step response of the RRC torque loop for varying inertia values of joint 1

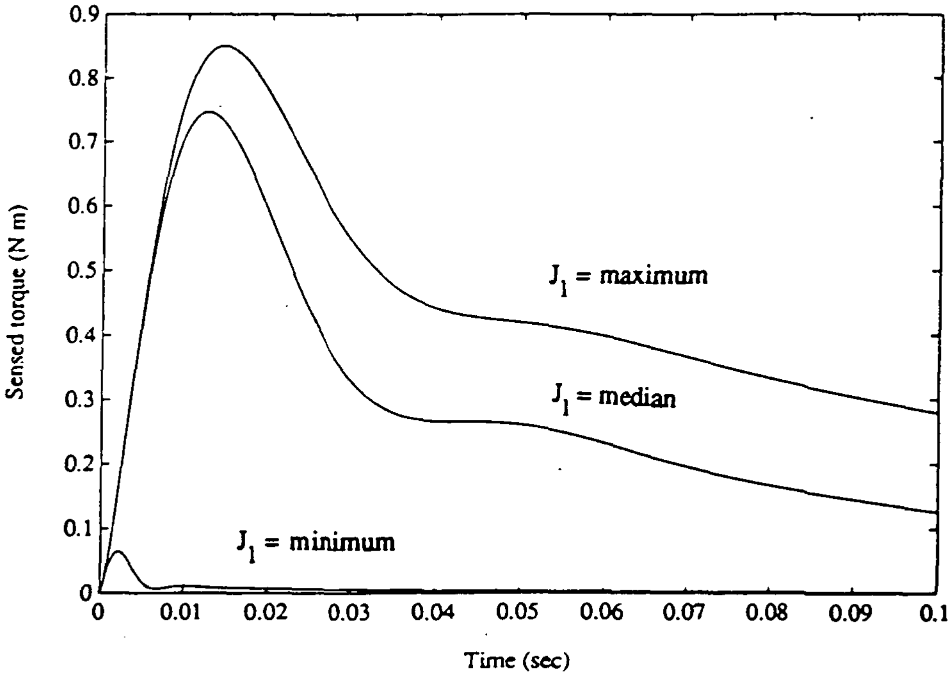

Figure 3.11: Step response of the RRC torque loop for varying inertia values of joint 7

Due to the variation in its dynamics, the RRC torque loop may not be used with the standard computed torque algorithm, Eq. (2.2), for the reasons discussed in Section 2.1.2. A modified computed torque algorithm is proposed in Section 3.4.1 in order to compensate for the effects of the configuration changes on the performance of the RRC torque loop.

3.3.2 Proposed Model Following Torque Loop

The standard computed torque algorithm, Eq. (2.2), was developed under the assumption that the actual torque applied to the link is equal to the commanded torque. Equation (2.6) illustrated the complexities which result when this is not the case. Both the open loop drive system and RRC torque loop were shown to have varying dynamics so that the applied torque does not track the commanded torque. In this section, a model following torque control system is proposed to more accurately track the desired torque command. Such a torque loop would allow direct implementation of the standard computed torque algorithm. The design and analysis of the continuous and discrete time controllers are presented.

In this work, the model following controllers are designed such that they can be added directly to the existing RRC torque loop, as shown in Fig. 3.12. This approach is taken in order to prevent any modification of the existing hardware. Nonetheless, the analysis is general in nature so that the controller could be designed and applied directly to the open loop joint drive system described in Section 3.2.

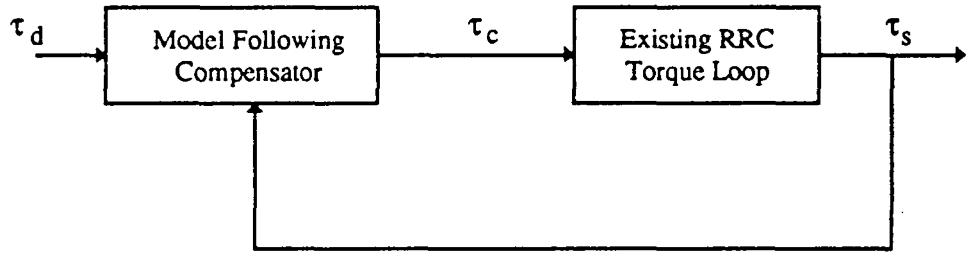

Figure 3.12: Model following torque loop

The continuous model following controller is considered to have fixed gains, since the hardware implementation of variable analog gains can be extremely complex. Therefore, the robustness of this torque control system with respect to varying plant parameters is critical. On the other hand, the discrete model following controller is easily implemented in software in a gain scheduling technique where the controller parameters are periodically updated based on the configuration of the robot arm.

The objective of the model following controller is to regulate the output of the torque loop such that it follows the output of a given model, if both are excited by the same input. This objective is accomplished by attempting to place the poles and zeros of the closed loop system at the exact locations of the poles and zeros of the model. Under several conditions given in Astrom [1989], perfect model following can be achieved with the general linear control law:

$$R\,\tau_c = T\,\tau_d - S\,\tau_s \tag{3.14}$$

where R, S, and T are polynomials in the Laplace (s) operator for continuous systems or the z operator for discrete systems.

The controller is comprised of a feedback compensator given by $-S/R$ and a forward path compensator given by T/R. Block diagrams for the continuous and discrete systems are shown in Figs. 3.13 and 3.14. The design process, given in Astrom [1989], is identical for both cases. A brief review of this technique is outlined below.

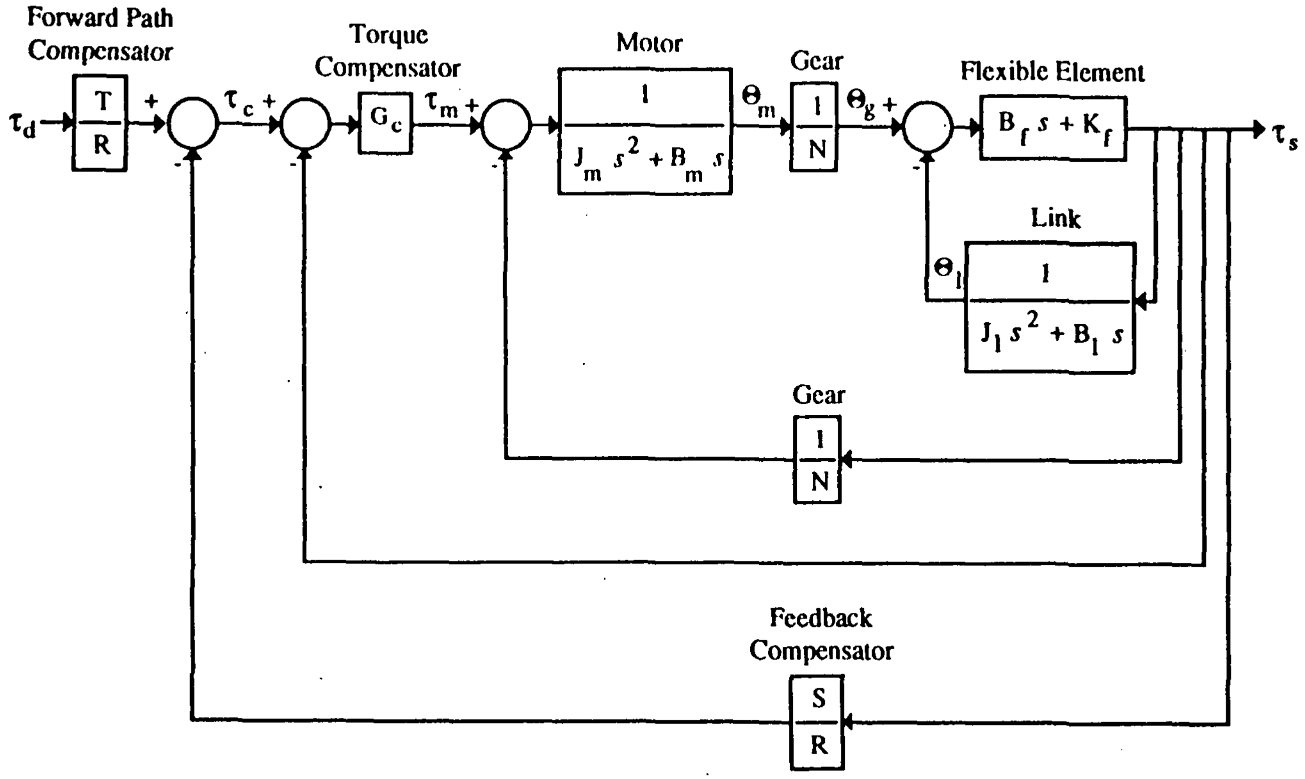

Figure 3.13: Block diagram of the continuous model following torque loop

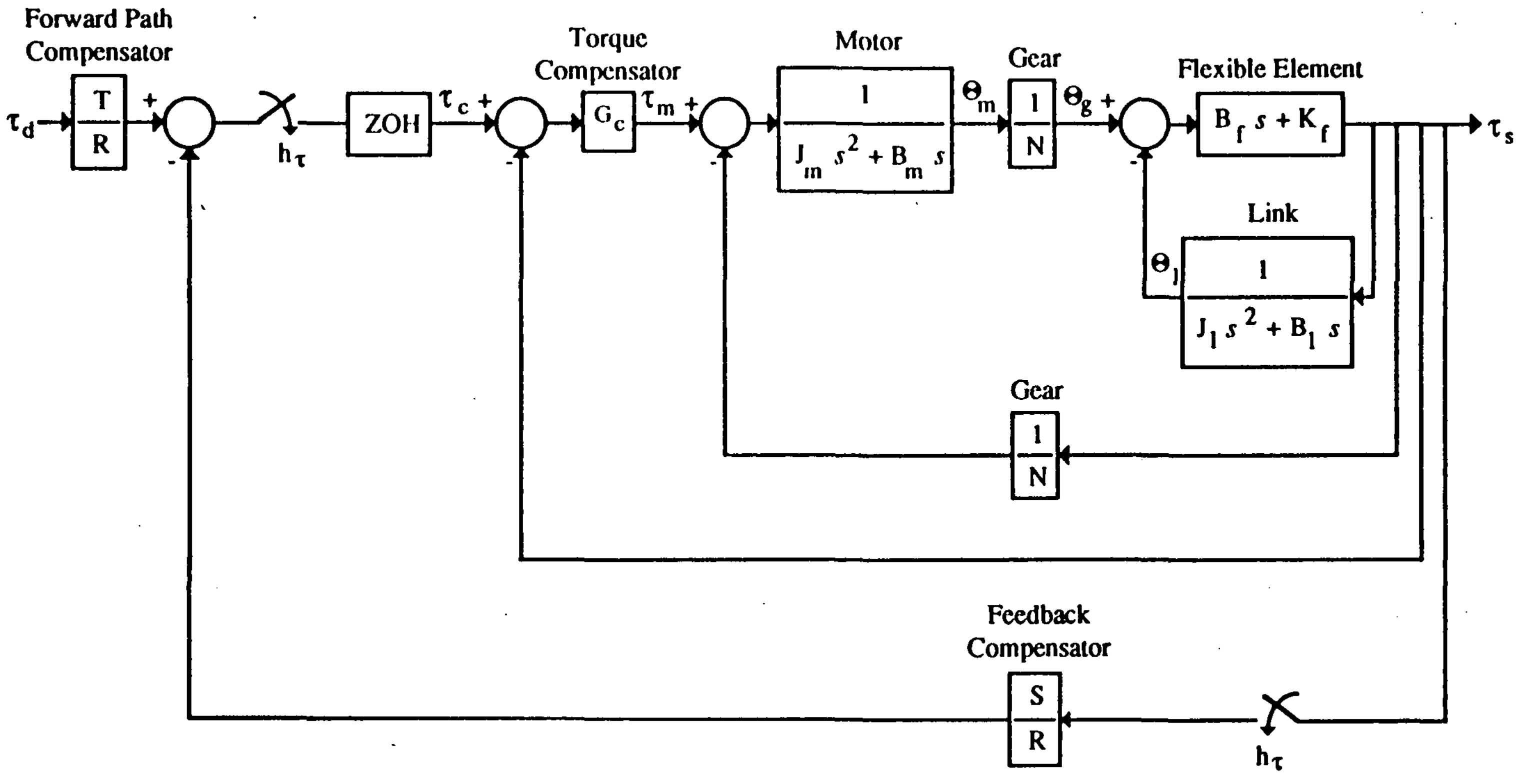

Figure 3.14: Block diagram of the discrete model following torque loop

The dynamics of the plant to be controlled are represented by:

$$\frac{\tau_s}{\tau_c} = \frac{B}{A} \tag{3.15}$$

Consider a model of the form:

$$\frac{\tau_s}{\tau_d} = \frac{B_m}{A_m} \tag{3.16}$$

For causality,

$$\deg A_m - \deg B_m \geq \deg A - \deg B \tag{3.17}$$

Furthermore, an observer polynomial, given by by A_o, may be required. Its degree is given by [Astrom, 1989]:

$$\deg A_o \geq 2 \deg A - \deg A_m - \deg B - 1 \tag{3.18}$$

Combining Eqs. (3.14) through (3.16) and considering the observer polynomial gives:

$$\frac{B\ T}{A\ R + B\ S} = \frac{A_o B_m}{A_o A_m} \tag{3.19}$$

The polynomial B is represented by the product:

$$B = B^{+} B^{-} \tag{3.20}$$

where B^{-} contains the unstable or marginally stable zeros. The stable plant zeros can now be cancelled by specifying R as:

$$R = R_b B^{+} \tag{3.21}$$

In this case, the polynomial T is given by:

$$T = \frac{A_o B_m}{B^{-}} \tag{3.22}$$

from which it is evident that $A_o B_m$ contains the factor B^{-}. The design process then reduces to solving the following Diophantine equation:

$$A R_b + B^{-} S = A_o A_m \tag{3.23}$$

to obtain R_b and S.

From Eq. (3.14), the model following controller for the torque loop is then given by:

$$\tau_c = \frac{T}{R} \tau_d - \frac{S}{R} \tau_s \tag{3.24}$$

3.3.2.1 Continuous Case

The design of a continuous model following controller is considered first. As previously mentioned, the controller gains are considered to be constant in this case. In order to design a robust control system, the minimum value of the link inertia is considered in the plant model. This approach was found to yield a most robust system to variations in inertia and damping.

Considering the block diagram of Fig. 3.8 and Eq. (3.13), the transfer function of the existing RRC torque control loop is given from:

$$\frac{B(s)}{A(s)} = \frac{B_3 s^3 + B_2 s^2 + B_1 s + B_0}{A_4 s^4 + A_3 s^3 + A_2 s^2 + A_1 s + A_0} \qquad (3.25)$$

where:

$$\begin{aligned}
B_3 &= NK_cT_1A_lB_f \\
B_2 &= NK_c\{T_1(A_lK_f+B_lB_f)+A_lB_f\} \\
B_1 &= NK_c(T_1B_lK_f+A_lK_f+B_lB_f) \\
B_0 &= NK_cB_lK_f
\end{aligned}$$

$$\begin{aligned}
A_4 &= N^2T_2A_lJ_m \\
A_3 &= N^2\big(T_2\{A_lB_m+(B_l+B_f)J_m\}+A_lJ_m\big)+T_2A_lB_f+NK_cT_1A_lB_f \\
A_2 &= N^2\big(T_2\{B_m(B_l+B_f)+J_mK_f\}+A_lB_m+(B_l+B_f)J_m\big)+ \\
&\quad T_2(A_lK_f+B_lB_f)+NK_cT_1A_lB_f+A_lB_f+ \\
&\quad NK_c\{T_1(A_lK_f+B_lB_f)+A_lB_f\}
\end{aligned}$$

$$A_1 = N^2\{T_2K_fB_m+(B_l+B_f)B_m+K_fJ_m\}+T_2B_lK_f+A_lK_f+B_lB_f+$$
$$NK_c(T_1B_lK_f+A_lK_f+B_lB_f)$$
$$A_0 = N^2K_fB_m+B_lK_f+NK_cB_lK_f$$

The values of all parameters in the above set of equations are known but proprietary (see Sections 3.2 and 3.3.1).

The torque loop model, $\frac{B_m(s)}{A_m(s)}$, is selected to be a second order system such that the natural frequency and damping of the torque loop may be tuned conveniently. The model is given by:

$$\frac{\tau_s}{\tau_d} = \frac{B_m(s)}{A_m(s)} = \frac{B_{m0}}{A_{m2}s^2 + A_{m1}s + A_{m0}} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \qquad (3.26)$$

which satisfies the causality condition given by Eq. (3.17). Furthermore, the order of the observer polynomial is given by Eq. (3.18) to be greater than or equal to 2. Thus, the following observer polynomial is selected:

$$A_o(s) = s^2 + A_{o1}\,s + A_{o0} = s^2 + 2\,a\,s + a^2 = (s + a)^2 \qquad (3.27)$$

Since the observer dynamics should be faster than the dynamics of the model [Astrom, 1990], the observer poles, $-a$, are chosen to be equal to $-2\omega_n$.

Equation (3.25) represents a minimum phase system for $T_1 > 0$, and therefore, all zeros of the plant can be cancelled, i.e. $B^- = 1$. The degrees of R_b and S are found from the Diophatine equation, Eq. (3.23), to be equal to 1 and 3, respectively. Equations (3.21) through (3.23) are then used to solve for $R(s)$, $T(s)$, and $S(s)$:

$$R(s) \quad = \quad R_3 s^3 + R_2 s^2 + R_1 s + R_0 \tag{3.28}$$

where:
$$\begin{aligned} R_3 &= R_{bc} B_3 \\ R_2 &= R_{bc} B_2 \\ R_1 &= R_{bc} B_1 \\ R_0 &= R_{bc} B_0 \end{aligned}$$

and:
$$R_{bc} = \frac{A_{m2}}{A_4}$$

$$T(s) \quad = \quad T_2 s^2 + T_1 s + T_0 \tag{3.29}$$

where:
$$\begin{aligned} T_2 &= B_{m0} \\ T_1 &= B_{m0} A_{o1} \\ T_0 &= B_{m0} A_{o0} \end{aligned}$$

$$S(s) \quad = \quad S_3 s^3 + S_2 s^2 + S_1 s + S_0 \tag{3.30}$$

where:
$$\begin{aligned} S_3 &= A_{m1}+A_{o1}-A_3R_{bc} \\ S_2 &= A_{m0}+A_{m1}A_{o1}+A_{o0}-A_2R_{bc} \\ S_1 &= A_{m1}A_{o0}+A_{m0}A_{o1}-A_1R_{bc} \\ S_0 &= A_{m0}A_{o0}-A_0R_{bc} \end{aligned}$$

Given the values of the parameters of the RRC torque loop and having specified the natural frequency, ω_n, and damping ratio, ζ, of the model, the polynomials, R, S, and T, are obtained for each joint. The choice for ω_n and ζ should be such that the dynamics of the continuous elements of the arm are not excited. Since the RRC torque loop is already tuned to avoid excitation of these dynamics, its bandwidth serves as the basis for selecting the natural frequency and damping for the model following torque loop. Figures 3.15 and 3.16 show the Bode magnitude plots for the RRC torque loops at joints 1 and 7. Their bandwidths range between 300 to 500 rad/sec. Based on this observation, the natural frequency and damping ratio for the model following torque loops at each joint are chosen to be 400 rad/sec and 1.0, respectively.

Robustness Analysis

It is emphasized that the parameters of the model following controller are determined using the minimum effective link inertia for each joint. However, as previously mentioned, the effective link inertia, J_l, changes with configuration, and the link damping, B_l, may be difficult to determine. Thus, the issue of robustness is of great concern for the model following torque loop. To examine robust stability, the small gain theorem is used [Morari, 1989; Dailey, 1990] which, for single-input/single-output systems, is equivalent to the Nyquist stability criterion.

Following the standard robustness analysis technique, the parameter uncertainty is represented by a transfer function, Δ, and the transfer function of the remaining

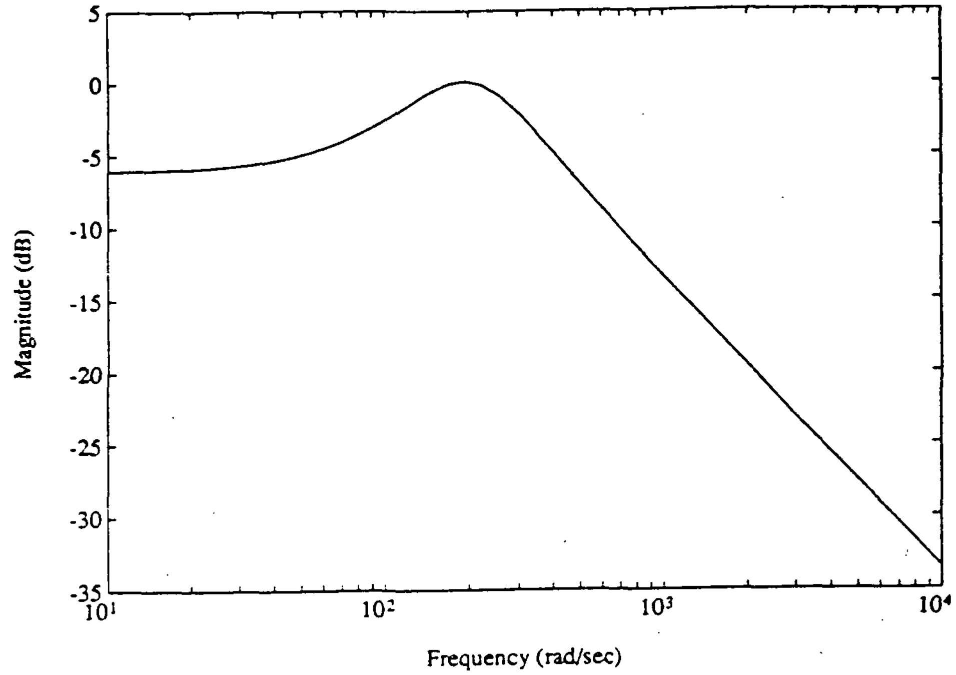

Figure 3.15: Bode magnitude plot of the RRC torque loop for joint 1

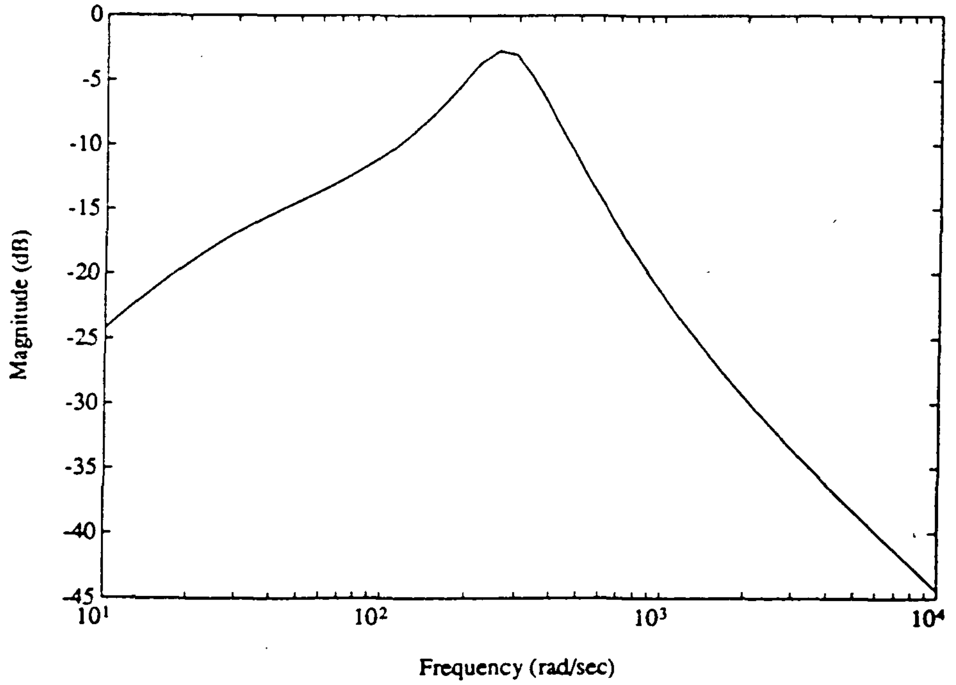

Figure 3.16: Bode magnitude plot of the RRC torque loop for joint 7

closed loop system is represented by M. Figure 3.17 shows the transformed torque loop. The link inertia and damping terms are represented by:

$$\begin{aligned} J_l &= J_{l_{nom}} + \Delta J_l \\ B_l &= B_{l_{nom}} + \Delta B_l \end{aligned} \tag{3.31}$$

where $J_{l_{nom}}$ and $B_{l_{nom}}$ are the nominal values of link inertia and damping that were used to design the model following controller. ΔJ_l and ΔB_l are the deviations from these nominal values. The transfer functions, Δ and M, are then given by:

$$\Delta = \frac{-(\Delta J_l\, s^2 + \Delta B_l\, s)}{(J_l + \Delta J_l)\, s^2 + (B_l + \Delta B_l)\, s} \tag{3.32}$$

$$M = \frac{N_M}{D_M} \tag{3.33}$$

$$\begin{aligned} \text{where:} \quad N_M &= N^2 R\,(T_2 s + 1)\,(B_f s + K_f)\,(J_m s^2 + B_m s) \\ D_M &= N^2 R\,(T_2 s + 1)\,(J_m s^2 + B_m s)\,(J_l s^2 + B_l s)\ + \\ &\quad N^2 R\,(T_2 s + 1)\,(J_m s^2 + B_m s)\,(B_f s + K_f)\ + \\ &\quad N\,K_c\,S\,(T_1 s + 1)\,(B_f s + K_f)\,(J_l s^2 + B_l s)\ + \\ &\quad N\,K_c\,R\,(T_1 s + 1)\,(B_f s + K_f)\,(J_l s^2 + B_l s)\ + \\ &\quad R\,(T_2 s + 1)\,(B_f s + K_f)\,(J_l s^2 + B_l s) \end{aligned}$$

