Accurate Trajectory Control of Robotic Manipulators

J. C. Van Winssen and C. W. deSilva

CMU-RI-TR-85-15

Department of Mechanical Engineering
The Robotics Institute
Carnegie-Mellon University
Pittsburgh, Pennsylvania 15213

April 1985

This project was funded in part through a grant from the Atlantic Richfield Foundation.

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

This report presents a control scheme for accurate trajectory following with robotic manipulators. The method uses feedforward control using model-based torques for fast operation and gross compensation, and adaptive feedback control for correcting deviations from the desired trajectory under feedforward control. The adaptive controller eliminates trajectory-following errors in the least squares sense. The control scheme takes into account dynamic nonlinearities (e.g., coriolis and centrifugal accelerations and payload changes), geometric nonlinearities (e.g., nonlinear coordinate-transformation matrices) and physical nonlinearities (e.g., nonlinear damping) as well as dynamic coupling in manipulators. Computer simulations are presented to indicate the effectiveness and robustness of the control scheme. When the desired trajectory is completely known before the control scheme is implemented, then off-line computations can be used to generate the adaptive feedback gains and the computational efficiency will not be a major limiting factor with this control scheme. If real-time changes in the desired trajectory have to be accommodated, the computational efficiency has to be improved using recursive relations to compute the adaptive gains. The necessary recursive relations are derived and presented in this report.

1. INTRODUCTION

Many robot applications today and in the future will require accurate tracking of a prespecified continuous path. Common examples of these tracking applications include seam tracking, arc welding, cutting (laser and water jet), spray painting, contours inspection, co-ordinated parts transfer and assembly operations. These tracking paths are usually specified with respect to the end effector of the robotic manipulator and can specify trajectories with respect to time as well as position. The problem with achieving this objective of temporal path following is that strong nonlinearities in the dynamics and geometry, unknown parameters, modeling errors, measurement errors, unplanned changes in operating conditions, and other disturbances are present in the manipulator and they make accurate control of the manipulator very difficult.

To achieve this goal of accurate path following, a control system is needed, which

1. accurately tracks the desired end effector trajectory, often in terms of time as well as position;

2. rejects a wide class of disturbances, such as parameter variations (i.e., changing payload), vibrations and the effects of static friction, and measurement errors;

3. has minimal complexity, is computationally fast, can accommodate a high sampling rate;

4. is very reliable, particularly in terms of robustness of the control scheme.

Many control systems, which meet these requirements with different degrees of success, have been proposed and some have been implemented. The control scheme developed in this report can accurately follow a prespecified trajectory while rejecting many classes of disturbances by using a feedback control scheme that minimizes position and velocity deviation in the least squares sense while allowing for the changing of the feedback control parameters to account for unknown changes in payload or desired trajectory. A two-link manipulator simulation shows the effectiveness of this control scheme for trajectory following. However, the computational effort required with this control scheme is high enough to limit the maximum sampling frequency allowed for manipulator control in real time. Therefore the maximum trajectory-following accuracy that this control scheme can achieve is also limited by the computational effort, if the desired trajectory is not known a priori, and is changing in real time.

1.1 Control Schemes

Linear servo control is the most common type of control in commercial use today [3]. This control method involves having a separate feedback loop closed over each manipulator joint that feedbacks the position (and sometimes velocity) of that joint. This control method has several problems which limit its commercial usefulness. Since each control loop is closed independently over each manipulator joint, it has poor compensation for the dynamic coupling (i.e., particularly coriolis forces and coordinate coupling) between joints because the effect of the motion of one joint on another is viewed as a disturbance which the feedback controller of the second joint must compensate for. At low speeds, these "disturbance" forces are small and can be easily compensated for, but at high speeds, these forces are major components in the dynamics of the manipulator, and the controller will fail to totally reject these "disturbances" and the end effector will no longer be following the correct path [8]. Another factor is that the servo parameters usually are tuned for one set of operating conditions and can not be changed to meet changing conditions like payload variations during robot operation. Furthermore, classical servo control assumes linear plants, which is not close to reality in the case of robotic manipulators.