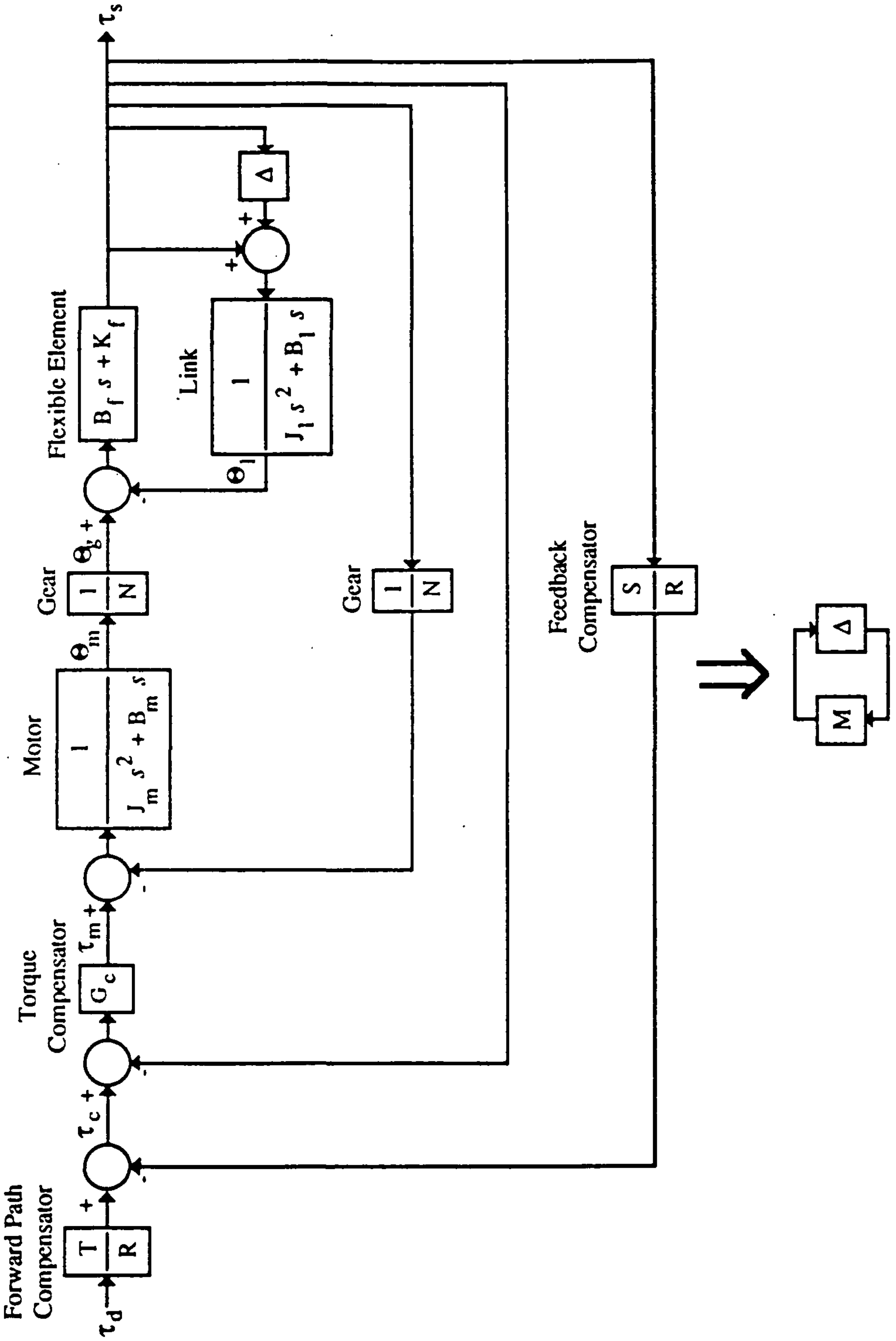

Figure 3.17: Robustness test model

The small gain theorem states that the closed loop system shown in Fig. 3.17 is stable if:

$$|\Delta(j\omega)\ M(j\omega)| \leq 1 \qquad \forall\, \omega \quad \in R^{+} \tag{3.34}$$

or equivalently,

$$|\Delta(j\omega)| \leq \frac{1}{|M(j\omega)|} \qquad \forall\, \omega \quad \in R^{+} \tag{3.35}$$

Based on Eq. (3.35), robust stability can be examined by plotting the magnitudes of the functions, $\Delta(j\omega)$ and $\frac{1}{M(j\omega)}$, versus ω and verifying that the magnitude of Δ is always less than the magnitude of $\frac{1}{M}$.

It is desirable for the model following torque loop to remain stable for all reasonable uncertainties in link inertia and damping. Since the minimum effective link inertia was used in the design of the model following controller, the term, ΔJ_l, should always be positive. From Eq. (3.32), it follows that the maximum possible absolute value of the uncertainty, $|\Delta(j\omega)|$, is 1, regardless of the uncertainty in damping. Thus, the model following torque loop will remain stable for all possible inertia and damping uncertainties if:

$$\frac{1}{|M(j\omega)|} \leq 1 \qquad \forall\, \omega \quad \in R^{+} \tag{3.36}$$

Figures 3.18 and 3.19 are plots of $\frac{1}{|M(j\omega)|}$ for joints 1 and 7, respectively, with $\omega_n = 400$ rad/sec and $\zeta = 1.0$. The plots indicate that these joints remain stable for all possible inertia and damping uncertainties. Similar results are obtained for the remaining joints.

To examine robust performance, simulations are again used under several possible values of inertia. Figures 3.20 and 3.21 show the simulated step responses of the model following torque loop for joints 1 and 7 under the same inertia variations considered in Figs. 3.10 and 3.11. These figures indicate that the dynamics of the model following torque loop change less drastically with configuration compared to the RRC torque loop.

3.3.2.2 Discrete Case

Unlike the continuous model following controller, the discrete version allows the on-line computation of the model following compensators, *R*, *S*, and *T*, based on the configuration of the robot. In this manner, the changing link inertias can be accounted for by updating *R*, *S*, and *T*. The on line tuning of these parameters can be considered as a gain scheduling technique in which the scheduling "variable" is the robot configuration. The controller design follows the same steps as described for the continuous case.

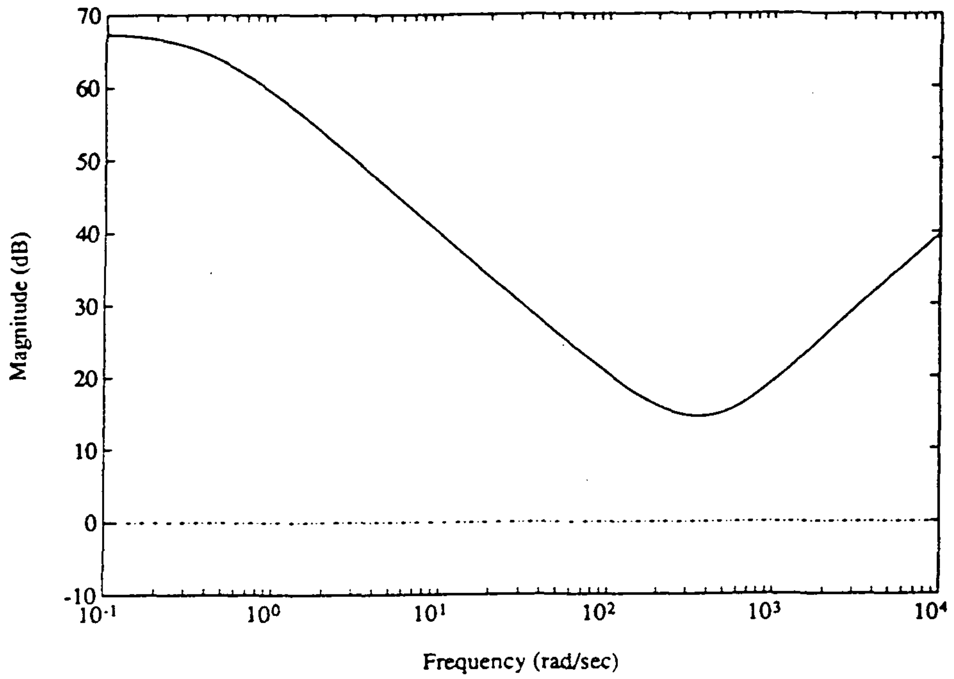

Figure 3.18: Robustness test plot for joint 1

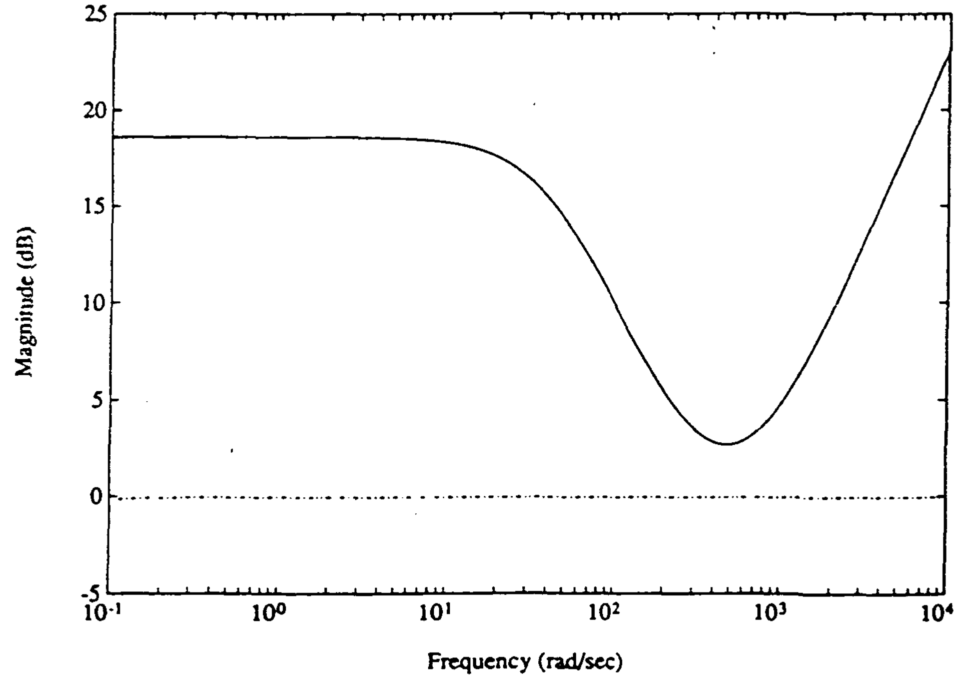

Figure 3.19: Robustness test plot for joint 7

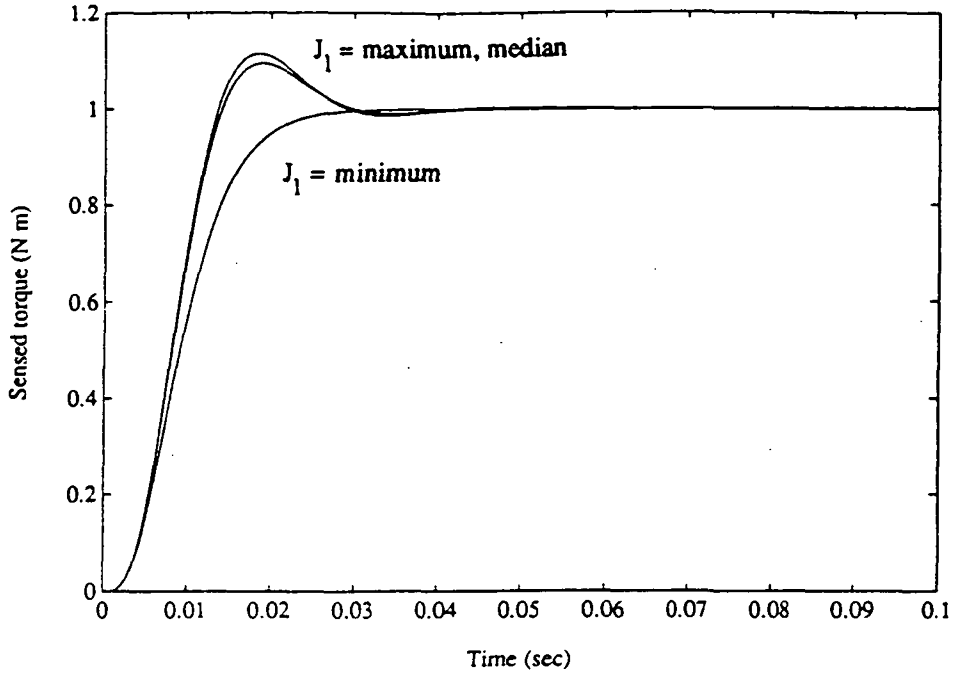

Figure 3.20: Step response of the continuous model following torque loop for joint 1

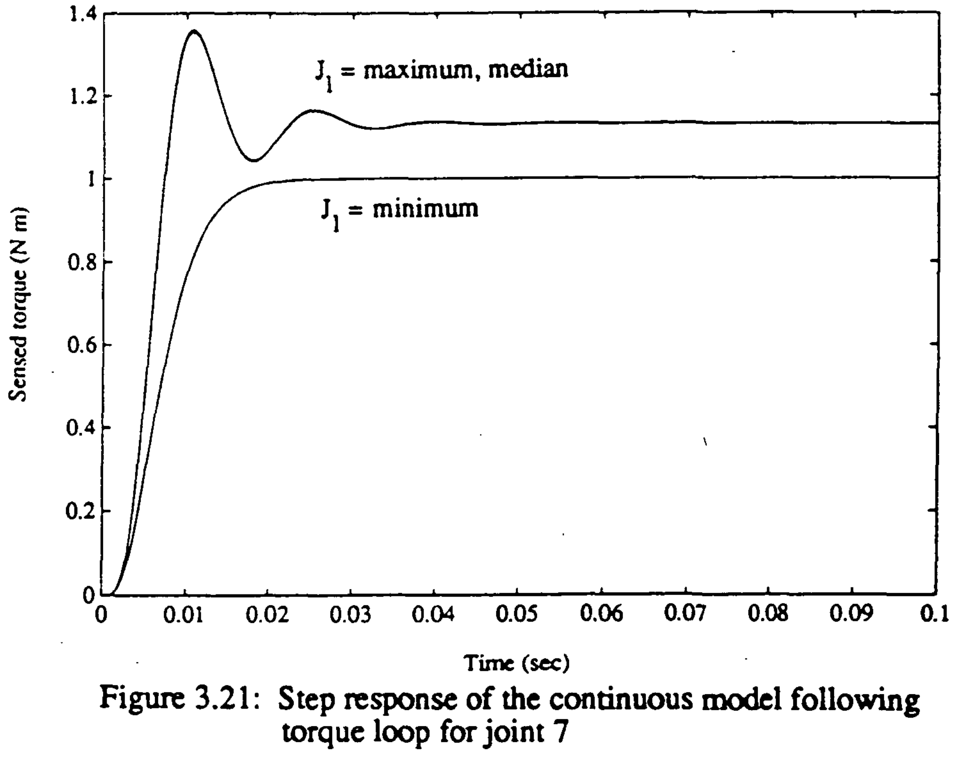

Figure 3.21: Step response of the continuous model following torque loop for joint 7

A discrete time representation of Eq. (3.25) is obtained by using the bilinear transformation given by Astrom [1989]:

$$s = \frac{2}{h_\tau} \frac{z - 1}{z + 1} \tag{3.37}$$

where h_τ is the sampling period. For small sampling periods, this transformation provides a good approximation of the ZOH equivalent of the plant. Substituting Eq. (3.37) into Eq. (3.25) gives:

$$\frac{B(z)}{A(z)} = \frac{\mathcal{B}_4 z^4 + \mathcal{B}_3 z^3 + \mathcal{B}_2 z^2 + \mathcal{B}_1 z + \mathcal{B}_0}{\mathcal{A}_4 z^4 + \mathcal{A}_3 z^3 + \mathcal{A}_2 z^2 + \mathcal{A}_1 z + \mathcal{A}_0} \tag{3.38}$$

where:

$$\begin{aligned}
\mathcal{B}_4 &= 8B_3h_\tau+4B_2h_\tau^2+2B_1h_\tau^3+B_0h_\tau^4 \\
\mathcal{B}_3 &= -16B_3h_\tau+4B_1h_\tau^3+4B_0h_\tau^4 \\
\mathcal{B}_2 &= -8B_2h_\tau^2+6B_0h_\tau^4 \\
\mathcal{B}_1 &= 16B_3h_\tau-4B_1h_\tau^3+4B_0h_\tau^4 \\
\mathcal{B}_0 &= -8B_3h_\tau+4B_2h_\tau^2-2B_1h_\tau^3+B_0h_\tau^4
\end{aligned}$$

$$\begin{aligned}
\mathcal{A}_4 &= 16A_4+8A_3h_\tau+4A_2h_\tau^2+2A_1h_\tau^3+A_0h_\tau^4 \\
\mathcal{A}_3 &= -64A_4-16A_3h_\tau+4A_1h_\tau^3+4A_0h_\tau^4 \\
\mathcal{A}_2 &= 96A_4-8A_2h_\tau^2+6A_0h_\tau^4 \\
\mathcal{A}_1 &= -64A_4+16A_3h_\tau-4A_1h_\tau^3+4A_0h_\tau^4 \\
\mathcal{A}_0 &= 16A_4-8A_3h_\tau+4A_2h_\tau^2-2A_1h_\tau^3+A_0h_\tau^4
\end{aligned}$$

Since deg $A(s)$ - deg $B(s) = 1$, the bilinear transformation yields a zero at -1. In this case, the polynomial, B^-, is given by:

$$B^- \;=\; (z+1) \tag{3.39}$$

For the discrete time model, the ZOH equivalent of the continuous time model with $\zeta = 1.0$ is chosen. An additional zero at -1 is necessary for the model to satisfy Eq. (3.22). The model is then given by:

$$\frac{B_m(z)}{A_m(z)} \;=\; \frac{0.5\,(z+1)\,(\mathcal{B}_{m1} z + \mathcal{B}_{m0})}{z^2 + \mathcal{A}_{m1} z + \mathcal{A}_{m0}} \tag{3.40}$$

where:

$$\mathcal{B}_{m1} \;=\; 1.0 - e^{-\omega_n h_\tau}\,(1.0 + \omega_n h_\tau)$$

$$\mathcal{B}_{m0} \;=\; e^{-\omega_n h_\tau}\,(e^{-\omega_n h_\tau} + \omega_n h_\tau - 1.0)$$

$$\mathcal{A}_{m1} \;=\; -2.0\,e^{-\omega_n h_\tau}$$

$$\mathcal{A}_{m0} \;=\; e^{-2\omega_n h_\tau}$$

The observer is also chosen as the ZOH equivalent of the continuous time observer and is given by:

$$A_o(z) \;=\; z^2 + \mathcal{A}_{o1}\, z \;+\; \mathcal{A}_{o0} \tag{3.41}$$

where:

$$\mathcal{A}_{m1} \;=\; -2.0\,e^{-ah_\tau}$$

$$\mathcal{A}_{m0} \;=\; e^{-2ah_\tau}$$

The observer parameter, a, is chosen to be twice the natural frequency of the model. The model natural frequency and damping ratio are selected by assuming a sampling rate of 1000 Hz and executing several simulations with different model parameters. The best combination of parameters was selected from these simulations to be ω_n = 300 rad/sec and $\zeta = 1.0$.

Eqs. (3.30) through (3.32) are used to solve for $R(z)$, $T(z)$, and $S(z)$ as:

$$R(z) \quad = \quad \mathcal{R}_3\, z^3 + \mathcal{R}_2\, z^2 + \mathcal{R}_1\, z + \mathcal{R}_0 \tag{3.42}$$

where:

$$\mathcal{R}_3 \quad = \quad R_{1d}\, \mathcal{B}_4$$

$$\mathcal{R}_2 \quad = \quad R_{1d}\, (\mathcal{B}_3 - \mathcal{B}_4)$$

$$\mathcal{R}_1 \quad = \quad R_{1d}\, (\mathcal{B}_4 - \mathcal{B}_3 + \mathcal{B}_2)$$

$$\mathcal{R}_0 \quad = \quad R_{1d}\, (\mathcal{B}_4 - \mathcal{B}_3 + \mathcal{B}_2 - \mathcal{B}_0)$$

and:

$$R_{1d} \quad = \quad \frac{C_4}{D_4}$$

$$\begin{aligned} C_4 \quad = \quad & \mathcal{A}_{o0}\, \mathcal{A}_{m0} - \mathcal{A}_{o1}\, \mathcal{A}_{m0} - \mathcal{A}_{o0}\, \mathcal{A}_{m1} + \mathcal{A}_{m0} + \mathcal{A}_{o1}\, \mathcal{A}_{m1} + \\ & \mathcal{A}_{o0} - \mathcal{A}_{m1} - \mathcal{A}_{o1} + 1 \end{aligned}$$

$$D_4 \quad = \quad \mathcal{A}_4 - \mathcal{A}_3 + \mathcal{A}_2 - \mathcal{A}_1 + \mathcal{A}_0$$

$$T(z) \quad = \quad \mathcal{T}_3\, z^3 + \mathcal{T}_2\, z^2 + \mathcal{T}_1\, z + \mathcal{T}_0 \tag{3.43}$$

where:

$$\mathcal{T}_3 \quad = \quad \mathcal{B}_{m1} \; -$$

$$\mathcal{T}_2 \quad = \quad \mathcal{B}_{m0} + \mathcal{B}_{m1}\, \mathcal{B}_{o1}$$

$$\mathcal{T}_1 \quad = \quad \mathcal{B}_{m0}\, \mathcal{A}_{o1} + \mathcal{B}_{m1}\, \mathcal{A}_{o0}$$

$$\mathcal{T}_0 \quad = \quad \mathcal{B}_{m0}\, \mathcal{A}_{o0}$$

$$S(z) \;=\; \mathcal{S}_3 z^3 + \mathcal{S}_2 z^2 + \mathcal{S}_1 z + \mathcal{S}_0 \qquad (3.44)$$

$$\begin{aligned}
\text{where:}\quad \mathcal{S}_3 &= \frac{-C_4 D_3}{D_4} - C_3 \\
\mathcal{S}_2 &= \frac{-C_4 D_2}{D_4} + C_2 \\
\mathcal{S}_1 &= \frac{-C_4 D_1}{D_4} - C_1 \\
\mathcal{S}_0 &= \frac{-C_4 D_0}{D_4} + C_0 \\
\text{and:}\quad C_3 &= \mathcal{A}_{o0}\mathcal{A}_{m0} - \mathcal{A}_{o1}\mathcal{A}_{m0} - \mathcal{A}_{o0}\mathcal{A}_{m1} + \mathcal{A}_{m0} + \mathcal{A}_{o1}\mathcal{A}_{m1} + \\
&\quad \mathcal{A}_{o0} - \mathcal{A}_{m1} - \mathcal{A}_{o1} \\
C_2 &= \mathcal{A}_{o0}\mathcal{A}_{m0} - \mathcal{A}_{o1}\mathcal{A}_{m0} - \mathcal{A}_{o0}\mathcal{A}_{m1} + \mathcal{A}_{m0} + \mathcal{A}_{o1}\mathcal{A}_{m1} + \\
&\quad \mathcal{A}_{o0} \\
C_1 &= \mathcal{A}_{o0}\mathcal{A}_{m0} - \mathcal{A}_{o1}\mathcal{A}_{m0} - \mathcal{A}_{o0}\mathcal{A}_{m1} \\
C_0 &= \mathcal{A}_{o0}\mathcal{A}_{m0} \\
D_3 &= \mathcal{A}_3 - \mathcal{A}_2 + \mathcal{A}_1 - \mathcal{A}_0 \\
D_2 &= \mathcal{A}_2 - \mathcal{A}_1 + \mathcal{A}_0 \\
D_1 &= \mathcal{A}_1 - \mathcal{A}_0 \\
D_0 &= \mathcal{A}_0
\end{aligned}$$

The coefficients of the *R*, *S*, and *T* polynomials given by the above equations can be determined given the parameters of the RRC torque loop.

Figures 3.22 and 3.23 show the simulated step responses of the discrete model following torque loop at joints 1 and 7 under the previously considered link inertia variations. It is evident from the figures that the discrete model following torque loop performs very well in simulation for the various link inertias. It is emphasized

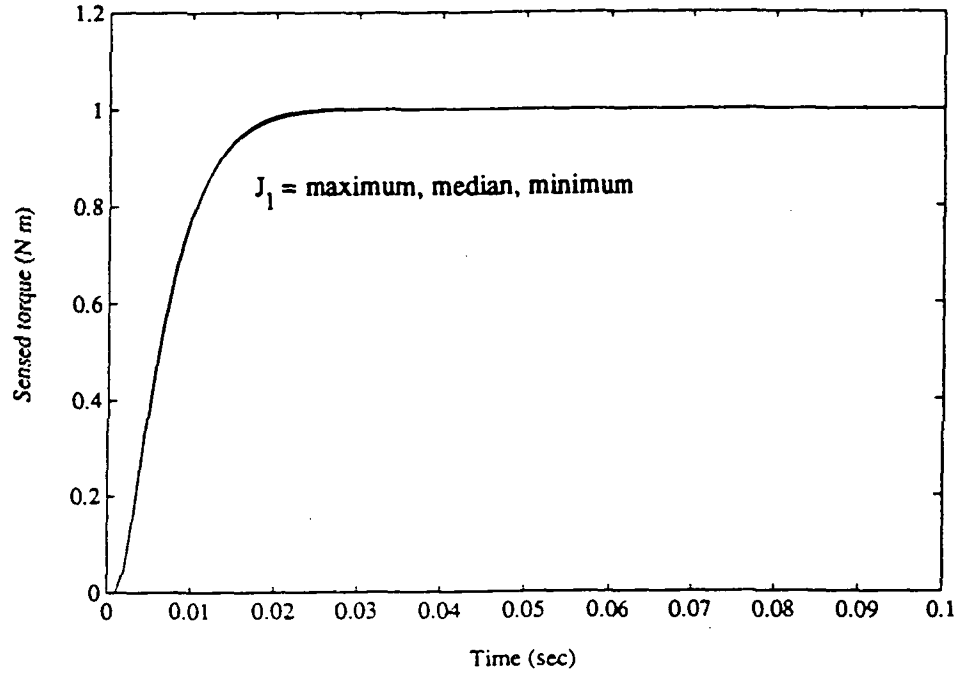

Figure 3.22: Step response of the discrete model following torque loop for joint 1

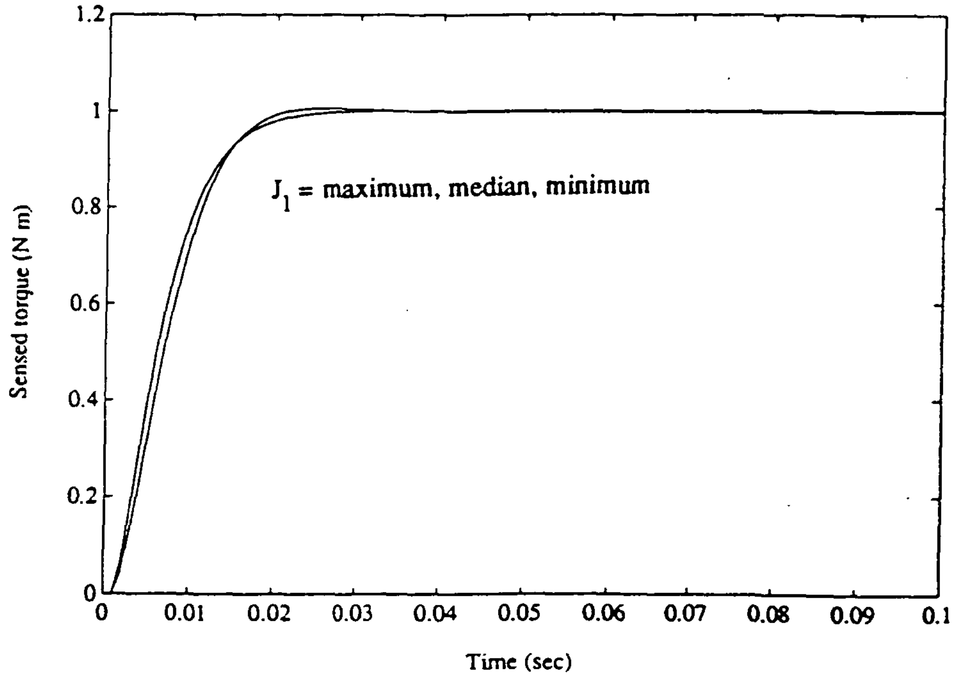

Figure 3.23: Step response of the discrete model following torque loop for joint 7

that this controller is easily implemented in software and, therefore, can compensate for the variations in link inertia. On line-compensation requires periodically computing J_l given the robot configuration and then calculating $R(z)$, $T(z)$, and $S(z)$ using Eqs. (3.42) through (3.44). This adjustment is not possible with the fixed gain RRC torque loop or the fixed gain continuous model following torque loop.

3.4 Computed Torque Control

Two approaches to torque control were presented in the Section 3.3: the RRC torque loop and a novel model following torque controller (both continuous and discrete). This section presents the computed torque algorithms to be used with these two types of torque loops. The proposed "Equivalent Computed Torque" algorithm is used with the existing RRC torque loop. The standard computed torque algorithm given by Eq. (2.2) is used with the proposed model following torque loops.

3.4.1 Proposed Equivalent Computed Torque Algorithm

A novel form of the computed torque algorithm is proposed which can be used directly with the existing RRC joint torque loops. It is based on the concept of "Equivalent Inertia and Damping", described below, and therefore, is referred to as the "Equivalent Computed Torque" (ECT) algorithm. It is noted that the standard computed torque algorithm is not appropriate for use with the RRC torque loop computed torque algorithm is not appropriate for use with the RRC torque loop since variations in the dynamics of the torque loop result in the non-linear, coupled dynamical equations given in Section 2.1.2 (see Eq. (2.6)).

The proposed algorithm is based on heuristics in which the instantaneous torque command required to produce a desired joint motion is approximated by the steady state torque value required to produce a desired steady state motion. The torque commands are determined by summing of three torque values: i) the steady state torque command, τ_{ca}, required to produce the desired acceleration when no damping losses are present; ii) the steady state torque command, τ_{cdamp}, required to compensate for the damping losses; iii) the steady state torque command τ_{cdist}, required to compensate for disturbance torques which result from the gravity loading and the motions of the other joints. The first two components are evaluated by introducing the concepts of equivalent inertia and damping.

Equivalent Inertia and Damping

The "Equivalent Inertia" and "Equivalent Damping" are defined as:

> Definition 1: *Equivalent inertia* is the steady state value of the commanded torque required to drive the link at a constant unit angular acceleration when no damping losses are considered.

Definition 2: *Equivalent damping* is the steady state value of commanded torque required to drive the link at a constant unit angular velocity.

According to these definitions, the equivalent inertia and damping can be found by examining the RRC torque loop transfer function given by:

$$\frac{\tau_s}{\tau_c} = \frac{G\ G_c}{1 + G\ G_c} \qquad (3.45)$$

where G is given by Eq. (3.10) and G_c is given by Eq. (3.13).

For the single joint model considered here, the link is driven by the output of the torque loop, and the link dynamics are given by:

$$\frac{\Theta}{\tau_s} = \frac{1}{J_l\, s^2 + B_l\, s} \qquad (3.46)$$

Substituting for G and G_c in Eq. (3.45), multiplying Eqs. (3.45) and (3.46), and inverting yields:

$$\frac{\tau_c}{\Theta} = \frac{N^2(B_f s + K_f)(T_2 s + 1) + N^2(J_l s^2 + B_l s)(T_2 s + 1)}{N\ K_c\ (T_1 s + 1)(Bfs + K_f)}\ J_m s^2\ +$$

$$\frac{(B_f s + K_f)(T_2 s + 1) + NK_c(T_1 s + 1)(B_f s + K_f)}{N\ K_c\ (T_1 s + 1)(B_f s + K_f)}\ J_l s^2\ +$$

$$\frac{N^2(B_f s + K_f)(T_2 s + 1) + N^2(J_l s^2 + B_l s)(T_2 s + 1)}{N\ K_c\ (T_1 s + 1)(B_f s + K_f)}\ B_m s\ +$$

$$\frac{(B_f s + K_f)(T_2 s + 1) + NK_c(T_1 s + 1)(B_f s + K_f)}{N\ K_c\ (T_1 s + 1)(B_f s + K_f)}\ B_l s \tag{3.47}$$

For $T_1 > 0$, Eq. (3.47) represents a stable system, and therefore, the steady state value of the torque required to drive the link at a constant unit acceleration with no damping losses can be found by setting the last two terms equal to zero and applying the final value theorem for a unit step in angular acceleration. The corresponding torque command is given by:

$$\tau_{ca} = \frac{N^2\ J_m}{N\ K_c} + \frac{(1 + N\ K_c)\ J_l}{N\ K_c} = J_{meq} + J_{leq} = J_{eq} \tag{3.48}$$

The two right hand side terms of Eq. (3.48) may be thought of as the motor equivalent inertia and link equivalent inertia, respectively.

Similarly, the equivalent damping may be found from Eq. (3.47) by applying the final value theorem for a unit step in angular velocity:

$$\tau_{cdamp} = \frac{N^2 B_m}{N K_c} + \frac{(1 + N K_c) B_l}{N K_c} = B_{meq} + B_{leq} = B_{eq} \quad (3.49)$$

Considering a single joint moving at an angular velocity, $\dot{\Theta}_{la}$, Eq. (3.49) implies that the steady state torque command:

$$\tau_{cdamp} = B_{eq} \dot{\Theta}_{la} \tag{3.50}$$

will compensate for the damping losses and, thus, maintain this angular velocity. Similarly, Eq. (3.48) implies that the steady state torque command given by:

$$\tau_{ca} = J_{eq} \ddot{\Theta}_{ld} \tag{3.51}$$

will yield a steady state angular acceleration, $\ddot{\Theta}_{ld}$, if no damping losses occur. Assuming that the instantaneous torque requirements are approximately equal to the steady state torque requirements, the sum of the two torque commands given by Eqs. (3.50) and (3.51) is considered to be the torque command required to obtain the desired angular acceleration, $\ddot{\Theta}_{ld}$, at the angular velocity, $\dot{\Theta}_{la}$. It is noted that this is a heuristic, the validity of which is tested in Sections 4 and 5.

This single joint case is generalized to include all joints in the next section. The complete Equivalent Computed Torque Algorithm is subsequently developed by including a third term to account for the disturbance torques.

Equivalent Computed Torque Algorithm

The sum, $\tau_{ca} + \tau_{cdamp}$, for all joints may be formed by considering that the terms, J_l, are given by the diagonal elements of M (see Section 3.2). Using Eqs. (3.48) and (3.49), $\tau_{ca} + \tau_{cdamp}$ can be written as:

$$\tau_{ca} + \tau_{cdamp} = \left[\mathrm{diag} \left[\frac{1 + N\ K_c}{N\ K_c} \right] \mathrm{diag}\ [M] + \mathrm{diag}\ [J_{meq}] \right] \ddot{\Theta}_d + \mathrm{diag}\ [B_{eq}]\ \dot{\Theta}_a \tag{3.52}$$

where diag indicates a diagonal matrix. If the position and velocity feedback terms of Eq. (2.2) are added to Eq. (3.52) and the estimates of the inertia and damping terms are used, the following form of the computed torque equation is obtained:

$$\tilde{\tau}_c = \left[\mathrm{diag} \left[\frac{1 + N\ K_c}{N\ K_c} \right] \mathrm{diag}\ [\hat{M}\ ] + \mathrm{diag}\ [\hat{J}_{meq}] \right] * \left\{ \ddot{\Theta}_d + K_d (\dot{\Theta}_d - \dot{\Theta}_a) + K_p (\Theta_d - \Theta_a) \right\} + \mathrm{diag}\ (\hat{B}_{eq})\ \dot{\Theta}_a \tag{3.53}$$

Equation (3.53) should include a third term, τ_{cdist}, which accounts for the disturbance torques that result from the off diagonal link accelerations and the Coriolis, centrifugal, and gravity torques. The steady state torque command, τ_{cdist}, required to cancel these disturbance effects can be found by considering the block diagram of Fig. 3.24. When a disturbance torque, τ_{dist}, is present, the net torque applied to the link, τ_{app}, is equal to the sum of the disturbance and the sensed torques. This sum can be found by reducing the block diagram to obtain:

$$\tau_{app} = -\tau_{dist} + \frac{B_3 s^3 + B_2 s^2 + B_1 s + B_0}{A_4 s^4 + A_3 s^3 + A_2 s^2 + A_1 s + A_0} \tau_c$$

$$+ \frac{B_7 s^3 + B_6 s^2 + B_5 s + B_4}{A_4 s^4 + A_3 s^3 + A_2 s^2 + A_1 s + A_0} \tau_{dist} \qquad (3.54)$$

where: $B_7 = N^2 T_2 J_m B_f$

$B_6 = N^2(T_2(J_m K_f + B_m B_f) + J_m B_f)$

$B_5 = N^2(T_2 B_m K_f + J_m K_f + B_m B_f)$

$B_4 = N^2 B_m K_f$

and the other terms are given by Eq. (3.25).

To cancel the effects of the disturbance torque such that $\tau_{app} = 0$, the steady state value of the commanded torque can be found Eq. (3.54), using the final value theorem:

$$\tau_{cdist} = \tau_c = \frac{-(1 + N K_c)}{N K_c} \tau_{dist} \qquad (3.55)$$

This scale factor, $\frac{1 + N K_c}{N K_c}$, is used to compute the torque commands that are necessary to reject the disturbances from the off diagonal inertia elements and the Coriolis, centrifugal, and gravity disturbances. Equation (3.55) may be written in vector form to include all joints. Utilizing the estimated dynamical parameters of $\hat{M}$ and $\hat{Q}$ gives:

$$\tau_{cdist} = \text{diag}\left[\frac{1 + N\ K_c}{N\ K_c}\right]\left[\hat{M} - \text{diag}\ [\hat{M}\ ]\right]\ddot{\Theta}_d + \text{diag}\left[\frac{1 + N\ K_c}{N\ K_c}\right]\hat{Q} \tag{3.56}$$

Adding τ_{cdist} to Eq. (3.53) gives the Equivalent Computed Torque equation:

$$\begin{aligned}\tau_c = &\left[\text{diag}\left[\frac{1 + N\ K_c}{N\ K_c}\right]\hat{M} + \text{diag}\ [\hat{J}_{meq}]\right] * \\ &\left\{\ddot{\Theta}_d + K_d(\dot{\Theta}_d - \dot{\Theta}_a) + K_p(\Theta_d - \Theta_a)\right\} + \text{diag}\left[\frac{1 + N\ K_c}{N\ K_c}\right]\hat{Q} + \\ &\text{diag}\ [\hat{B}_{meq}]\ \dot{\Theta}_a\end{aligned} \tag{3.57}$$

It is noted that only the link, and not the motor rotor or transmission, inertia and damping parameters are included in the $\hat{M}$ and $\hat{Q}$ terms.

The control scheme that utilizes Eq. (3.57) will be refered to as the Equivalent Computed Torque (ECT) controller. It is similar in form to the standard computed torque controller of Eq. (2.2), with the addition of the motor equivalent inertia and damping terms, and the scaling of the rigid link manipulator dynamics by the diagonal matrix, $\text{diag}\left[\frac{1 + N\ K_c}{N \cdot K_c}\right]$.

Implementation

The complete ECT robot controller is implemented as shown in the block diagram of Fig. 3.25. It consists of two parts: the ECT algorithm and the RRC torque loop. The torque commands are computed from Eq. (3.57) at finite time intervals. The update rate depends directly on the computational requirements of the rigid body

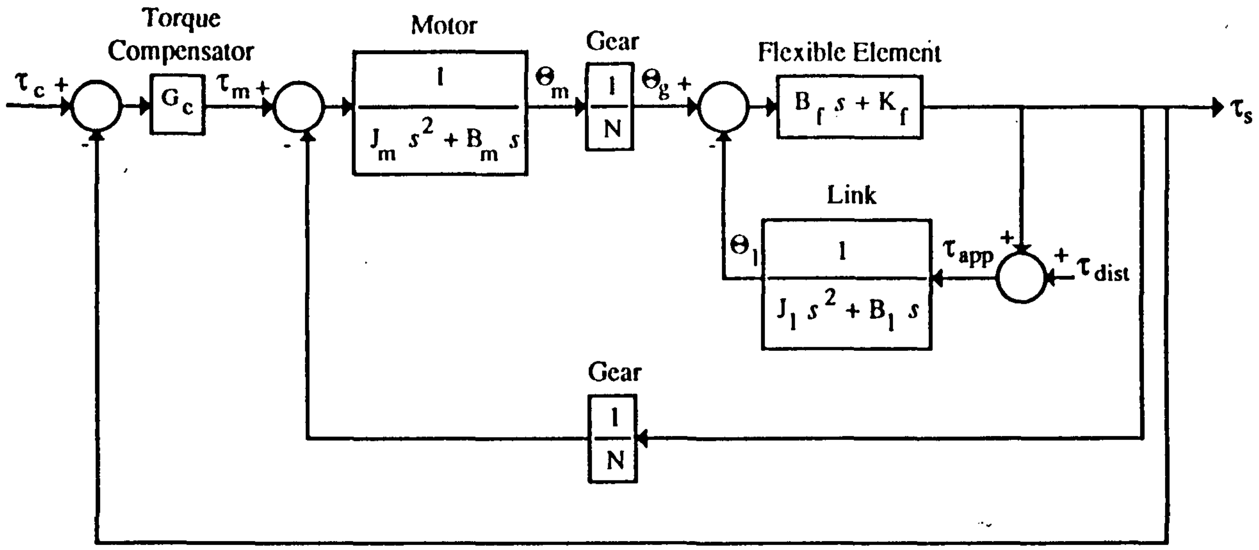

Figure 3.24: Disturbance model for the RRC torque loop

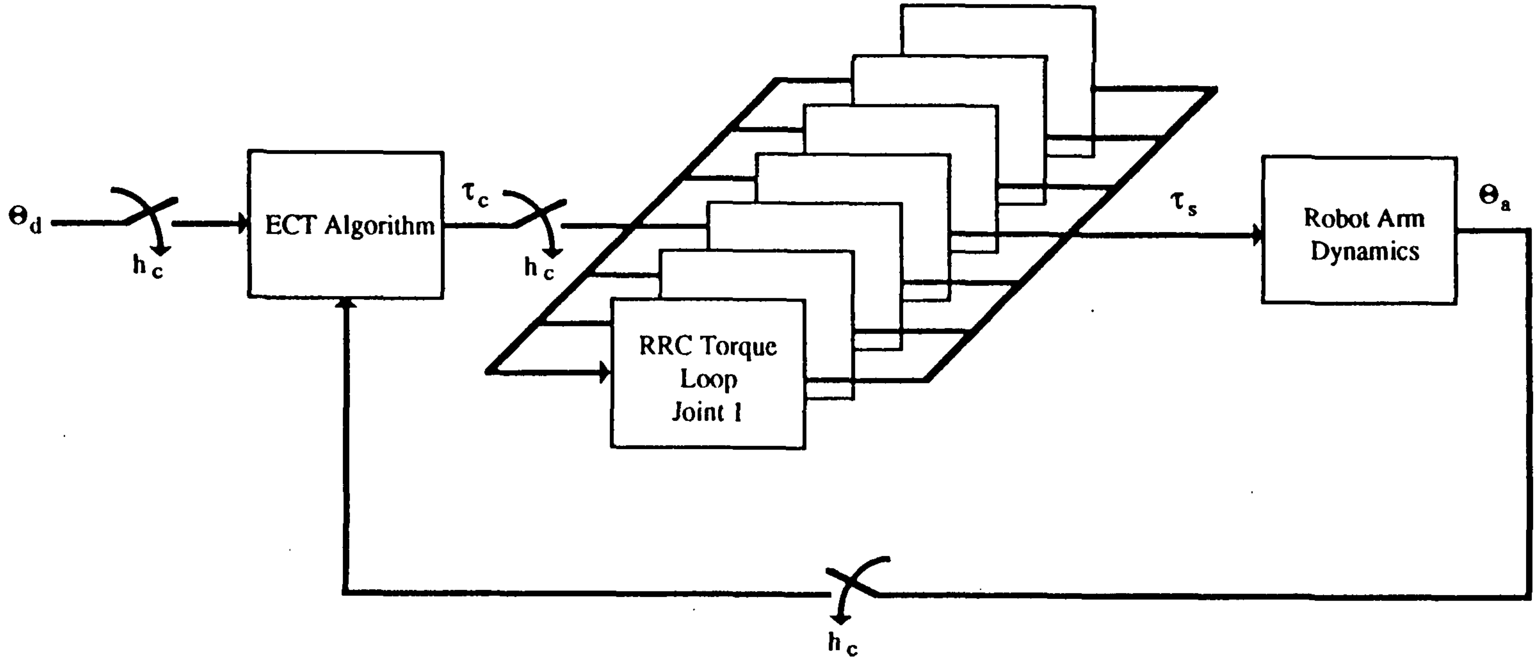

Figure 3.25: Equivalent computed torque controller

dynamics which are calculated using a recursive Newton-Euler formulation, as described in Craig [1986]. The RRC torque loop is built from analog hardware, and therefore, is a continuous time controller. The torque commands are given to the RRC torque loop through a DAC and a ZOH.

In order to compute the torque commands, the desired joint position, velocity, and acceleration must be given. If the path to be traversed is known a priori, a polynomial curve is typically fit to each joint trajectory. From this curve, the joint accelerations and velocities may be found by differentiation. However, for certain applications, particularly in teleoperation, the path is not known a priori. In this case, either the desired acceleration and velocity terms are set to zero, or they are approximated by finite difference schemes. In this study, the desired velocity and acceleration terms are found from the position commands using a backwards difference scheme, i.e.:

$$\dot{\Theta}_d{}^n = \frac{\Theta_d{}^n - \Theta_d{}^{n-1}}{h_c} \tag{3.58}$$

$$\ddot{\Theta}_d{}^n = \frac{\dot{\Theta}_d{}^n - \dot{\Theta}_d{}^{n-1}}{h_c} \tag{3.59}$$

where n is the time index and h_c is the command update period.