Other control schemes have been proposed that eliminate some of these problems but none have been commercially implemented. These methods include Model-Referenced Adaptive Control, Sliding Mode Control (a method of designing switching feedback regulators based on minimum time, bang-bang control), optimal control, nonlinear feedback control and feedforward control. Application of these control techniques, particularly for real-time control, is hindered by the complexity of the associated control algorithms, which increases the computation-cycle time and decreases the control bandwidth.

In model-reference adaptive control [4, 5], feedback controller parameters are adaptively changed to drive the manipulator response toward that of a reference model. This reference model need not represent the actual manipulator and is chosen to suit the required dynamic behavior. For example, a simple oscillator (a linear second-order differential equation) could be used as the reference model for each joint of the manipulator.

Controller parameters are adjusted according to a differential law that uses the error signal (the difference between response of the reference model and the actual robot) as the input. There exist several drawbacks in this scheme, including the following:

1. Structure of the feedback controller is not automatically generated by the control scheme.

2. The adaptive law has to be derived from scratch for the particular reference model chosen.

3. The control law is completely independent of the robot model.

4. The control action has to be generated faster than the speed at which the nonlinear terms in the robot change.

5. The adaptive law is derived on the assumption that some of the nonlinear terms in the robot model remain constant.

It is clear that even though this technique can produce satisfactory results, particularly due to the presence of adaptive feedback loops, there is no guarantee that the required accuracy is obtained in a given situation of trajectory following.

A control technique that strives to obtain linear behavior from a nonlinear manipulator is known as sliding mode control [9]. In the generalized case of this method (only the two dimensional case is presented by Klein and & Maney [9]), the state space is partitioned into several regions that are bounded by a space trajectory conformal to the desired linear behavior. The objective of the control would be to drive the manipulator along the desired trajectory. This is accomplished by assigning a different control law for each region in the partitioned state space. If the manipulator deviates from the desired trajectory and enters a particular region of the state space the corresponding control law is switched on. This will drive the manipulator back into the desired trajectory. If it overshoots, however, the control law of the new region which the manipulator entered will be automatically switched on to drive the the manipulator into the desired trajectory. If the alternative control laws that are assigned to the various regions can be switched on at infinite frequency, which is of course not realistic, it is possible in theory, to obtain ideal behavior. In practice, however, the response will chatter about the desired trajectory. The amplitude of chatter will depend on the manipulator dynamics as well as control gains used. In addition the switching frequency will depend on the deadband of control. These shortcomings of sliding mode control can be aggravated by the fact that the control laws are selected in a heuristic manner, without even employing a model to represent the actual dynamics of the manipulator. At its best, sliding mode control usually brings about time delays (non-synchronous response) in addition to chatter. This technique too, has not been implemented in commercial robots.

In optimal control, the feedback control law is designed by optimizing a suitable performance index using a dynamic model for the manipulator. Control laws obtained in this manner can be highly complex except in a very few special cases. A nonlinear control approach that has been proposed for robotic manipulator control is aimed at obtaining a desirable linear behavior from the manipulator by employing a highly nonlinear feedback law [6, 1]. Unlike the model-referenced adaptive control method, this control law is derived from an accurate nonlinear model for the robot. The main disadvantage of the method, as has been warned by Asada & Hanafusa, [1] is the feedback law that is so complex, it is virtually impossible to compute the feedback parameters in real time for practical robots. Furthermore, performance of this nonlinear control system is known to be quite sensitive to fidelity of the robot model that is employed.

2. CONTROL

This control scheme developed in this report involves the combination of feedforward control with a least squares adaptive feedback control scheme.

2.1 Feedforward Control

This is an open loop control method. This method involves calculating the torques that must be applied at each manipulator joint so as to have the end effector follow the desired trajectory. These torques are computed by from the differential equation which models the dynamics of the n-degree of freedom robotic manipulator. This is known as the inverse-dynamics problem;

$$\mathbf{M}(\mathbf{q},W)\ddot{\mathbf{q}} + \mathbf{f}(\mathbf{q},\dot{\mathbf{q}},W) = \tau(t) \tag{1}$$

where

W : payload

$\mathbf{q}$: vector of generalized joint positions

$\mathbf{M}(\mathbf{q},\mathrm{W})$: inertia matrix (n x n)

$\mathbf{f}(\mathbf{q},\dot{\mathbf{q}},\mathrm{W})$: vector representing centrifugal, coriolis, dissipation and gravitational forces

$\tau(\mathrm{t})$: input torques or forces at the manipulator joints

In practical manipulators, input signals (e.g., field voltages, servovalve commands) produce motor torques at the joints, with some dynamic delay. Motor torques are converted into the torques that are actually applied to the links of the manipulator, with additional dynamic delay. Manipulator displacements are a result of these joint torques. It is therefore clear that, by either measuring or computing joint torques it is possible to eliminate part of the delay in a manipulator control system. Consequently, feedforward control has the advantage of speeding up the manipulator response. Furthermore, torque disturbances can be calculated or measured, they can be completely rejected using feedforward control. A main disadvantage of feedforward control, in the present context, is that due to model errors and unknown disturbances, the calculated torque is not the ideal torque and as a result errors can grow in an unstable manner unless some form of feedback control is used.

Since in inverse dynamics a mathematical model of the manipulator is used to calculate the joint torques required, when these torques are applied to the actual manipulator it might not follow the desired trajectory accurately. This would be due to the cumulative effects of modeling inaccuracies, computational limitations, and unaccounted for effects like vibrations and static friction. Therefore, for accurate tracking using feedforward control a precise dynamic model has to be employed and the manipulator must be made very rigid with strong structural links and precision gear trains and actuators. Another problem with this method is that the computational effort required to accurately compute the necessary torques in a real-time situation can become very significant if the desired trajectory is not known a priori and may not allow a sufficiently high sampling rate for good control bandwidth.

An adaptive feedback is used in the present control method to correct for these problems.

2.2 Background Theory

In most instances, feedforward control needs a feedback controller to correct for unaccounted disturbances in the system. Since linear-servo control offers only a limited ability to compensate for nonlinearities, model errors, measurement errors and disturbances a more adaptive feedback controller was developed by R.P. Paul [2]. This controller is based on a nonlinear coupled dynamic model of the manipulator, and therefore takes into account effects that linear control usually neglects. It also allows for updating the control parameters to take care of unknown external disturbances and payload variations. The basic block diagram for the control system is seen in figure 1.

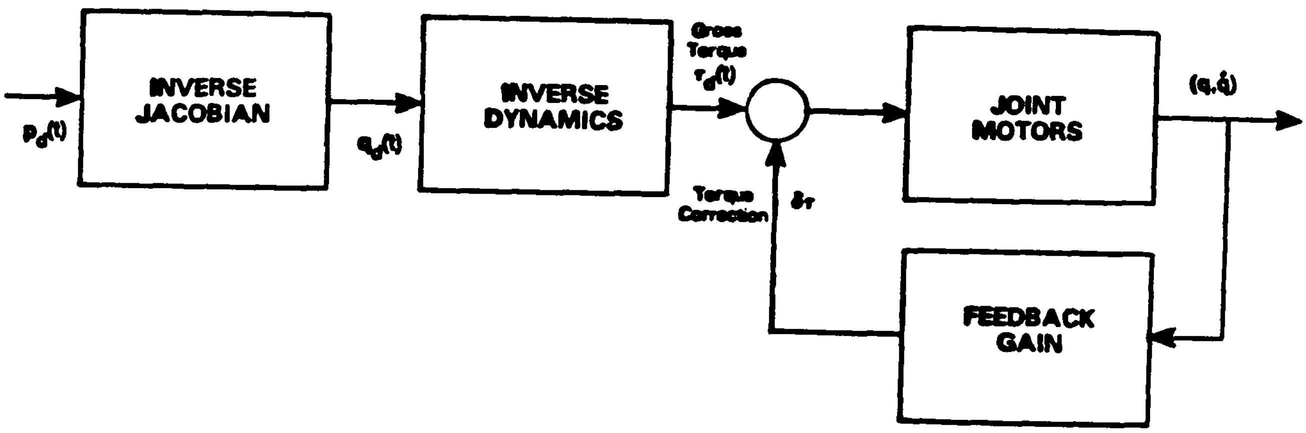


Figure 1. Basic control diagram for the manipulator

2.2.1 Linearization

We can linearize the nonlinear set of differential equations (1) with respect to small perturbations, $\delta \mathbf{q}$, from the desired trajectory, $\mathbf{q}_d(t)$, caused by small torque disturbances, $\delta \tau(t)$