The diagonal gain matrices, K_p and K_d, are determined from:

$$K_p = k_p I \tag{3.60}$$

$$K_p = k_d I \tag{3.61}$$

where k_p and k_d are scalars and I is the identity matrix. Using this formulation, the ECT controller is completely specified by three parameters: the proportional gain, k_p, the derivative gain, k_d, and the command update period, h_c.

3.4.2 Computed Torque with Model Following Torque Loop

The computed torque controller to be used with the model following torque loops is given by Eq. (2.2). It is emphasized that only the link, and not the motor rotor or transmission, inertia and damping parameters are included in the $\hat{M}$ and $\hat{Q}$ terms.

In the case of the continuous model following torque loop, the two-level robot controller is implemented in the same manner as the ECT controller (see Fig. 3.25). However, at each joint the model following controllers are added to the existing RRC torque loops. The two sets of parameters, k_p, k_d, h_c, and ω_n, ζ, a, completely specify the dynamics of the computed torque algorithm and the continuous model following torque loops, respectively.

In the case of the discrete model following torque loop, the two-level robot controller is implemented as shown in the block diagram of Fig. 3.26. It consists of the computed torque controller, Eq. (2.2) and the discrete model following torque loops, Eq. (3.23). In this case, both the computed torque controller and the torque loops are discrete time controllers. Furthermore, the parameters of *R*, *S*, and

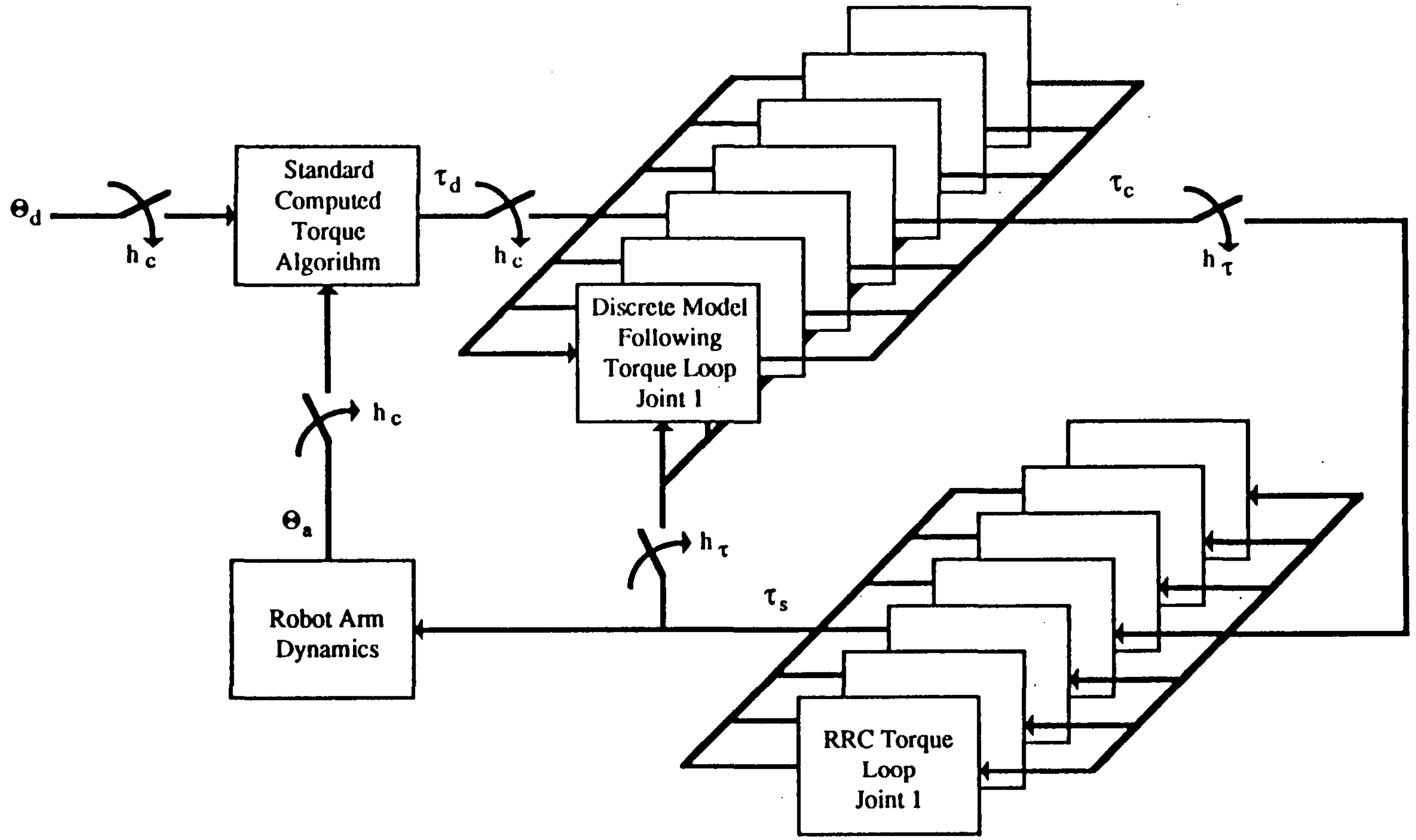

Figure 3.26: Computed torque controller with discrete model following torque loop

T are updated on-line at a specified rate. This study assumes that the torque commands and the parameters of the discrete model following torque loop are updated at the same time interval. This approach is chosen since the torque loop parameters depend on the terms of the inertia matrix which are, in turn, updated as part of the torque command calculation. It is emphasized that the sampling rate of the torque loop must be much higher than the update rate of the torque loop parameters. The two sets of parameters, k_p, k_d, h_c, and ω_n, ζ, a, and h_τ completely specify the dynamics of the computed torque algorithm and the discrete model following torque loops, respectively.

Chapter 4: Performance Evaluation of the Proposed Controllers Using Simulation

In this chapter, the three controllers presented in Chapter 3 are evaluated using simulation. The PD controller currently implemented on the RRC arm is utilized as the basis for comparison. The simulation package employed and the performance evaluation procedures are discussed. The results of the numerical simulations for several test cases are presented. Important factors studied in these tests include: i) the update rate of the joint angle and torque commands; ii) the presence of a known payload; iii) the use of inaccurate payload inertia parameters; iv) the exclusion of the desired acceleration and velocity terms from the computed torque algorithm.

4.1 Robot Arm Simulation Software

Simulations are used to evaluate the various control algorithms before implementing them on the actual hardware. The appropriate range of controller gains for satisfactory system performance is also determined from these simulation results.

A flowchart of the simulation software is shown in Fig. 4.1. A modified version of the RDS (Robot Dynamic Simulation) package [Chen, 1990] forms the core of the simulation software . It provides the dynamical equations of the robot arm and the appropriate numerical integration routines. Two additional programs are added to the RDS package. The first program computes the desired joint trajectories (see

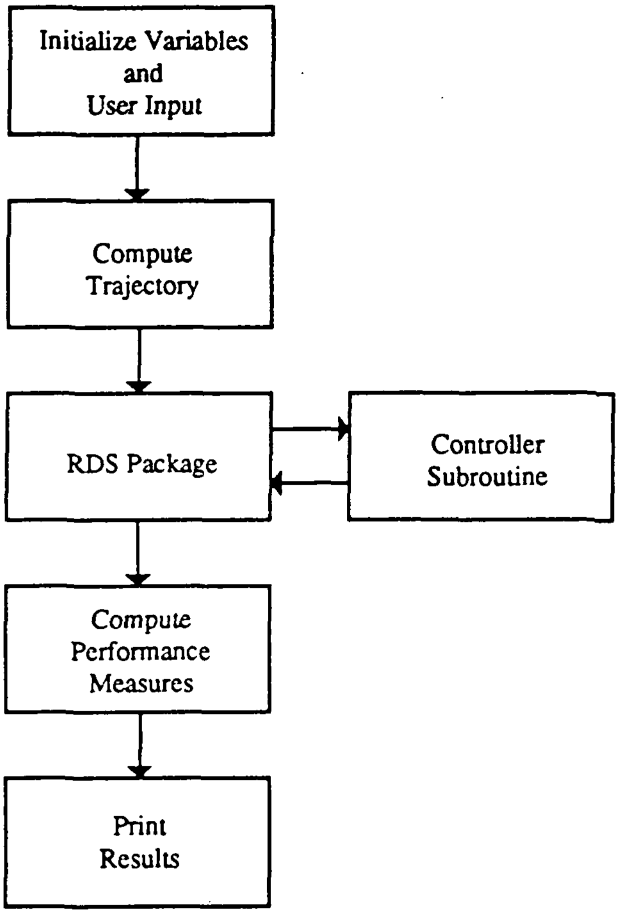

Figure 4.1: Flowchart of the simulation software

Section 4.2.1) and provides them to the RDS software. The second program computes the performance measures based on the actual joint angle trajectories returned from the RDS software.

RDS is an efficient simulation package for the analysis of open-chain manipulators with revolute joints that are driven by motors through speed reduction mechanisms. It utilizes the Macsyma symbolic manipulation program to generate the dynamic equations of motion of the robot manipulator. A highly efficient Adams-Bashford-Moulton integration routine from the Samsan numerical methods library performs the integration of the system of first order differential equations, which are generated by the symbolic manipulation program.

The various control algorithms under study are incorporated into the software through separate control subroutines. These subroutines are called, at a user defined rate, to compute the motor torque commands, given the dynamical state of the robot arm. Any controller states, such as the torque compensator states, which are to be integrated require their derivatives to be computed in the control subroutine and, subsequently, passed to the Adams-Bashford-Moulton integration routine. Detailed information on the RDS package may be found in Chen [1990].

4.2 Performance Tests

In evaluating the performance of a robot manipulator, the task should be carefully chosen to fully exercise the capabilities of the arm in a variety of configurations.

Furthermore, appropriate performance measures should be selected to accurately reflect the characteristics being evaluated. Currently, two standards have been proposed for evaluation of robot performance: ANSI/RIA R.15.05 and ISO/9283. The ANSI standard [1990] considers only the point-to-point and static characteristics of robot manipulators. The ISO standard [1990] provides tests for evaluating both static and dynamical robot performance. The ISO standard is used as the basis for the performance evaluation in this work.

4.2.1 Test Path

The ISO/9283 standard defines two paths for evaluating tracking performance; one is circular and the other square. Both paths lie in the diagonal plane of a cube, the edges of which are parallel to the base coordinate system (see Fig. 4.2). This cube is located within the portion of the manipulator workspace that has the greatest anticipated use. Furthermore, it occupies the maximum allowable volume.

In the current study, both tracking performance and cornering overshoot are evaluated. For tracking performance, the circular ISO path is chosen in order to avoid discontinuities in the desired velocity and acceleration terms of the computed torque equations. Such discontinuities result in large (theoretically infinite) torque commands, and therefore, should be avoided when evaluating tracking performance. The controller performance in the presence of velocity and acceleration discontinuities is considered by performing a cornering task and examining the resulting overshoot. To realize this task, one corner of the square path specified in ISO/9283 is used. A single path that allows for the evaluation of both tracking performance and cornering overshoot is chosen and is shown in Fig.4.2. The robot end-effector starts at the indicated point and performs the cornering task first. It then proceeds to track the circular path. The commanded orientation of the end-effector remains parallel to the base coordinate system for the entire path. For all test cases, the circle was centered at the point (0.0, -0.2, 1.2) m in the base coordinate system and had a radius of 0.5 m. The speed along the trajectory was selected to be 0.25 m/s. These parameters were found by trial and error in order to maximize both the radius of and the velocity around the circle, while remaining within joint position and velocity limits, respectively.

The test path is specified in Cartesian coordinates by a series of discrete points. Each point is computed given the radius of the circle, the speed along the trajectory (scalar speed), and the command update rate. Figure 4.3 shows the time history of the path projected along each axis of the base Cartesian coordinate system. The corresponding joint angles, Fig. 4.4, are found using a recursive, pseudoinverse Jacobian, inverse kinematics algorithm [Asada, 1986].

4.2.2 Performance Measures

Each section of the test path, cornering and circular, is comprised of a sequence of points. For the cornering section, the commanded path is comprised of l points, (x_{ci}, y_{ci}, z_{ci}), i = 1, 2, ... l, expressed with respect to the base coordinate system.

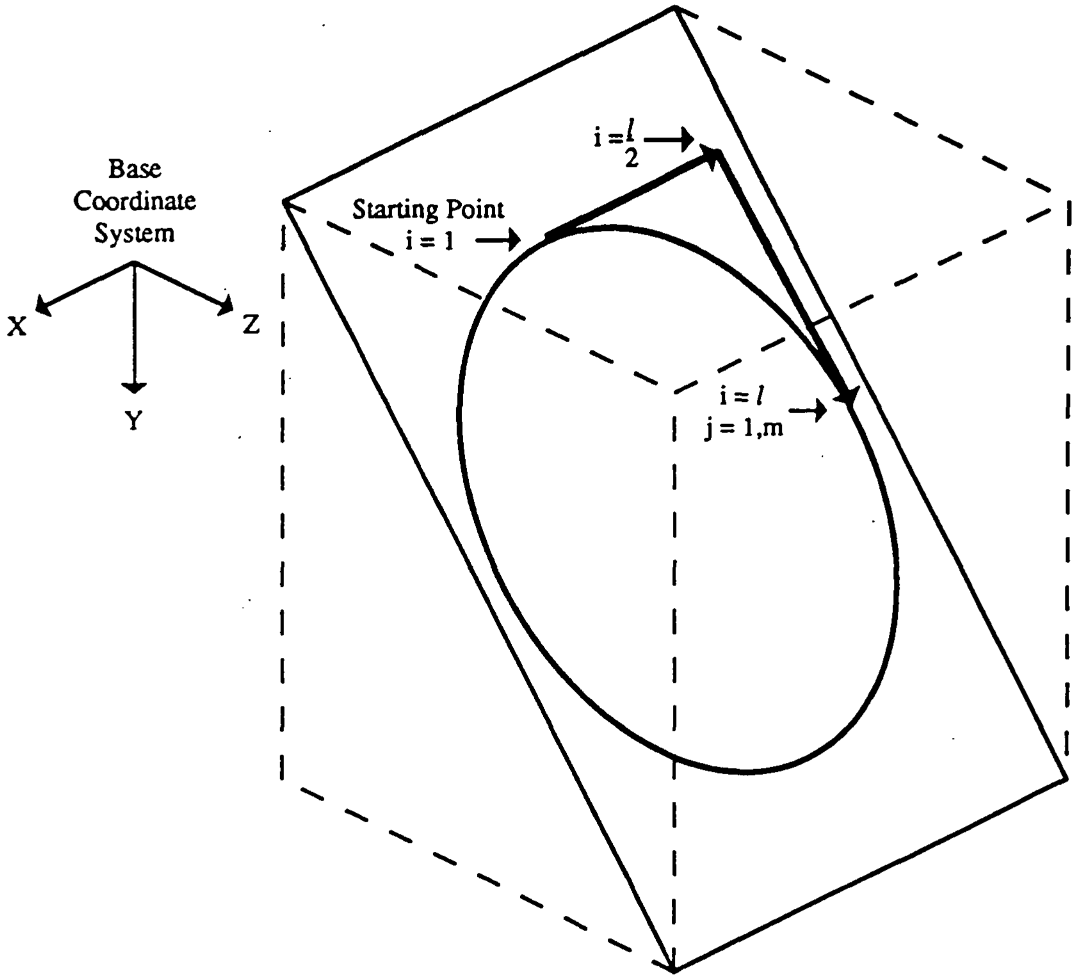

Figure 4.2: Standard test path

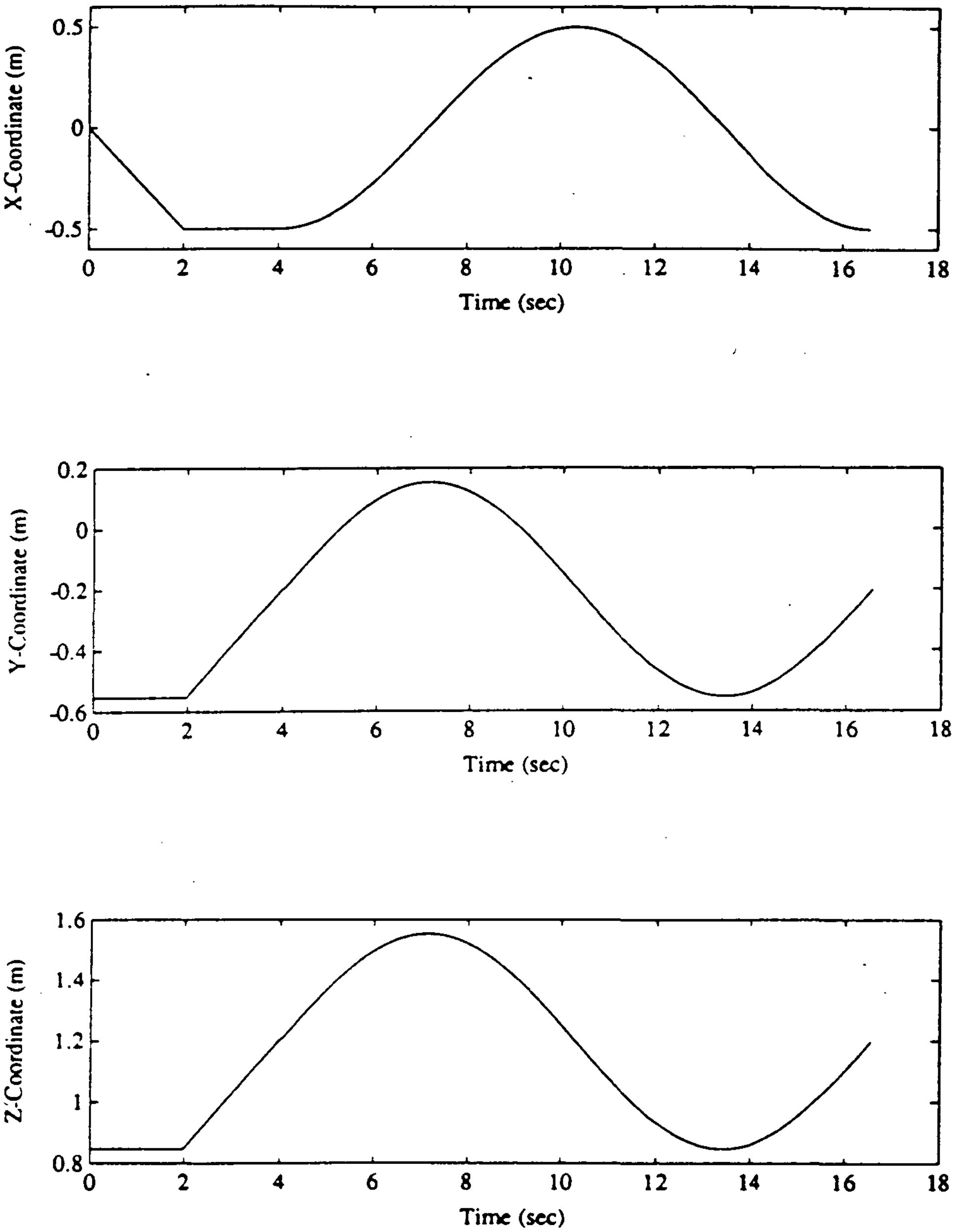

Figure 4.3: Test path trajectory of the end-effector in Cartesian space

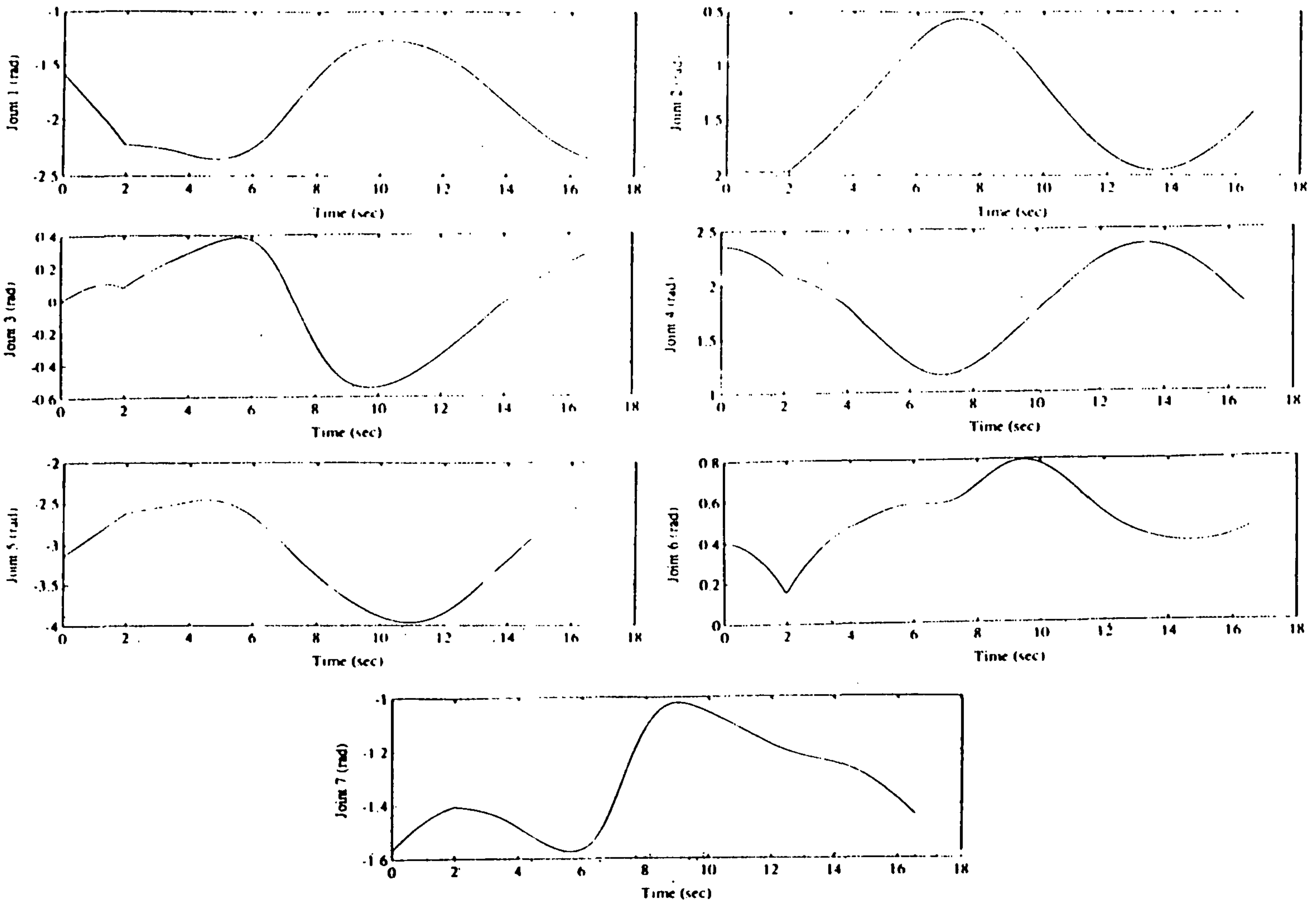

Figure 4.4: Test path trajectory in joint space

To quantify the Cornering Overshoot (CO), the following equation is proposed by the ISO standard:

$$CO = \max_{i = \frac{l}{2}}^{l} \left\{ \left[(x_{ci} - x_i)^2 + (y_{ci} - y_i)^2 + (z_{ci} - z_i)^2 \right]^{\frac{1}{2}} \right\} \qquad (4.1)$$

where (x_i, y_i, z_i), $i = 1, 2, \ldots l$, are the Cartesian coordinates of the actual path and $\frac{l}{2}$ is the index corresponding to the corner point (see Fig. 4.2). Therefore, the cornering overshoot is the maximum deviation from the commanded path after the robot has passed the corner point of the task.

The circular section is comprised of m commanded points, (x_{cj}, y_{cj}, z_{cj}), $j = 1, 2, \ldots m$, expressed with respect to the base coordinate system. This section is executed n times. Path accuracy is defined from:

$$\text{Path Accuracy} = \mathcal{L}_{\infty}(T) = \max_{j=1}^{m} \left\{ \left[(x_{cj} - \bar{x}_j)^2 + (y_{cj} - \bar{y}_j)^2 + (z_{cj} - \bar{z}_j)^2 \right]^{\frac{1}{2}} \right\}$$

(4.2)

where $(\bar{x}_j, \bar{y}_j, \bar{z}_j)$, $j = 1, 2, \ldots m$, are the mean values of the j^{th} point coordinates for the n repetitions of the path, i.e.:

$$\bar{x}_j = \frac{1}{n} \sum_{k=1}^{n} x_{jk} \qquad \bar{y}_j = \frac{1}{n} \sum_{k=1}^{n} y_{jk} \qquad \bar{z}_j = \frac{1}{n} \sum_{k=1}^{n} z_{jk} \qquad (4.3)$$

and (x_{jk}, y_{jk}, z_{jk}) is the j^{th} point of the actual path during the k^{th} repetition. Thus, path accuracy is the maximum deviation of the commanded path from the "mean" actual path.

Equation (4.2) defines path accuracy through the $\mathcal{L}_\infty$-norm (i.e. the maximum) of the translational error. Another common norm used to quantify performance is the $\mathcal{L}_2$-norm, which is defined from:

$$\mathcal{L}_2(T) = \left(\frac{1}{m} \sum_{j=1}^{m} (x_{cj} - \bar{x}_j)^2 + (y_{cj} - \bar{y}_j)^2 + (z_{cj} - \bar{z}_j)^2 \right)^{\frac{1}{2}} \qquad (4.4)$$

Although this measure is not proposed by the ISO standard, it will be adopted in this work, since it represents the root mean square translational error along the entire trajectory.

To completely define the end-effector position, both translational and orientational information is required. One method of representing the orientation of a given frame with respect to a reference frame is to use an "equivalent angle-axis" [Craig, 1986]. This representation describes the axis of rotation, K, about which the reference frame must be rotated by an angle, α, to become parallel to the frame under consideration (see Fig. 4.5). In this study, the measure of orientation error is the magnitude of the rotation angle, α, between the actual end-effector orientation

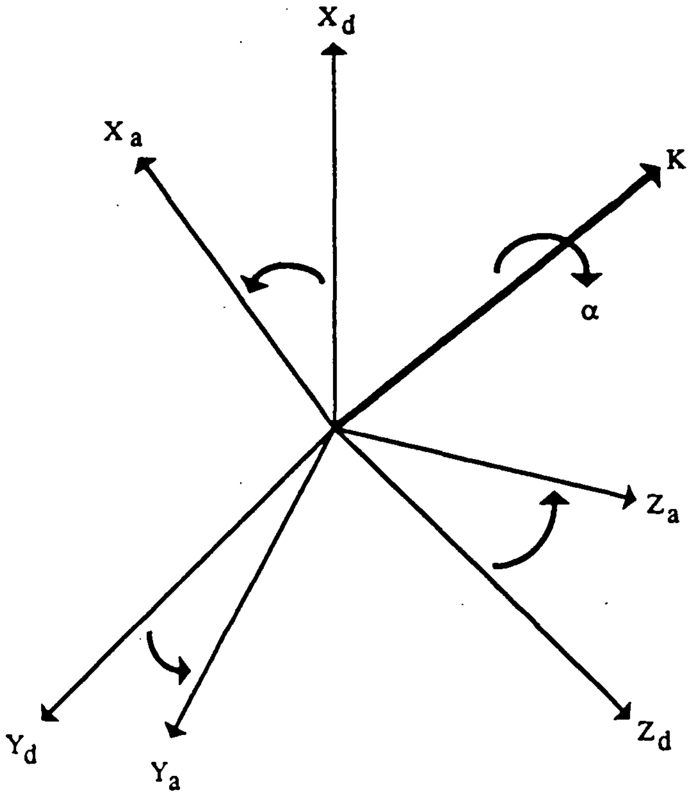

Figure 4.5: Equivalent angle-axis representation

and the commanded orientation.[†] This quantity gives an indication of the error in orientation without regard to the direction of rotation.

Measures of orientation error which are analogous to the translation error measures can now be defined. The $\mathcal{L}_\infty$ orientation error is given by:

$$\mathcal{L}_\infty(O) = \max_{j=1}^{m} \left\{ \left| \bar{\alpha}_j \right| \right\} \tag{4.5}$$

where $\bar{\alpha}_j$, j = 1, 2, ... m, is the mean value of the angle of rotation corresponding to the j^{th} point for the n repetitions, i.e.:

$$\bar{\alpha}_j = \frac{1}{n} \sum_{k=1}^{n} \alpha_{jk} \tag{4.6}$$

α_{jk} is the angle of rotation corresponding to the j^{th} point during the k^{th} repetition. Furthermore, the $\mathcal{L}_2$ orientation error is given by:

$$\mathcal{L}_2(O) = \left(\frac{1}{m} \sum_{j=1}^{m} \bar{\alpha}_j^{\,2} \right)^{\frac{1}{2}} \tag{4.7}$$

[†] This measure was also suggested by J. J. Craig during a private conversation on January 17, 1991.

4.3 Test Cases and Simulation Results

Four controllers are evaluated through simulation: the PD controller currently implemented on the RRC arm, the Equivalent Computed Torque controller (ECT), the Computed Torque controller with Continuous Model Following torque loop (CTCMF), and the Computed Torque controller with Discrete Model Following torque loop (CTDMF). All parameters of the robot arm are assumed to be known in the simulations. Several test cases are considered to examine the effects of: i) the update rate of the joint angle and torque commands; ii) the presence of a known payload; iii) the use of inaccurate payload inertia parameters; iv) the exclusion of the desired acceleration and velocity terms from the computed torque algorithm.

4.3.1 Baseline Test Case

Table 4.1 presents the controller parameters for the baseline test case. The command update rate, $\frac{1}{h_c}$, was selected to be 40 Hz for all controllers since this is the maximum rate that may be used with the actual robot hardware. This limit is due to the time period required to compute the inverse robot dynamics. For the computed torque controllers, k_d was chosen to correspond to a damping ratio of 0.707 in Eq. (2.3), and k_p was chosen through simulations by increasing its value until substantial improvement in performance could no longer be realized. For the PD controller, k_p was chosen from tests on the actual robot arm to be the highest gain for which the arm remained stable in all test cases. It is emphasized that k_p has significantly different physical interpretation between the PD and the computed

Table 4.1: Controller parameters for the baseline test case

	PD	ECT	CTCMF	CTDMF
Payload	none	none	none	none
k_p	6 (sec^{-1})	200 (sec^{-2})	200 (sec^{-2})	200 (sec^{-2})
k_d (sec^{-1})	NA	20	20	20
h_c (sec)	0.025	0.025	0.025	0.025
ω_n (rad/sec)	NA	NA	400	300
ζ	NA	NA	1	1
a (rad/sec)	NA	NA	800	600
h_τ (sec)	NA	NA	NA	0.001

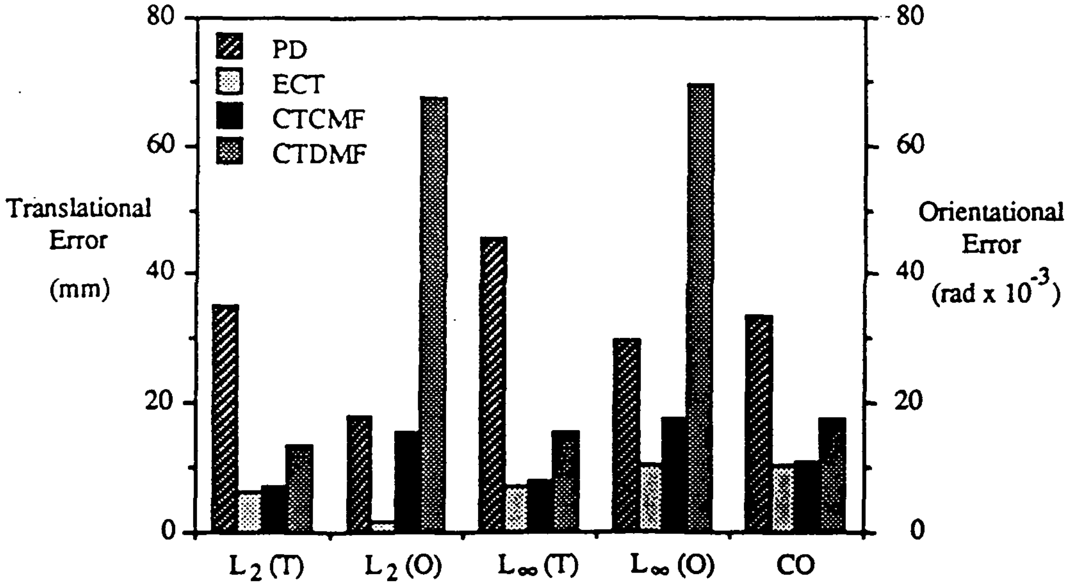

Figure 4.6: Baseline Test Case Results

torque controllers (see Sections 2.1.1 and 3.1). This fact is shown by the different units of k_p corresponding to these two cases. The model following controller parameters are chosen under the guidelines discussed in Section 3.3.

Figure 4.6 shows the values of the performance measures obtained from the baseline test simulation. The ECT controller has the best performance in all categories, while the CTCMF controller has the next best. The CTDMF controller performs well in terms of translational error and cornering overshoot, but performs poorly in terms of orientational error. With this one exception, all computed torque controllers show improved performance over the PD controller. Improvement ranges between 10% and 90%.

To further study the poor performance of the CTDMF controller with respect to orientation, the actual time histories of the joint angles are shown in Fig. 4.7. The figure shows that all joints perform well with the exception of the last two, namely joints 6 and 7. Although the errors at these joints do not contribute substantially to the translational error, due to their short link lengths, they contribute greatly to the orientational error.

The poor performance of joints 6 and 7 can be attributed to the fact that, at these joints, the discrete model following controller is not robust with respect to disturbance torques, τ_{dist} (see Fig. 3.18). It is noted that these disturbances result from the coupling of joint motions and the gravity loading (see Section 3.4.1). Figure 4.8 shows the output, τ_s, of the discrete model following torque loops at joints 1,6, and 7 for a unit step in disturbance torque, τ_{dist}. The figure indicates

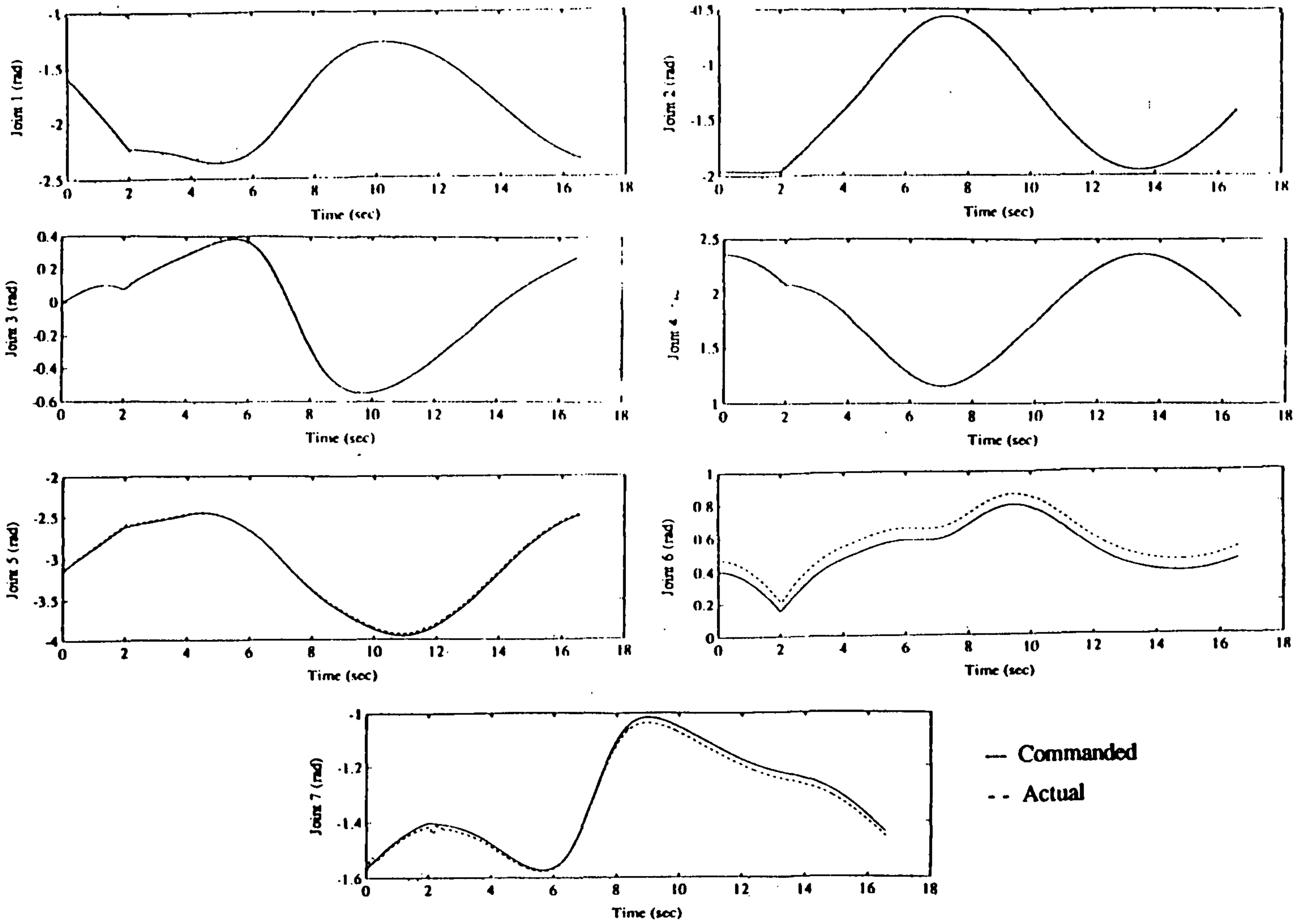

Figure 4.7: Joint trajectories of the CTDMF controller for the baseline test case

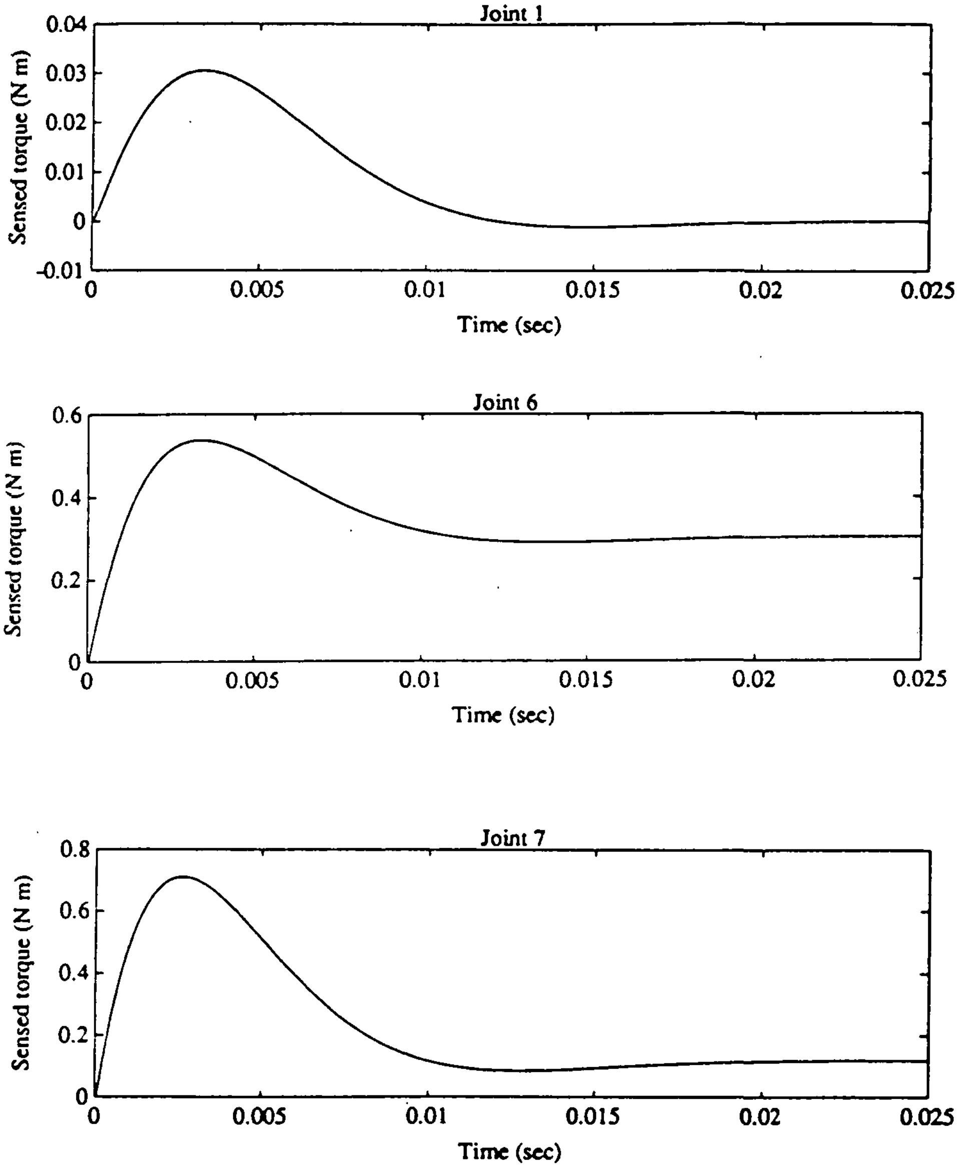

Figure 4.8: Response of discrete model following torque loop to step in disturbance torque

that the discrete model following controller of joint 1 is very robust with respect to the disturbance input since τ_s quickly converges to zero. However, joints 6 and 7 show a significant steady state output torque. This result indicates that the discrete model following controllers of these joints are not able to accurately regulate the output torque in the presence of the inevitable disturbances, and therefore, the performance of the CTDMF controller is degraded (see also Section 2.1.2).

4.3.2 Effects of Update Rate

The second test case considers the effect of the joint angle and torque command update rate, $\frac{1}{h_c}$, on performance. Since the commanded path is specified as a sequence of discrete points, the closeness of this sequence to the actual path is highly dependent on the rate at which the points are provided to the controller. Furthermore, since the velocity and acceleration commands for the computed torque controller are approximated from the position commands through a backwards difference scheme, the command update rate may have significant effects on the accuracy of these approximations.

The command update rate for this test case was increased to 100 Hz. All other controller parameters are identical to those used in the baseline test case. Figure 4.9 displays the results obtained from the simulations in this test case. The relative performance of each of the controllers is similar to the baseline test case, with the ECT and CTCMF controllers showing the best performance. Although, the performance of the PD and CTDMF controllers remain at approximately the same

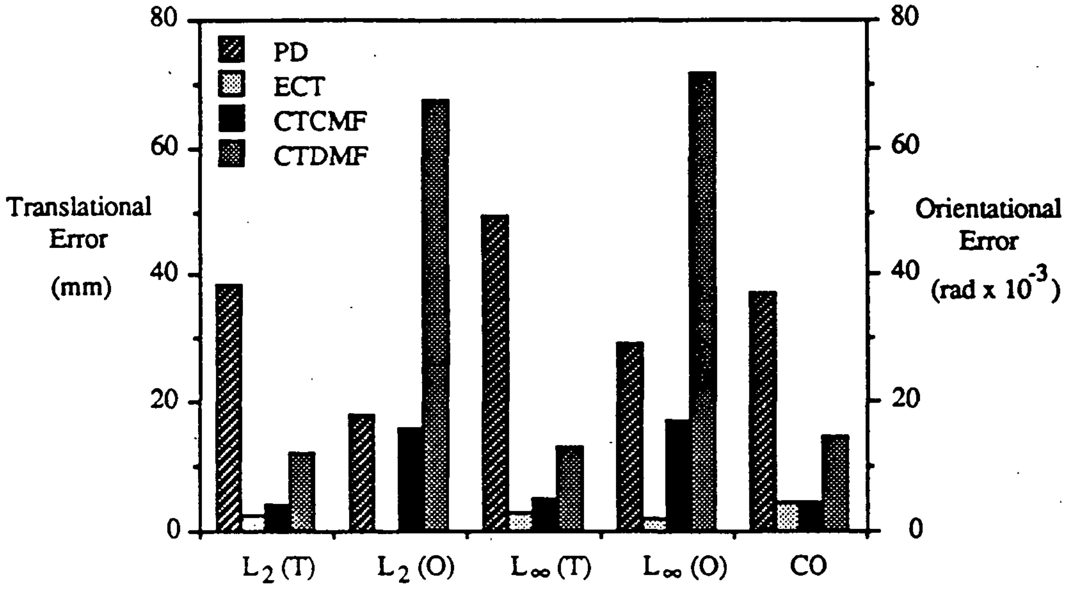

Figure 4.9: Controller comparison at 100 Hz update rate

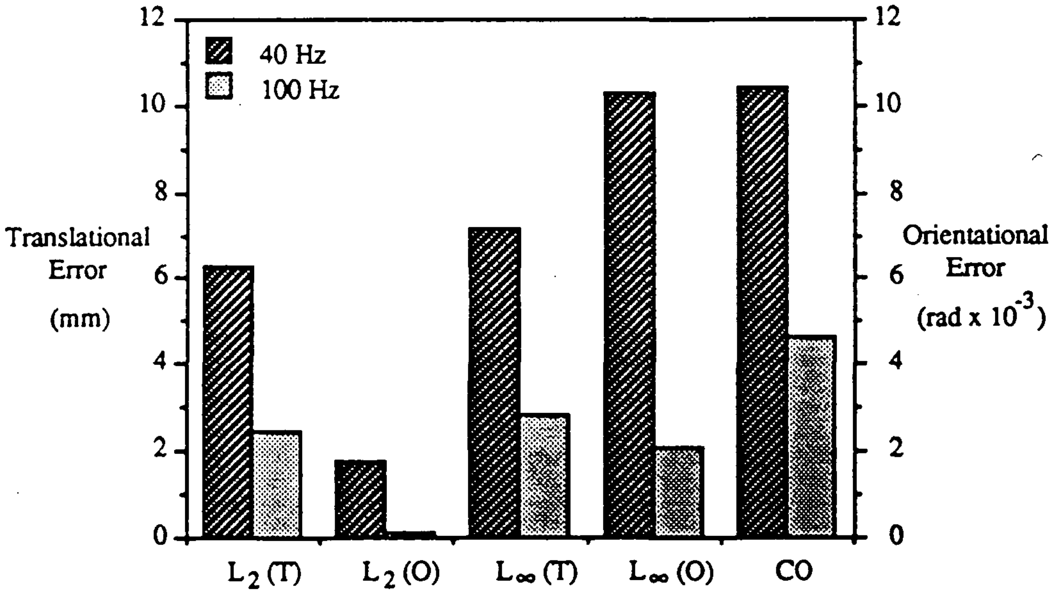

Figure 4.10: Effect of update rate on the performance of the ECT controller

levels, the performance of the ECT and CTCMF controllers improves as the rate is increased from 40 to 100 Hz.

Figure 4.10 compares the performance of the ECT controller for the 40 and 100 Hz cases. Improvement is seen in all performance measures and ranges between 60% and 90%. The improved performance results from the fact that: i) The desired velocity and acceleration commands are more accurately approximated by the backwards difference scheme in the 100 Hz case; ii) The torque commands are updated more frequently in this case, thus better reflecting the true torque requirements of the arm.

Figure 4.11 compares the performance of the CTCMF controller for the 40 and 100 Hz cases. This controller shows improved performance only in the translational and cornering overshoot measures. Improved performance is due to the same factors discussed above. As seen with the CTDMF controller, joints 6 and 7 have the poorest tracking performance, indicating that the continuous model following controller is also not robust with respect to disturbance torques.

4.3.3 Effects of Payload and Inaccurate Payload Inertia Parameters

The ability to incorporate the payload inertia into the controller equations is a distinct advantage of the computed torque controllers over the fixed gain PD controller. This test case examines the effect of the payload on performance.