$$\mathbf{M}(\mathbf{q}_d, W)\,\delta \ddot{\mathbf{q}} + \ddot{\mathbf{q}}_d \frac{\partial \mathbf{M}}{\partial \mathbf{q}}(\mathbf{q}_d, W)\,\delta \mathbf{q} + \frac{\partial \mathbf{f}}{\partial \mathbf{q}}(\mathbf{q}_d, \dot{\mathbf{q}}_d, W)\,\delta \mathbf{q} + \frac{\partial \mathbf{f}}{\partial \mathbf{q}}(\mathbf{q}_d, \dot{\mathbf{q}}_d, W)\,\delta \mathbf{q} = \delta \tau(t) \tag{2}$$

where

$$\left[\ddot{\mathbf{q}}_d \frac{\partial \mathbf{M}}{\partial \mathbf{q}}\right]_{ij} = \sum_{k=1}^{n} \frac{\partial \mathbf{M}_{ik}}{\partial \mathbf{q}_j} \ddot{\mathbf{q}}_k$$

This equation can be rearranged in vector-matrix form

$$\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{M} \end{bmatrix}_d \begin{bmatrix} \delta \dot{\mathbf{q}} \\ \delta \ddot{\mathbf{q}} \end{bmatrix} + \begin{bmatrix} \mathbf{0} & -\mathbf{I} \\ \ddot{\mathbf{q}} \frac{\partial \mathbf{M}}{\partial \mathbf{q}} + \frac{\partial \mathbf{f}}{\partial \mathbf{q}} & \frac{\partial \mathbf{f}}{\partial \dot{\mathbf{q}}} \end{bmatrix}_d \begin{bmatrix} \delta \mathbf{q} \\ \delta \dot{\mathbf{q}} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{I} \end{bmatrix} \delta \tau(t) \tag{3}$$

where, $[\;]_d$, denotes terms evaluated in terms of the desired trajectory, $q_d(t)$.

This is, in fact, a state space representation with the state vector and the input vector given by

$$\mathbf{x} = [\ \delta \mathbf{q},\ \delta \dot{\mathbf{q}}\ ]^T, \qquad \mathbf{u} = \delta \tau$$

thus,

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \tag{4}$$

where, the system matrix

$$\mathbf{A}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \mathrm{W})_d = - \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{M}^{-1} \end{bmatrix}_d \begin{bmatrix} \mathbf{0} & -\mathbf{I} \\ \ddot{\mathbf{q}} \frac{\partial \mathbf{M}}{\partial \mathbf{q}} + \frac{\partial \mathbf{f}}{\partial \mathbf{q}} & \frac{\partial \mathbf{f}}{\partial \dot{\mathbf{q}}} \end{bmatrix}_d \tag{5}$$

and the input gain matrix is

$$\underline{\mathbf{B}}(\mathbf{q}_d, \mathrm{W}) = \begin{bmatrix} \mathbf{0} \\ \mathbf{M}^{-1} \end{bmatrix}_d \tag{6}$$

Since what is developed would be implemented as a digital control scheme, we need the discrete form of the state space representation for $$\frac{dx}{dt} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}(t)$$

The solution to this linear differential system starting at $t=t_o$, can be represented as

$$x(t) = \Phi(t,t_o)x(t_o) + \int_{t_o}^{t} \Phi(t,\beta)B(\beta)u(\beta)d\beta \tag{7}$$

which assuming time invariance in the neighborhood of the perturbations, can be expressed as the set of difference equations

$$\mathbf{x}(k+1) = \Phi\mathbf{x}(k) + \Gamma\mathbf{u}(k) \qquad k = 0,1,2,3,...$$

in which

$\Phi = e^{AT}$ = state transition matrix

$\Gamma = \int_0^T e^{A\beta}d\beta\mathbf{B}$ = input gain matrix

T = data sampling period

2.2.2 Minimization

Since the state vector $\mathbf{x}$ represents the deviation in position and velocity, from the desired trajectory, then the objective of the minimization is to drive $\mathbf{x}$ to zero as fast as possible. This will be accomplished in the least squares sense by using the following objective index

Least Squares Minimization Performance Index :

$$J = \sum_{k=1}^{N} [\Phi\mathbf{x}(k) + \Gamma\mathbf{u}(k)]^T\, \mathbf{Q}\, [\Phi\mathbf{x}(k) + \Gamma\mathbf{u}(k)] \tag{8}$$

where $\mathbf{Q}$ is a diagonal weighting matrix. $\mathbf{Q}$ is used to weight the relative importance of each joint position or velocity. This allows the motions of critical joints to be more heavily weighted than the motions of other joints.