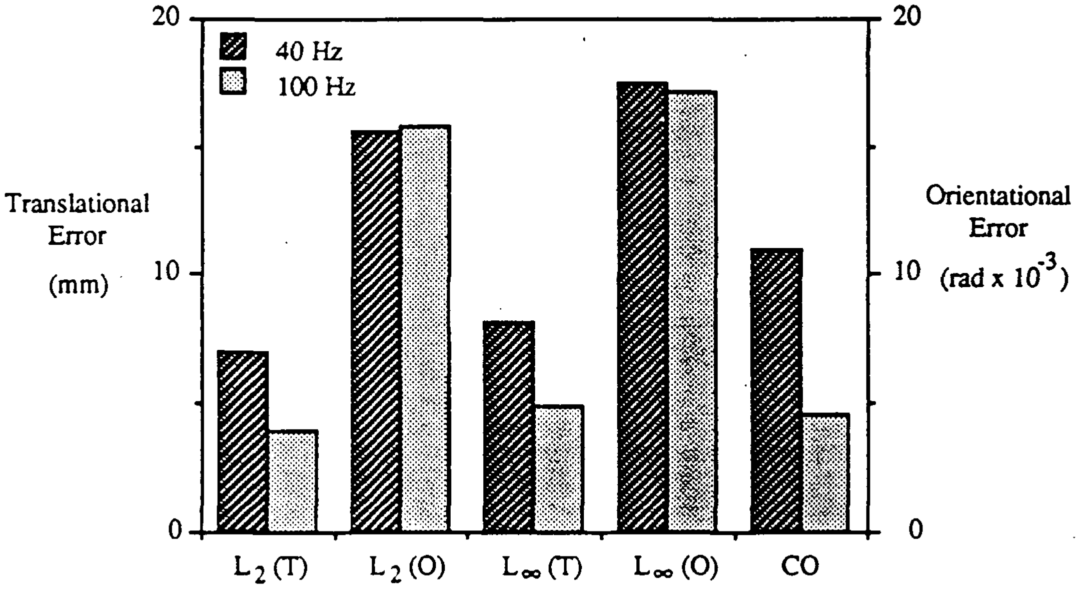

Figure 4.11: Effect of update rate on the performance of the CTCMF controller

Furthermore, since uncertainty in the parameter values of the dynamical model may have a significant impact on the performance of the computed torque controllers, the effect of using inaccurate payload parameters is also examined. To this end, the inertial properties of the payload are under- and over-estimated by 25%. The payload considered is a 1.2 m long hollow steel beam with rectangular cross-section, 51 x 76 mm outside, and wall thickness of 6.5 mm. Its mass is 12.2 kg and the corresponding inertia matrix, with respect to the end effector coordinate system, is given by:

$$I = \begin{bmatrix} 1.5 & 0.0 & 0.0 \\ 0.0 & 0.12 & 0.0 \\ 0.0 & 0.0 & 1.5 \end{bmatrix} \text{kg m}^2$$

These values correspond to approximately half of the payload capacity of the RRC arm. All controller parameters in this test are equal to those used in the baseline test case.

Figure 4.12 displays the simulation results obtained when the exact payload values are used in the computed torque equations. The ECT controller shows the best performance for all measures. Furthermore, the CTDMF controller performs equally well in terms of translational error and cornering overshoot. It is also noted that the CTDMF controller shows better performance than the CTCMF controller in all categories.

It is of interest to examine the performance of each controller separately. Figure 4.13 shows the performance measures obtained for the PD controller with and without the payload. It indicates that the performance of the PD controller

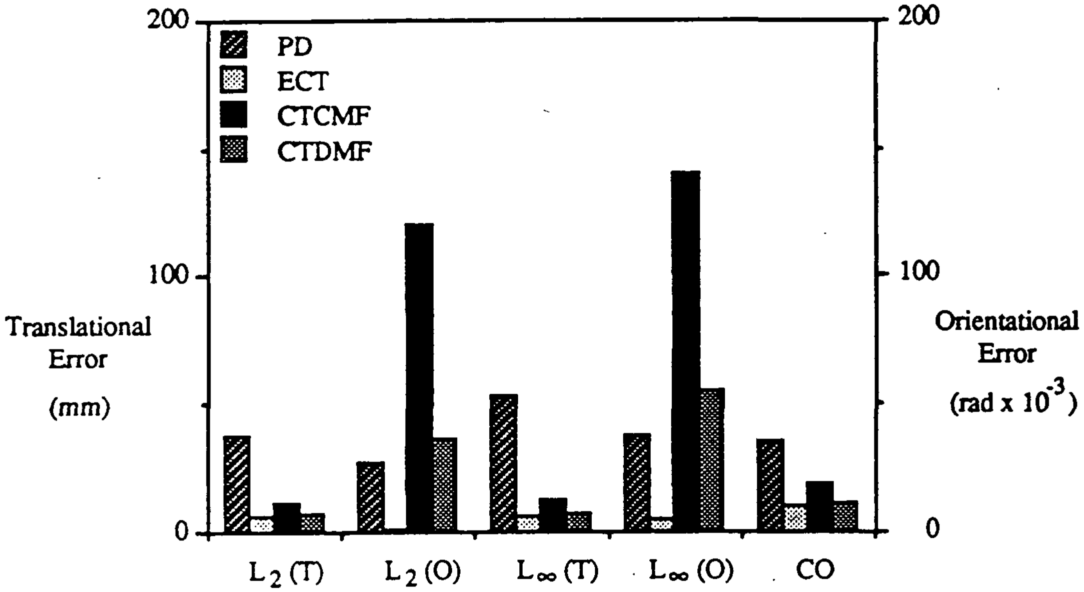

Figure 4.12: Controller comparison with known payload

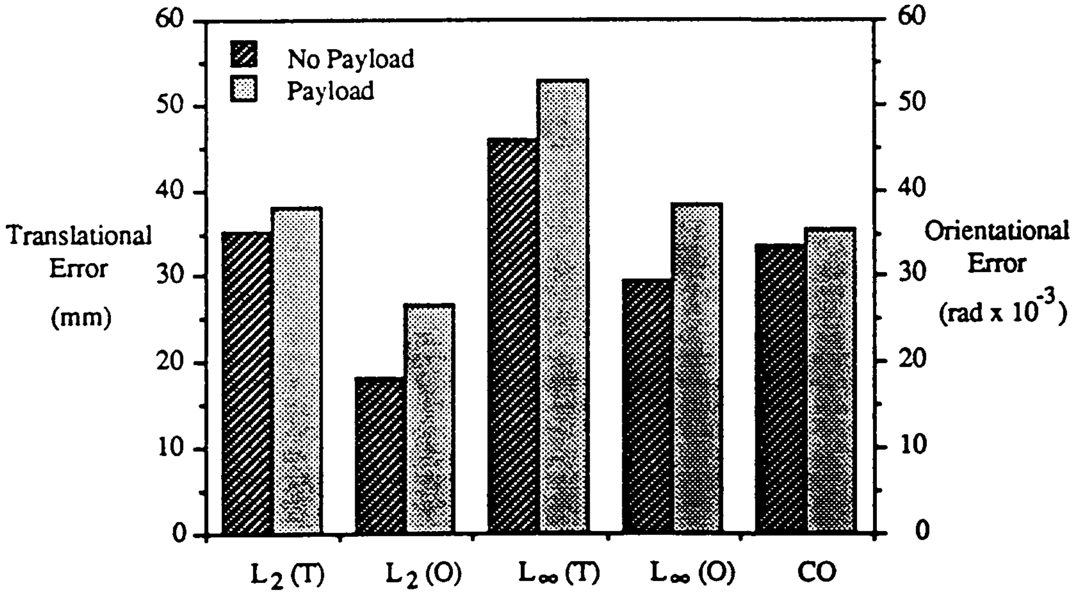

Figure 4.13: Effect of payload on the performance of the PD controller

decreases with the addition of the payload. This is primarily due to increased gravity loading of the robot arm. It is noted that higher values of the proportional gains, which correspond to a stable system when no payload was present, resulted in severe oscillations in the joint positions when the payload was added. Since the payload parameters are not utilized in the PD controller, the under- and over-estimated payload tests have no meaning for this controller.

Figure 4.14 shows the performance measures obtained for the ECT controller in the following cases: i) no payload; ii) known payload; iii) under-estimated payload inertia characteristics by 25%; iv) over-estimated payload inertia characteristics by 25%. The performance of the ECT controller remains approximately unchanged, as expected. The figure also shows degradation in performance when inaccurate payload estimates are used.

Figure 4.15 shows the performance measures obtained for the CTCMF controller under the same conditions. The performance of the system degrades significantly with the addition of the payload, particularly in terms of the orientational error. The poor performance is due to the fact that the parameters of the continuous model following torque loop are fixed. It is also noted that the controller is designed based on the inertia characteristics of the unloaded arm. However, the link inertias change significantly with the addition of the payload, and therefore, the model following controller parameters are not the appropriate ones for the loaded case.

Figure 4.16 shows the performance measures obtained for the CTDMF controller under the four payload conditions. The performance improves slightly with the

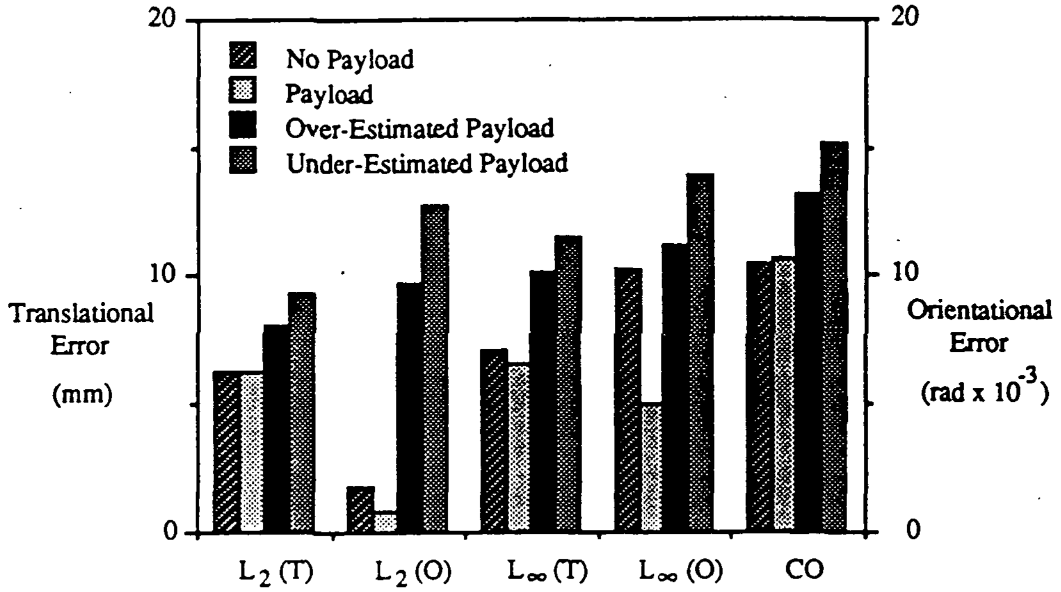

Figure 4.14: Effect of payload and inaccurate payload inertia estimates on the performance of the ECT controller

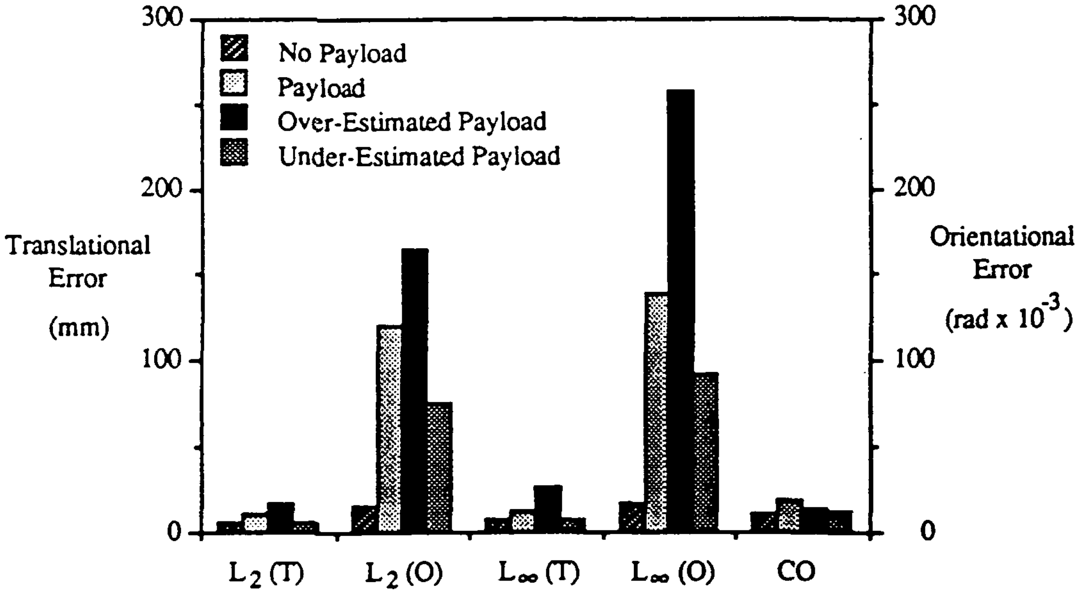

Figure 4.15: Effect of payload and inaccurate payload inertia estimates on the performance of the CTCMF controller

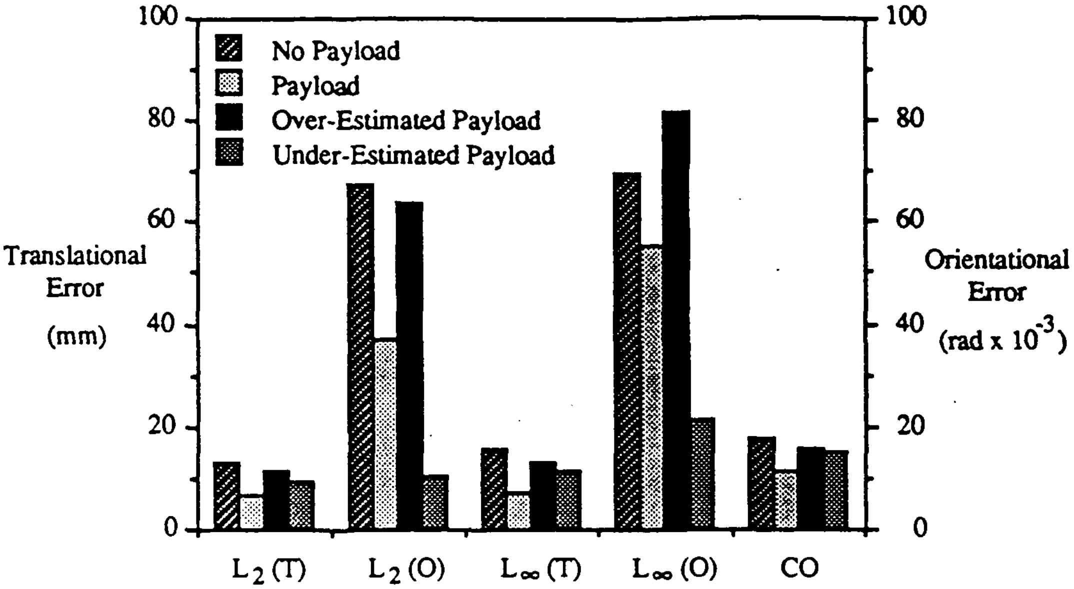

Figure 4.16: Effect of payload and inaccurate payload inertia estimates on the performance of the CTDMF controller

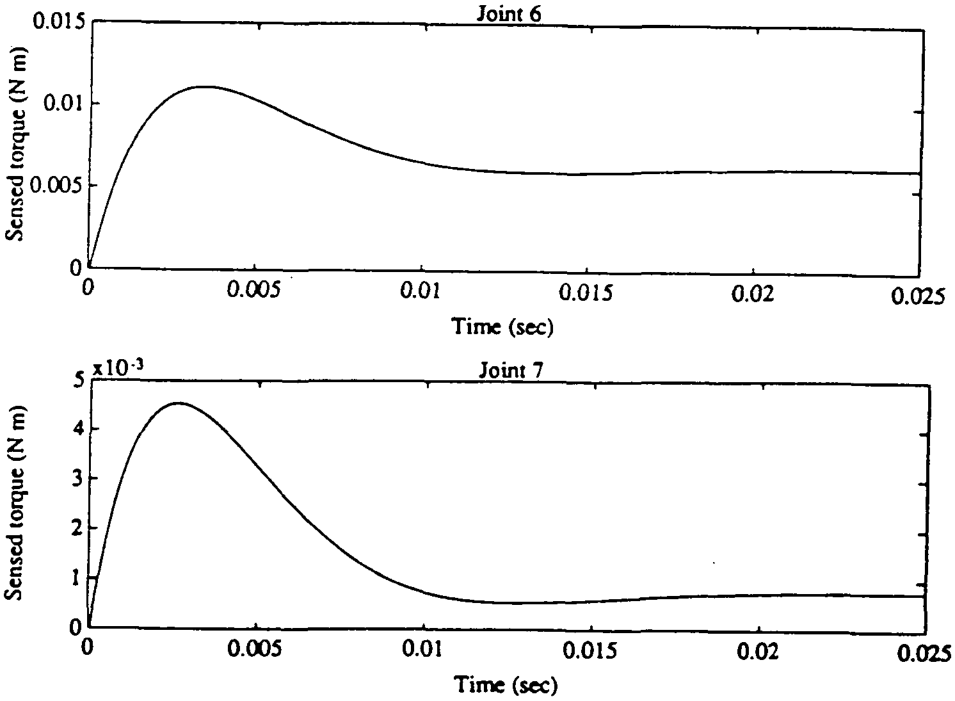

Figure 4.17: Response of discrete model following torque loop to step in disturbance torque with payload present

addition of the payload, even if the payload is over or under-estimated. This improvement is due to the enhanced robustness of the torque loops at the last joints when a payload is present. Figure 4.17 shows the output, τ_s, of the discrete model following torque loop for a unit step in disturbance torque, τ_{dist}, at joints 6 and 7 when the payload is present. A comparison with Fig. 4.15 shows that the torque loop is much more robust with respect to the disturbance torque in this case.

4.3.4 Effects of Eliminating the Desired Acceleration and Velocity Terms from the Computed Torque Equations

The last test case is used to examine the effect of excluding the desired acceleration and velocity terms from the computed torque equations. This case is of interest since these terms are approximated by a backwards difference scheme. This approach was considered for the reasons discussed in Section 3.4.1. All controller parameters are equal to those used in the baseline test case.

Figures 4.18, 4.19, and 4.20 show the performance measures corresponding to the ECT, CTCMF, and CTDMF controllers for this test case. Small changes in translational and orientational performance are observed when the desired acceleration terms are excluded. However, there is a significant effect in translational performance when the desired velocity terms are eliminated. The cornering overshoot also increases as the two terms are eliminated.

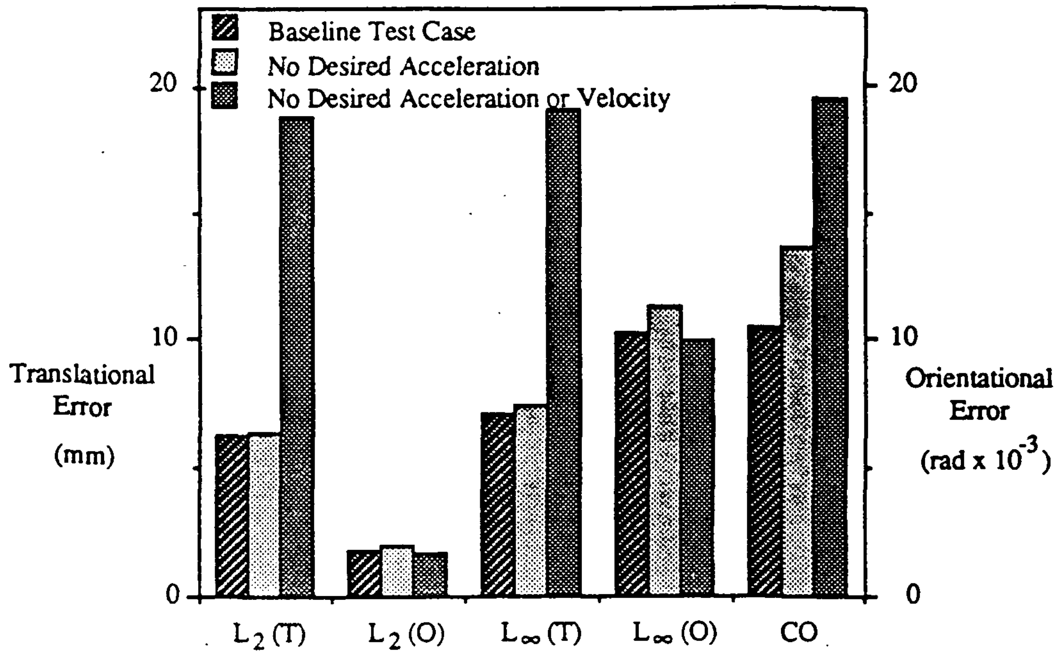

Figure 4.18: Effect of eliminating the desired acceleration and velocity terms from the ECT controller

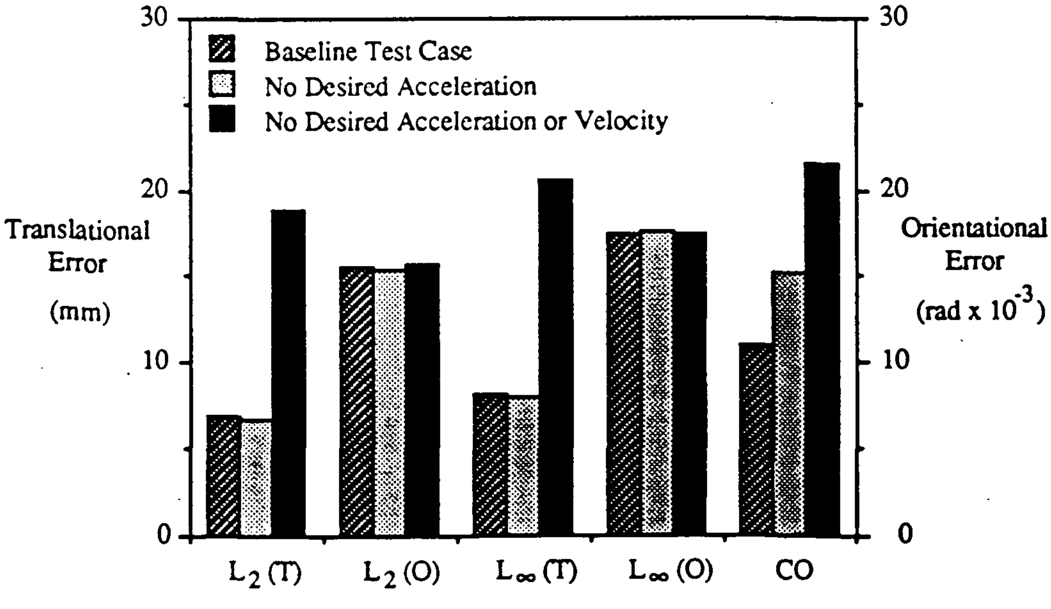

Figure 4.19: Effect of eliminating the desired acceleration and velocity terms from the CTCMF controller

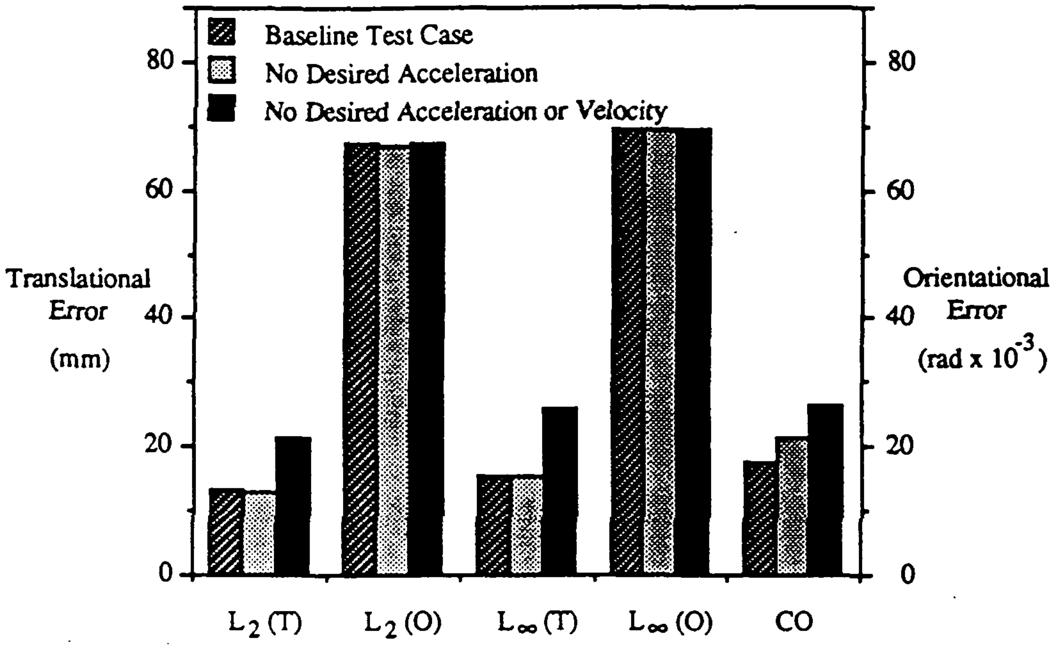

Figure 4.20: Effect of eliminating the desired acceleration and velocity terms from the CTDMF controller

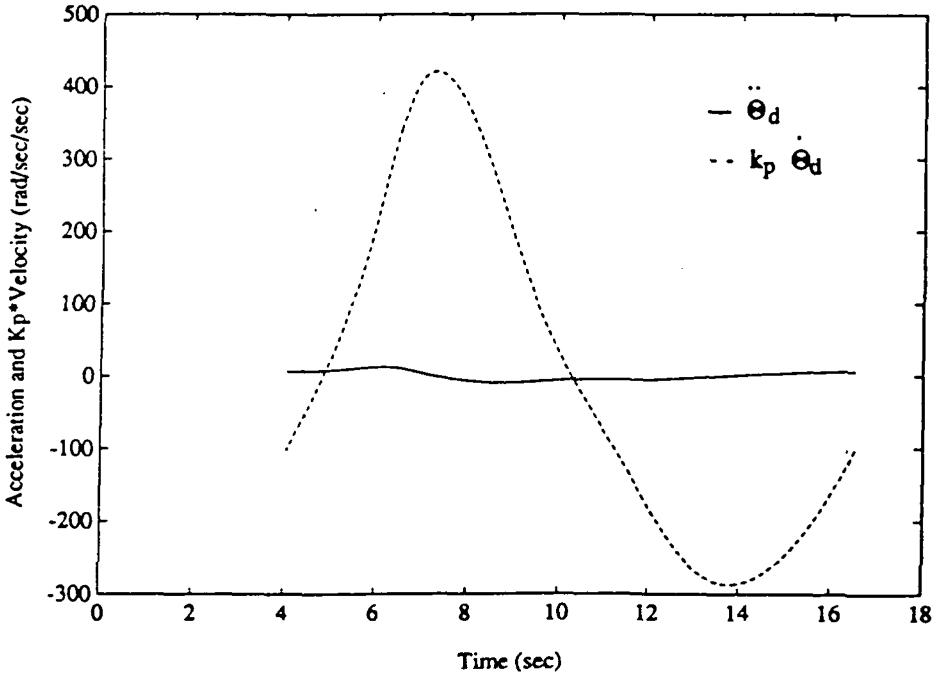

Figure 4.21: Contributions of the desired acceleration and velocity terms

The fact that the desired acceleration terms have no significant effect on the performance suggests that either these terms are small for the given trajectory or that they are not well approximated by the backwards difference scheme at this sampling rate. Figure 4.21 compares the contributions of the desired acceleration and velocity terms as they appear in the computed torque equations (Eq. (2.2) and Eq. (3.57)), i.e. $\ddot{\Theta}_d$ versus $k_p \dot{\Theta}_d$. The values correspond to joint 1 over the circular section of the test path. The figure indicates that the desired velocity term is much more significant than the acceleration term for this joint.

Chapter 5: Experimental Performance Evaluation of the ECT Controller

The simulation tests of Chapter 4 showed that the ECT controller had the best overall performance with respect to all performance measures. Furthermore, this controller is easily implemented with the existing hardware. Therefore, the ECT controller is experimentally studied using the actual robot system. The existing PD controller is again used as the basis for comparison. The experimental set-up and the results obtained are presented in this chapter. The experimental and simulation results are also compared.

5.1 Experimental Set-up

The experimental set-up is shown in Fig. 5.1 and consists of the RRC 1607HP mechanical and electrical hardware and an Intel Multibus I bucket equipped with an Intel 386/387 microprocessor board. The control testing software is downloaded to this board from a Micro-VAX computer. In the case of the ECT controller, the desired joint angles, angular velocities, and angular accelerations are updated at each command cycle. The actual joint angles and angular velocities are measured by the resolvers and are converted to digital signals by an ADC board in the Multibus I bucket. These values are used by the control software that implements the ECT algorithm to compute the torque commands. The torque commands are subsequently passed to the analog torque loop through the DACs of the RRC hardware. A similar procedure is followed for the PD controller. In this case,

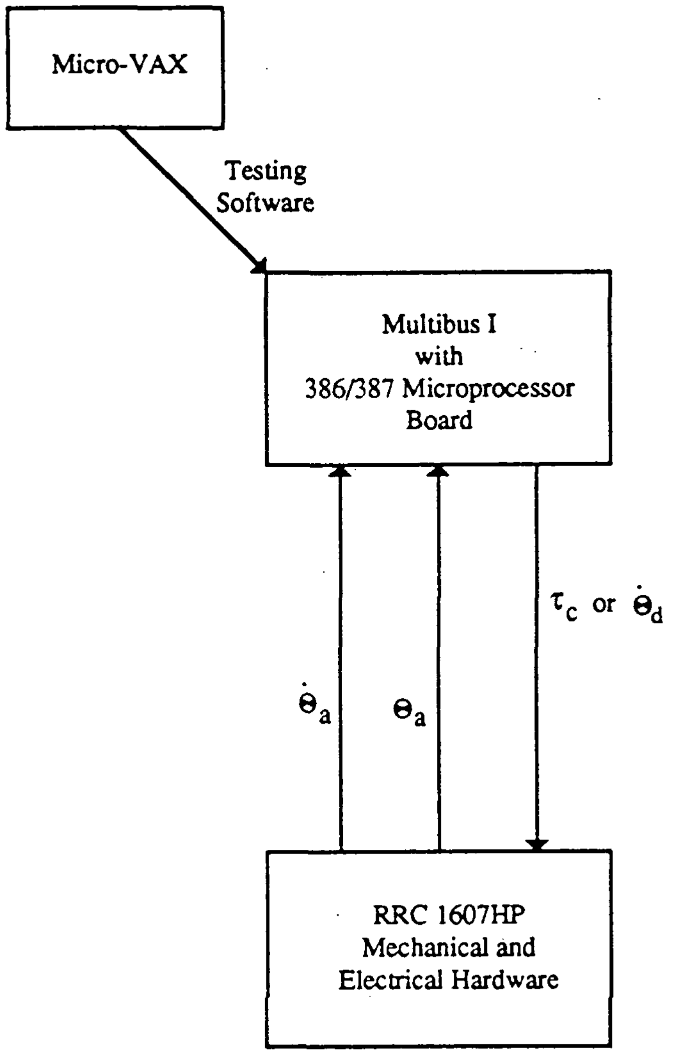
Micro-VAX
Testing Software
Multibus I with 386/387 Microprocessor Board
$\dot{\Theta}_a$
Θ_a
τ_c or $\dot{\Theta}_d$
RRC 1607HP Mechanical and Electrical Hardware

Figure 5.1: Experimental set-up

however, only the desired joint angles are updated. The PD controller subsequently computes the velocity commands for the analog velocity loop (see Section 3.1).

The architecture of the testing software is shown in Fig. 5.2. Prior to the execution of any tests, the arm must be brought to a reference, or home, position. This procedure is necessary since the resolver counts are measured with respect to this known position. The user specifies which measurements are to be saved in the memory of the microprocessor board for further processing. Joint angles, angular velocities, or the end-effector position may be selected. The end-effector position is computed from the actual joint angle measurements using the robot forward kinematics algorithm. The user must select the controller to be tested and specify the appropriate parameters.

After the user inputs have been properly completed, the test is executed. The points along the standard path, shown in Fig. 4.2, are computed first. The arm then moves from its current position to the starting position, where it remains for 5 seconds for all transients to settle. The path is executed by updating the desired joint positions at the specified time interval. The torque or velocity commands are calculated from these desired positions, as described above. The actual data used to compute these commands are saved in the memory of the microprocessor board. After the test has been completed, the performance measures are evaluated from these data and displayed. The raw data may also be transferred to the Micro-VAX computer for further processing.

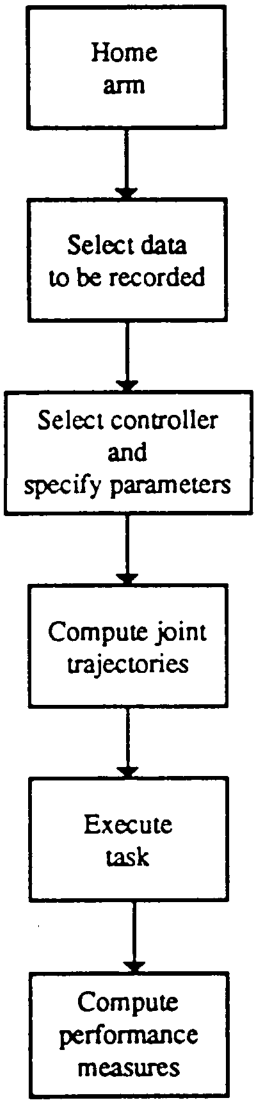
Home arm
Select data to be recorded
Select controller and specify parameters
Compute joint trajectories
Execute task
Compute performance measures

Figure 5.2: Flowchart of the testing software

Several important safety features are incorporated into the software. In order to avoid physical damage to the robot arm, position, velocity, and torque limits are specified for each joint. The control algorithm checks both the commanded and actual values against these limits before issuing the commands to the joint controllers. If a commanded or actual value is out of range, the arm is immediately disabled and an appropriate message is displayed.

5.2 Test Cases and Experimental Results

Those test cases which are examined in the simulation tests are also studied experimentally, with the exception of the update rate case. It is noted that the update rate could not be increased beyond 40 Hz due to hardware limitations. In the experimental tests, the $\mathcal{L}_2$ and $\mathcal{L}_\infty$ performance measures are computed using 5 repetitions of the circular section, as specified by the ISO standard (see Section 4.2.2).

5.2.1 Baseline Test Case

The controller parameters for the baseline test case are identical to those given in Table 4.1. Figure 5.3 shows the experimental results obtained for this case. As expected, the performance of the ECT controller is superior (40% to 80%) to that of the PD controller in terms of the translational and cornering overshoot measures. The PD controller, however, shows better performance in terms of the orientation

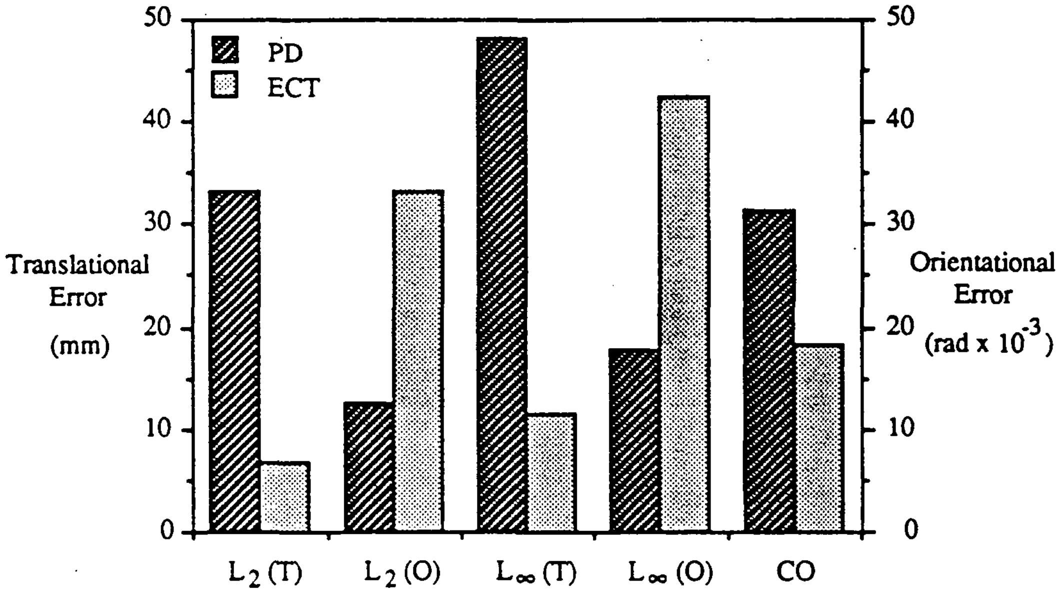

Figure 5.3: Baseline test case results

error. The joint angle trajectories for the ECT controller are shown in Fig. 5.4. Joints 6 and 7 do not perform well in this case, which contributes to the poor performance in orientation. This result is most likely due to the inaccurate model used. It is noted that friction dominates the dynamics of the last joints since their inertia values are small. No effort was made in this work to verify the frictional model used in the computation of the torque commands, and its fidelity is therefore unknown. This explanation is further supported by the results of the payload test case, described next.

5.2.2 Effects of Payload

The inertial properties of the payload are identical to those given in Section 4.3.3. All controller parameters are kept at the same values used in the baseline test case. However, for this test, the desired acceleration commands were excluded from the equivalent computed torque algorithm. This change was necessary since, in this case, the desired acceleration terms resulted in torque commands which exceeded the limits set in the testing software.

Figure 5.5 shows the experimental results obtained from the payload test. The performance of the ECT controller is superior (15% to 85%) to the performance of the PD controller, with respect to all measures. Figures 5.6 and 5.7 show the experimental performance measures of the PD and ECT controllers, respectively, both with and without the payload. As expected, the performance of the PD controller decreases with the addition of the payload . This result is primarily due

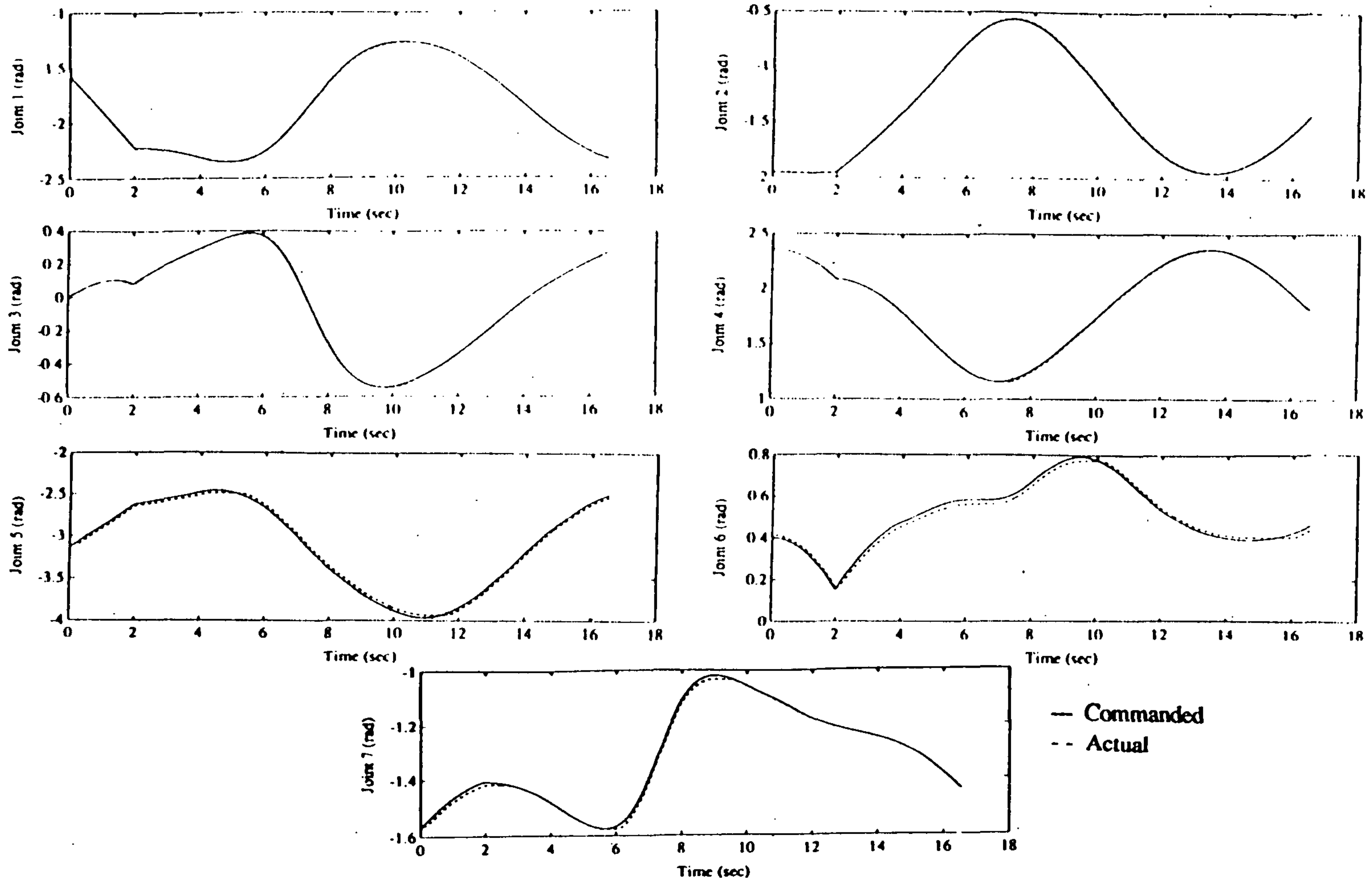

Figure 5.4: Joint trajectories of the ECT controller for the baseline test case

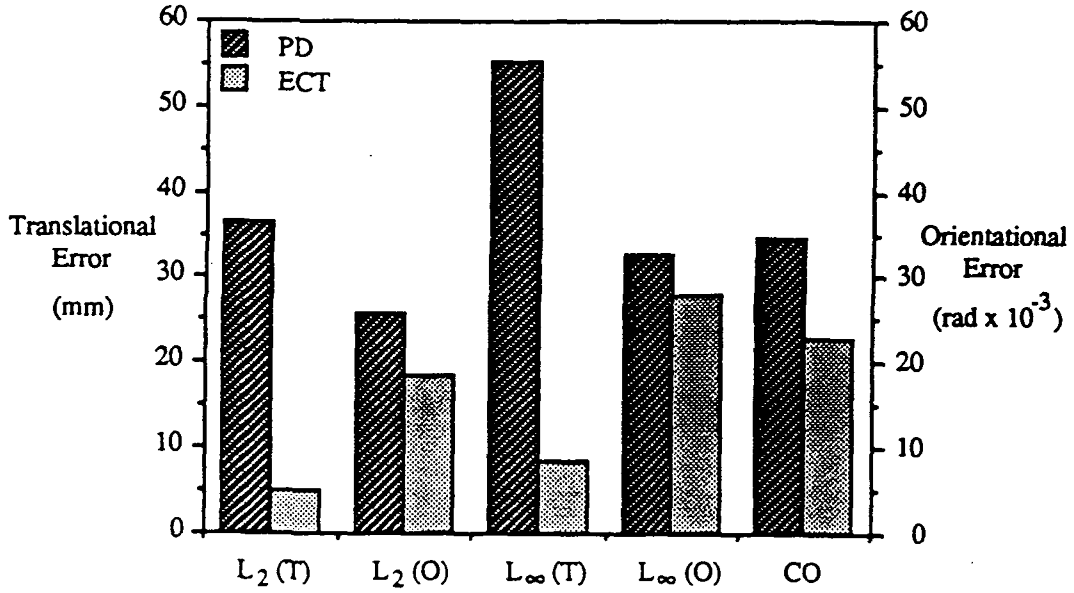

Figure 5.5: Controller comparison with payload

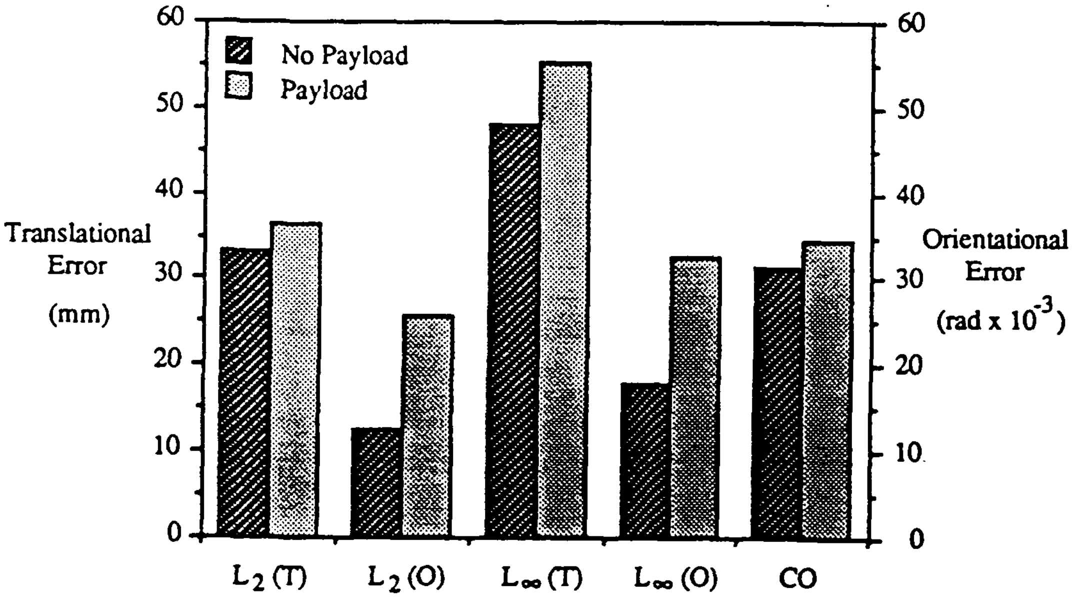

Figure 5.6: Effect of payload on the performance of the PD controller

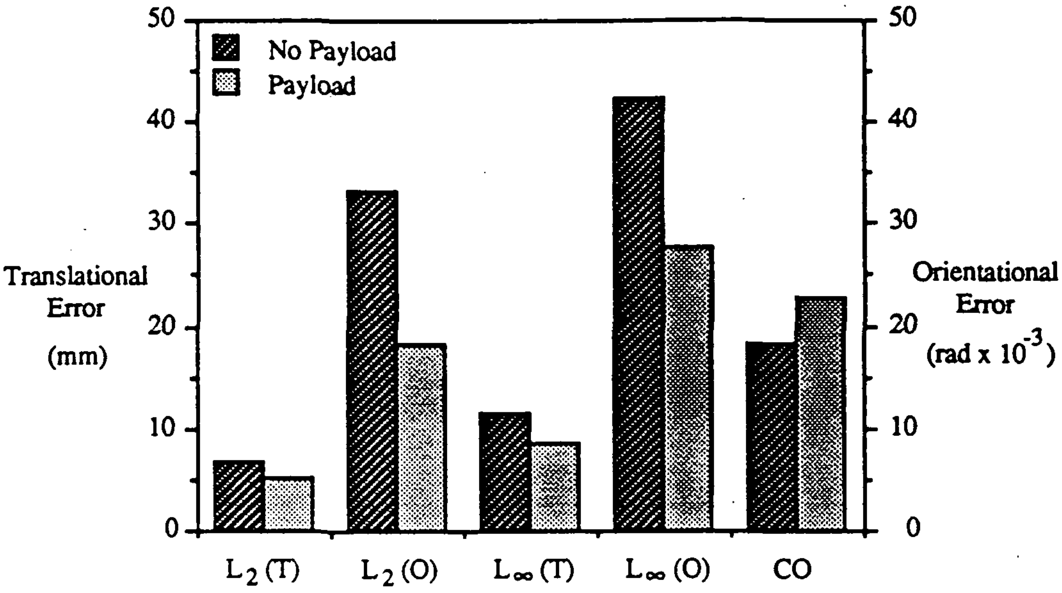

Figure 5.7: Effect of payload on the performance of the ECT controller

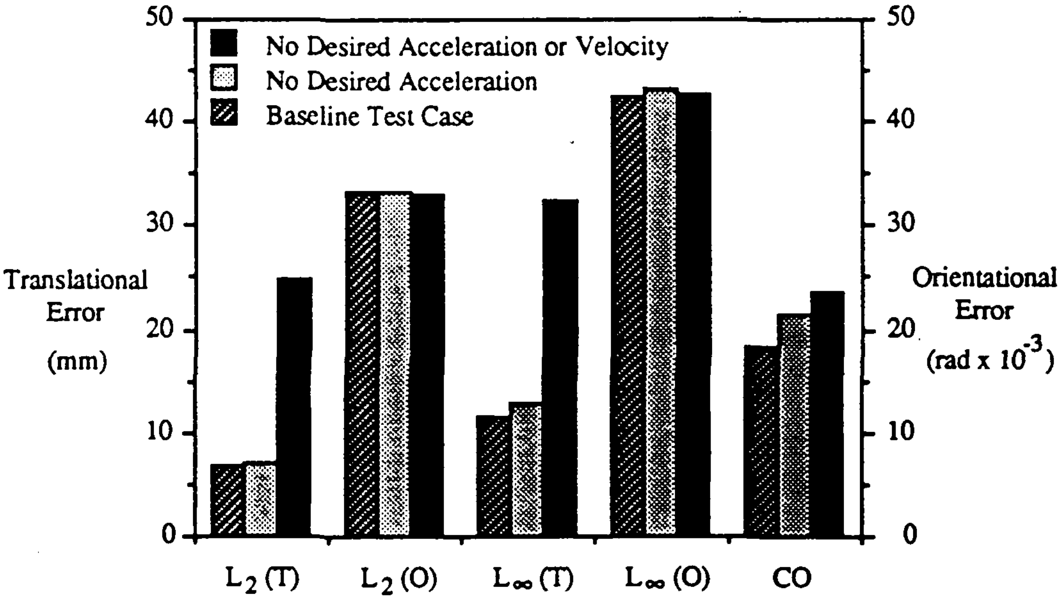

Figure 5.8: Effect of eliminating the desired acceleration and velocity terms from the ECT controller

to the increased gravity loads. However, the ECT controller is not effected in terms of the translational error, and it improves substantially in terms of the orientation error. The improved performance in orientation is due to the change in the relative importance of the inertia versus the frictional terms for the last two joints. In this case, the inertia terms for these joints have increased significantly, thus dominating the dynamical behavior.