This minimization is a Linear Quadratic Regulator (LQR) minimization problem so the optimal feedback gain should be in some form of the steady-state Ricatti equation.

2.2.3 Optimal Feedback Gain

Using straightforward calculus it can be shown that the optimal control law is given by

$$\mathbf{u}(k) = -\mathbf{K}\mathbf{x}(k) \tag{9}$$

where $$\mathbf{K} = (\Gamma^T \mathbf{Q} \Gamma)^{-1} \Gamma^T \mathbf{Q} \Phi \mathbf{x}(k) \tag{10}$$

It should be noted that this feedback control law is realizable if

$$\text{rank}(\Gamma^T \mathbf{Q} \Gamma) = n \tag{11}$$

In particular, if

$\mathbf{Q}$ is positive definite, we must have

$$\text{rank}(\Gamma) = n \tag{12}$$

where, n = degrees of freedom of manipulator

2.3 Control Strategy

The complete control strategy for the manipulator is shown in figure 2. First the desired end-effector trajectory of the manipulator is generated. Then, using some inverse kinematics scheme, each incremental displacement, velocity and acceleration of the end-effector is translated into the corresponding motions of the n joints. With the inverse dynamics of the manipulator, the desired gross torques for each joint can be calculated. These torques are applied to the actual manipulator in a feedforward manner. The actual joint positions and velocities are then measured once every period, T_s, using resolvers or encoders. The difference between the actual and the desired joint motions is then multiplied by the optimal feedback gain matrix, K, to produce the vector of torque corrections that need to be added to the gross torque vector for proper control. A suitable criterion is needed to decide when to update the feedback gain matrix, K. In the present work the following criterion is used:

- Initially specify the weighting matrix $\mathbf{Q}$ and calculate, Φ, and ,Γ.
- Compute the initial feedback gain matrix, K using equation (10).
- Update the feedback gain matrix, K, according to the criterion

1.	If $\lVert \mathbf{x} \rVert < \epsilon_0$	Skip torque error feedback
2.	If $\lVert \mathbf{x} \rVert > \epsilon_1$	Update $\Phi,\Gamma,\mathbf{Q}$, and K
3.	If $\lVert \mathbf{x} \rVert > \epsilon_2$	Excessive Error, terminate operation

Note that $\epsilon_0 < \epsilon_1 < \epsilon_2$. The error norm is defined as $\lVert \mathbf{x} \rVert = \sum_{i=1}^{n} a_i \, |x_i|$

- Update the weighting matrix, $\mathbf{Q}$, by changing the diagonal elements in proportion to the maximum absolute value of the state, $|x_i|_{max}$