5.2.3 Effects of Eliminating the Desired Acceleration and Velocity Terms from the Computed Torque Equations

The last experimental test case examines the effect of excluding the desired acceleration and velocity terms from the computed torque equations. Figure 5.8 shows the results of this test case. As expected, the desired acceleration term has very little effect on the translational and orientational performance. However, setting the desired velocity term equal to zero results in a dramatic performance deterioration in terms of the translational measures. The cornering overshoot also increases as the desired acceleration and velocity terms are removed. These results are identical to those obtained from the simulation tests. They indicate that the velocity terms are both significant and well approximated by the backwards difference scheme in the circular section of the path, while the acceleration terms are less significant.

5.3 Experimental Versus Simulation Results

The fidelity of the simulation package is of particular interest in controller synthesis studies. The simulation and experimental results were compared to validate the simulation tests.

Figures 5.9 and 5.10 compare the simulation and experimental results obtained from the baseline test case using the PD and ECT controllers, respectively. Figure 5.9 shows good agreement for the PD controller. However, in the case of the ECT controller, the orientational measures are significantly different. This disagreement results from the inaccuracy of the friction model, as discussed in Section 5.2.1. Since the PD controller does not rely on knowledge of the plant, the fidelity of the PD simulations is not as greatly effected by model inaccuracies.

Figures 5.11 and 5.12 compare the simulation and experimental results obtained from the payload test case using the PD and ECT controllers, respectively. Good agreement is again observed between the PD controller results. However, the results corresponding to the ECT controller show significant differences, especially in terms of orientation error. The fact that the agreement has not improved with the addition of the payload indicates that other causes, in addition to the friction model, are responsible for this divergence. Another possible source of error is the use of inaccurate estimates for the RRC torque loop parameters, particularly the torque compensator gain, K_c (see Fig. 3.8). This gain is tuned by hand for each individual RRC arm, and therefore, the actual value for the arm is different from the nominal one quoted by RRC. Variations in K_c significantly impact the ECT controller, as shown by Eqs. (3.47), (3.48), and (3.62).

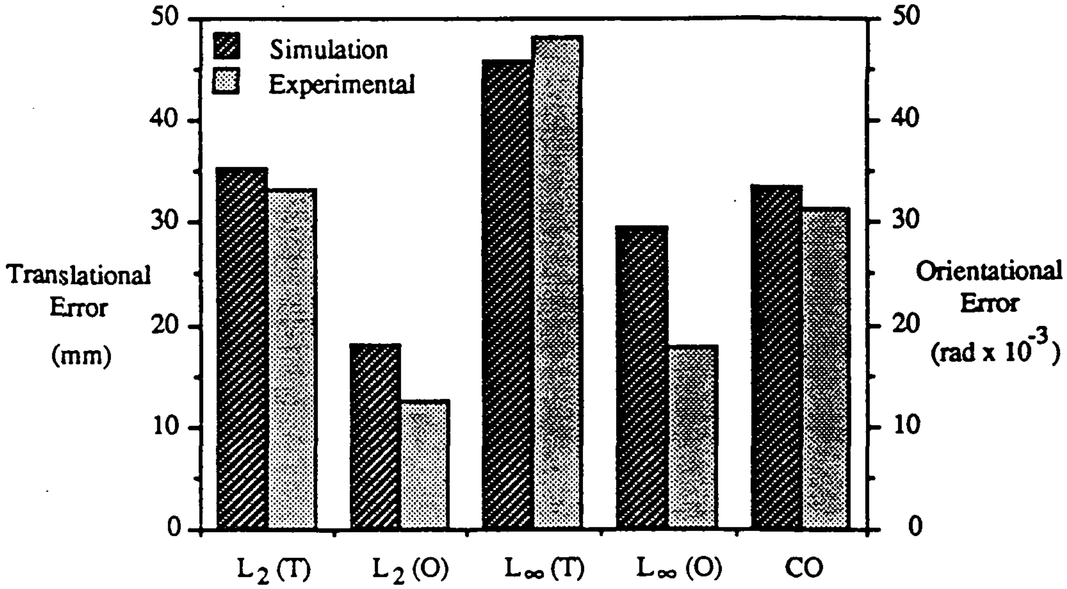

Figure 5.9: Experimental vs simulation results for the PD controller in the baseline test case

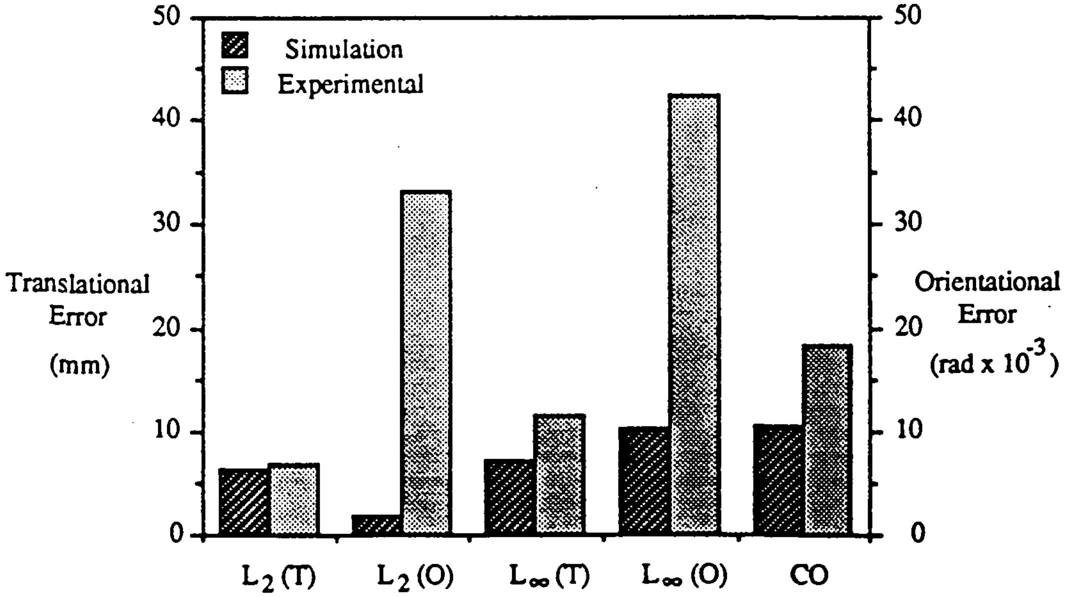

Figure 5.10: Experimental vs simulation results for the ECT controller in the baseline test case

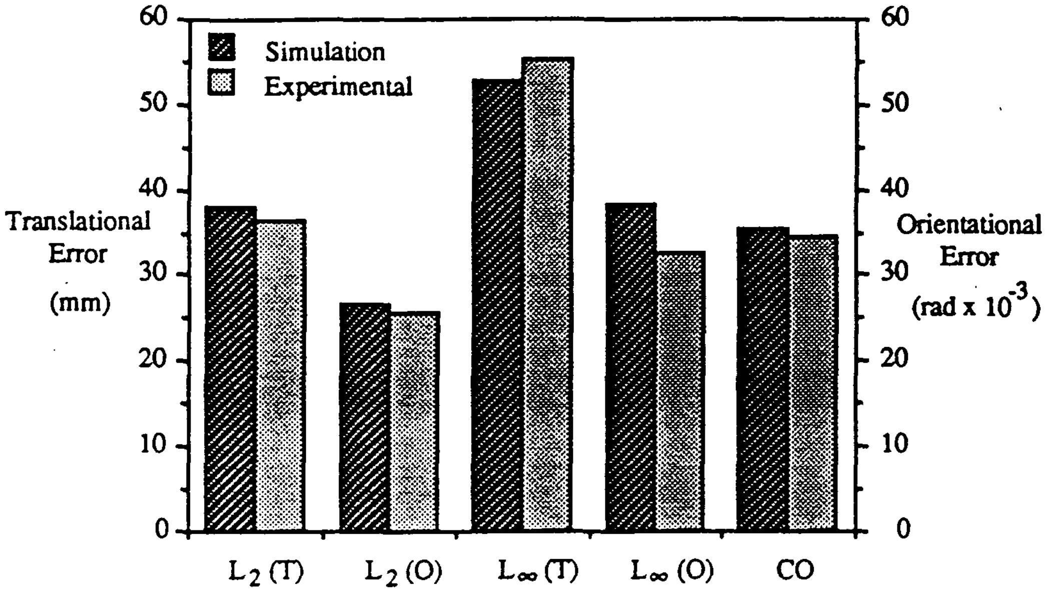

Figure 5.11: Experimental vs simulation results for the PD controller in the payload test case

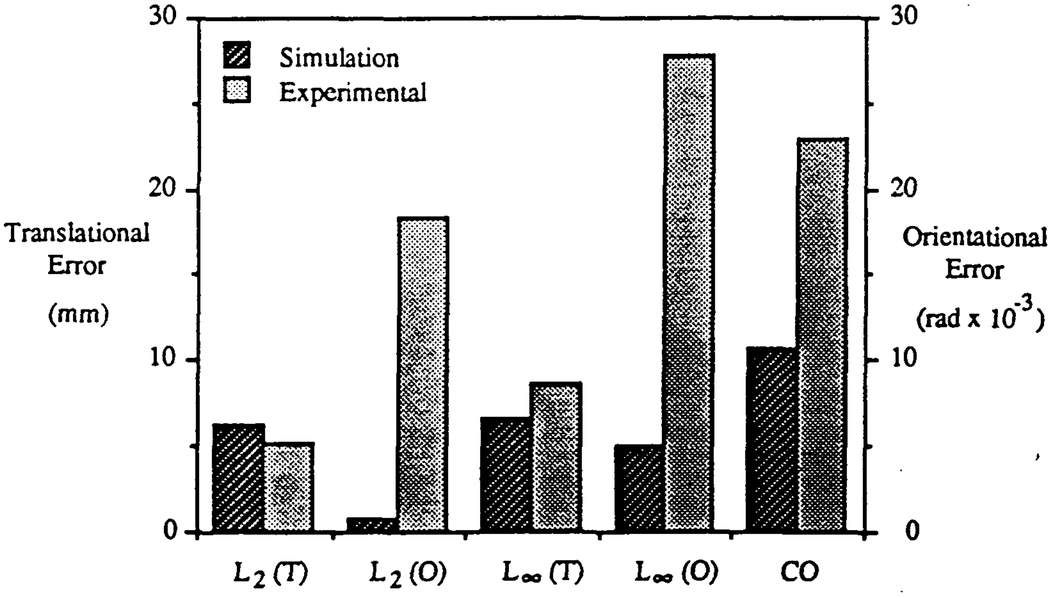

Figure 5.12: Experimental vs simulation results for the ECT controller in the payload test case

Chapter 6: Conclusions and Directions for Further Research

This study examined the implementation of computed torque controllers on a robot arm with flexible, geared, joint drive systems that are typical in many industrial robot arms. The standard computed torque algorithm is not directly applicable to this class of manipulators due to the flexibility of the joint drive system. The proposed approach combined a global computed torque algorithm with local torque controllers at each joint. The former provides the torque commands for all joints, while the latter regulates the actual torque output of the joint drive system.

Three novel control schemes were proposed. The first utilized the joint torque controllers currently implemented on the RRC robot arm and a novel form of the computed torque algorithm (ECT controller). The other two utilized the standard computed torque algorithm combined with novel joint torque controllers based on model following principles. Both continuous (CTCMF controller) and discrete (CTDMF controller) model following torque loops were developed.

The study showed that the proposed approach leads to improved tracking performance over a conventional PD controller. The ECT controller showed the best overall performance in the simulation tests. Furthermore, both model following controllers showed promising results. The ECT controller was implemented in the actual robot arm and, in experiments, also showed superior performance over the existing PD control scheme.

One of the greatest advantages of the computed torque schemes is their ability to incorporate the payload inertial properties into the control algorithm. The performance of the PD controller deteriorated when a payload was added to the end of the arm. Furthermore, the PD gains had to be tuned in order to maintain stability. The ECT and CTDMF controllers had excellent performance in the presence of the payload. Although inaccurate model parameters were shown to affect the performance of the proposed control schemes, they all exhibited robust stability in the presence of payload uncertainty. Furthermore, the ECT controller performed well in the experimental tests, despite the fact that an inaccurate friction model was used.

Increasing the joint angle and torque command update rate was shown to improve the performance of all computed torque controllers. This improvement is attributed to the increased accuracy of the backwards difference approximations for the desired acceleration and velocity commands. Furthermore, at higher update rates, the torque commands more accurately reflect the true torques required to track the desired trajectory.

There are several natural extensions to this work. These include: i) further development of the model following torque loops and implementation on the actual robot arm; ii) identification of the robot arm parameters; iii) extension of the proposed schemes to indirect self-tuning regulators.

The simulation results showed that the model following torque loops of the last joints were not robust with respect to disturbances when the arm was unloaded. A different reference model may be used to enhance the robustness of these controllers. Several suggestions are made in Astrom [1989] for the placement of the model poles and zeros so that the system is robustness with respect to disturbances. Once a more suitable model is developed, implementation on the actual robot hardware and experimental evaluation should be performed. Implementation of the discrete model following controllers appears to be the next natural step since it can be accomplished with the current hardware. However, a dedicated microprocessor board will be required to provide for the calculations of the model following controllers at the required rate (at least 1000 Hz).

The friction model for the arm was found to be inaccurate in the tests of the ECT controller. Furthermore, the parameters of the RRC torque loop may also be inaccurate. Better estimates of these parameters would result in further improvement of the ECT controller. Identification could be performed off-line through a series of well designed tests. An alternative course would be to identify these parameters on-line through an appropriate estimation scheme [Neuman, 1985; Canudas de Wit, 1991].

All three proposed controllers may be used in indirect adaptive control schemes which would include appropriate on-line estimation algorithms. This approach should result in improved performance since the parameters of the robot arm would be directly identified.

References

Armstrong, W. M. (1979), "Recursive Solution to the Equations of Motion of an N-Link Manipulator," *Proceedings 5th World Congress on the Theory of Machines and Mechanisms*, vol. 2, pp. 1343-1346.

An, C. H., C. G. Atkeson, J. M. Hollerbach (1988a), "Model Based Control of a Direct Drive Arm, Part I: Building Models," *Proceedings IEEE Conference on Robotics and Automation*, pp 1374-1379.

An, C. H., C. G. Atkeson, J. M. Hollerbach (1988b), "Model Based Control of a Direct Drive Arm, Part II: Control," *Proceedings IEEE Conference on Robotics and Automation*, pp 1386-1391.

ANSI/RIA R15.05 (1990), "Point-to-Point and Static Performance Characteristics-Evaluation."

Asada, H., T. Kanade, I. Takeyama, "Control of a Direct Drive Arm," *Journal of Dynamic Systems, Measurement, and Control*, vol. 105, 1983. pp.136-142.

Asada, H., J. J. E. Slotine (1986), *Robot Analysis and Control*, John Wiley and Sons, pp 133-170.

Astrom, K. J., B. Wittenmark (1989), *Adaptive Control*, Addison-Wesley Publishing Company, pp 105-155.

Astrom, K. J., B. Wittenmark (1990), *Computer Controlled Systems*, Addison-Wesley Publishing Company, pp 185-205.

Buhler, M., D. E. Koditschek, P. J. Kindlmann (1990), "Planning and Control of Robotic Juggling Tasks," *Fifth International Symposium on Robotics Research*, pp. 321-332.

Canudas de Wit, C., A. Aubin (1991), "Robot Parameter Identification via Sequential Hybrid Estimation Algorithm," *Proceedings IEEE Conference on Robotics and Automation*, pp 952-957.

Chen, J. (1990), "Robot Dynamics Simulation - User's Guide, Release 2," University of Maryland, College Park, MD.

Craig, J. J. (1986), *Introduction to Robotics - Mechanics and Control,* Addison Wesley Publishing Company, pp 223-252.

Craig, J. J., P. Hsu, S. Sastry (1987), "Adaptive Control of Mechanical Manipulators," *International Journal of Robotics Research*, vol. 6, no. 2, pp 16-28

Dailey, R. L. (1990), "Lecture Notes for the Workshop on H and u Methods for Robust Control," 1990 American Control Conference, pp 43-74.

DeLuca, A. (1988), "Dynamic Control of Robots with Joint Elasticity," *Proceedings IEEE International Conference on Robotics and Automation*, pp. 152-159.

Fu, K. S., R. C. Gonzalez, C. S. G. Lee (1987), *Robotics: Control, Sensing, Vision, and Intelligence*, McGraw-Hill Book Company, pp 201-265.

Ghorbel, F., J. Y. Hung, M. W. Spong (1989), "Adaptive Control of Flexible Joint Manipulators," *Proceedings IEEE International Conference on Robotics and Automation*, pp. 1188-1193.

ISO/9283 (1990), "Manipulating Industrial Robots - Performance Criteria and Related Test Methods."

Karlen, J. P., J. M. Thompson, H. I. Vold, J. D. Farrell, P. H. Eismann (1990), "A Dual-Arm Dexterous Manipulator System with Anthropomorphic Kinematics," *Proceedings IEEE Conference on Robotics and Automation*, pp 368-373.

Kanade, T., P. K. Kholsla, N. Tanaka (1984), "Real Time Control of the CMU Direct Drive Arm II Using Customized Inverse Dynamics," *Proceedings of the 23rd IEEE Conference on Decision and Control*, pp 1345-1352.

Khosla, P. K. (1988), "Some Experimental Results on Model-Based Control Schemes," *Proceedings IEEE International Conference on Robotics and Automation*, pp. 1380-1385.

Khosla, P. K., T. Kanade (1989), "Real-Time Implementation and Evaluation of Computed-Torque Scheme," *IEEE Transaction on Robotics and Automation*, vol. 5, no. 2, pp 245-253.

Leahy, M. B., K. P. Valavanis, G. N. Saridis (1989), "Evaluation of Dynamic Models for PUMA Robot Control," *IEEE Transaction on Robotics and Automation*, vol. 5, no. 2, pp 242-245.

Levin, F., M. Buhler, L. Whitcomb, D. E. Koditschek (1989), "Transputer Computer Juggles Real-Time Robotics," *Electronic Systems Design*, vol. 19, no. 2, pp 77-82.

Luh, J. Y. S., W. D. Fisher, R. P. C. Paul (1980a), "On-Line Computational Scheme for Mechanical Manipulators," *ASME Journal of Dynamic Systems, Measurements and Control*, vol. 120, pp 69-76.

Luh, J. Y. S., M. W. Walker, R. P. C. Paul (1980b), "Resolved Acceleration Control of Mechanical Manipulators," IEEE Transactions on Automatic Control, AC-25, pp 468-474.

Luh, J. Y. S., W. D. Fisher, R. P. C. Paul (1983), "Joint Torque Control by direct feedback for Industrial Robots," *IEEE Transactions on Automatic Control*, vol. AC-28, no. 2, February, pp 153-161.

Miller, D. L. (1991), "A Gravity Compensated Robot Joint Servo," A Profesional Intern Program Project, Code 714, NASA Goddard Space Flight Center, Greenbelt, MD.

Morari, M., E. Zafiriou (1989), *Robust Process Control, Prentice Hall, Inc.*, pp 221-225.

Neuman, C. P., P. K. Khosla (1987), "Identification of Robot Dynamics: An Application of Recursive Estimation," *Proceedings 4th Yale Workshop on Application of Adaptive Systems Theory*, New Haven, pp 42-49.

Paul, R. P. (1972), "Modelling, Trajectory Calculation, and Servoing of a Computer Controlled Arm," AIM-177, Stanford University Artificial Intelligence Laboratory.

Paul, R. P. (1981), Robot Manipulators: Mathematics, Programming, and Control, MIT Press, pp 157-194.

Pfeffer, L. E., O. Khatib, J. Hake (1989), "Joint Torque Sensory Feedback in the Control of Puma Manipulator," *IEEE Transactions on Robotics and Automation*, vol. 5, no. 4, August, pp 418 - 425.

Samsan Version 3.1 User's Guide (1990): Modern Numerical Methods for Classical Sampled System Analysis, NASA/Goddard Space Flight Center, Greenbelt, MD 20771, June.

Schmitz, D., P. K. Khosla, Y. Kanade (1985), "Development of CMU Direct-Drive Arm II," *Proceedings 15th International Symposium on Industrial Robotics.*

Shieh, Win-Bin (1990), "Dynamic Motion Simulation of a Class of Multi-Link Robot Manipulators," M.S. Thesis, University of Maryland, College Park, MD.

Slotine, J. J. E., W. Li (1987), "On the Adaptive Control of Robot Manipulators," *International Journal of Robotics Research*, vol. 6, no. 3, pp 49-59.
Spong, M. W. (1987), "Modelling and Control of Elastic Joint Manipulators," *ASME Journal of Dynamic Systems, Measurement, and Control*; vol. 109, pp 310-319.

Sweet, L. M., M. C. Good (1985), "Redefinition of the Robot Motion Control Problem," *IEEE Control Systems Magazine*, August, pp 18-25.

Tarn, T. J., A. K. Bejczy, S. Ganguly, Z. Li (1990), "Nonlinear Feedback Method of Robot Control: A Preliminary Study," *Proceedings IEEE International Conference on Robotics and Automation*, pp. 2052-2057.

Tourassis, V. D., C. P. Neuman (1985), "Robust Nonlinear Feedback Control for Robotic Manipulators," IEE Proceedings, vol. 132, part d, no. 4, July, pp 134-143.

Uicker, J. J. (1965), "On the Dynamic Analysis of Spatial Linkages Using 4x4 Matrices," Ph.D. dissertation, Northwestern University, Evanston, IL.

Whitcomb, L. L., A. A. Rizzi, D. E. Koditschek (1991), "Comparative Experiments with a New Adaptive Controller for Robot Arms," *Proceedings IEEE International Conference on Robotics and Automation*, pp. 2-7.