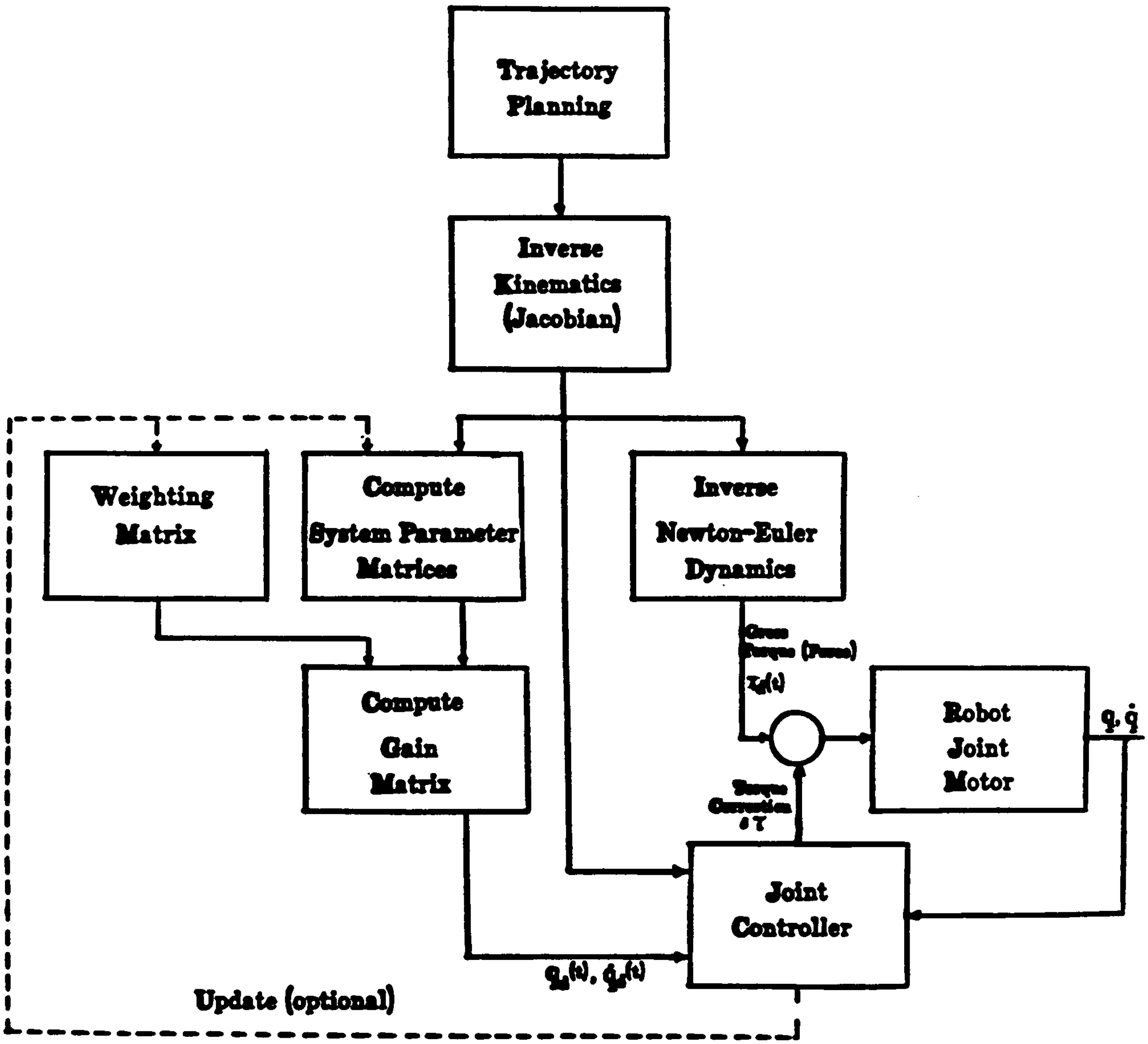


Figure 2. Complete block diagram for control strategy

2.3.1 Stability

If the manipulator model is significantly different from the actual robot, then the feedback law could cause instability in our control system. Stability is guaranteed if the closed-loop state transition matrix, Φ^c, has all its eigenvalues inside the unit circle on the Z-plane. Note that

$$\Phi^c = \Phi - \Gamma\ [(\Gamma_o^T\ Q\ \Gamma_o)^{-1}\ \Gamma_o^T\ Q]\ \Phi_o$$

where

Φ , Γ = actual plant manipulator matrices

Φ_o , Γ_o = manipulator model matrices

3. SIMULATION RESULTS

The effectiveness of the control strategy presented in this report, is examined using a two-degree-of-freedom manipulator. The manipulator equations are given in Appendix A. Two types of disturbances were tested for this control scheme:

1. a 7% external disturbance (figures 3.1 and 5.1), and

2. a 7% error in link lengths and a 9% error in link inertias (figure 4.1).

Typical results corresponding to these three cases are presented in figures 3, 4, and 5. In all three cases the feedforward control alone produces an unstable trajectory following. By adding the adaptive optimal feedback controller the actual trajectory was brought very close (8% maximum position error) to the desired trajectory.

It appears that our control scheme satisfies three of the four design goals for the controller: accurately tracks the end effector, rejects a wide class of disturbances, and is very reliable. The last goal is minimal complexity, or making the scheme computationally fast enough to allow an adequate sampling rate for on-line trajectory generation and control.

3.1 Two-Link Manipulator Results

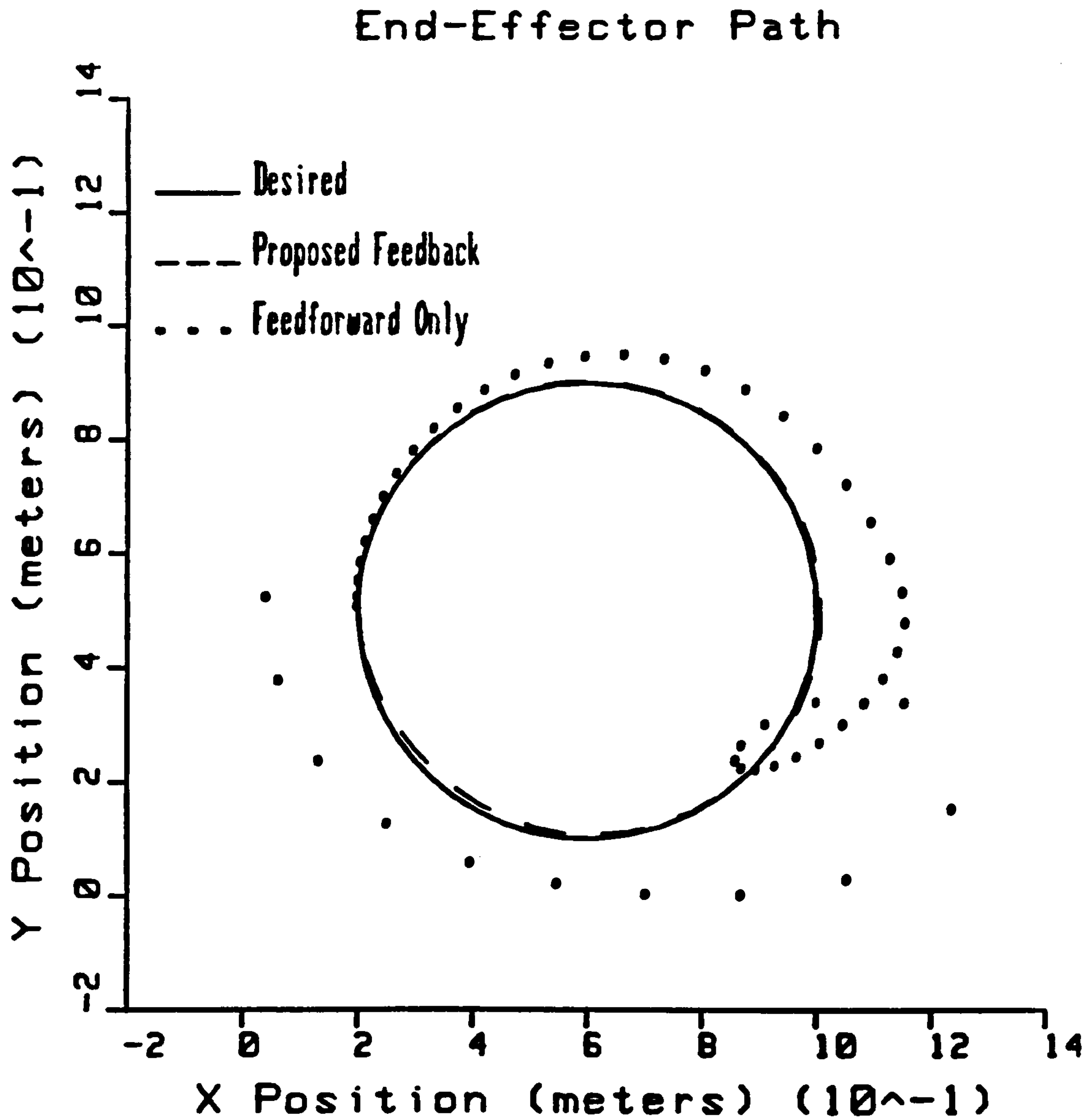


Figure 3.1 End-effector path with input disturbances

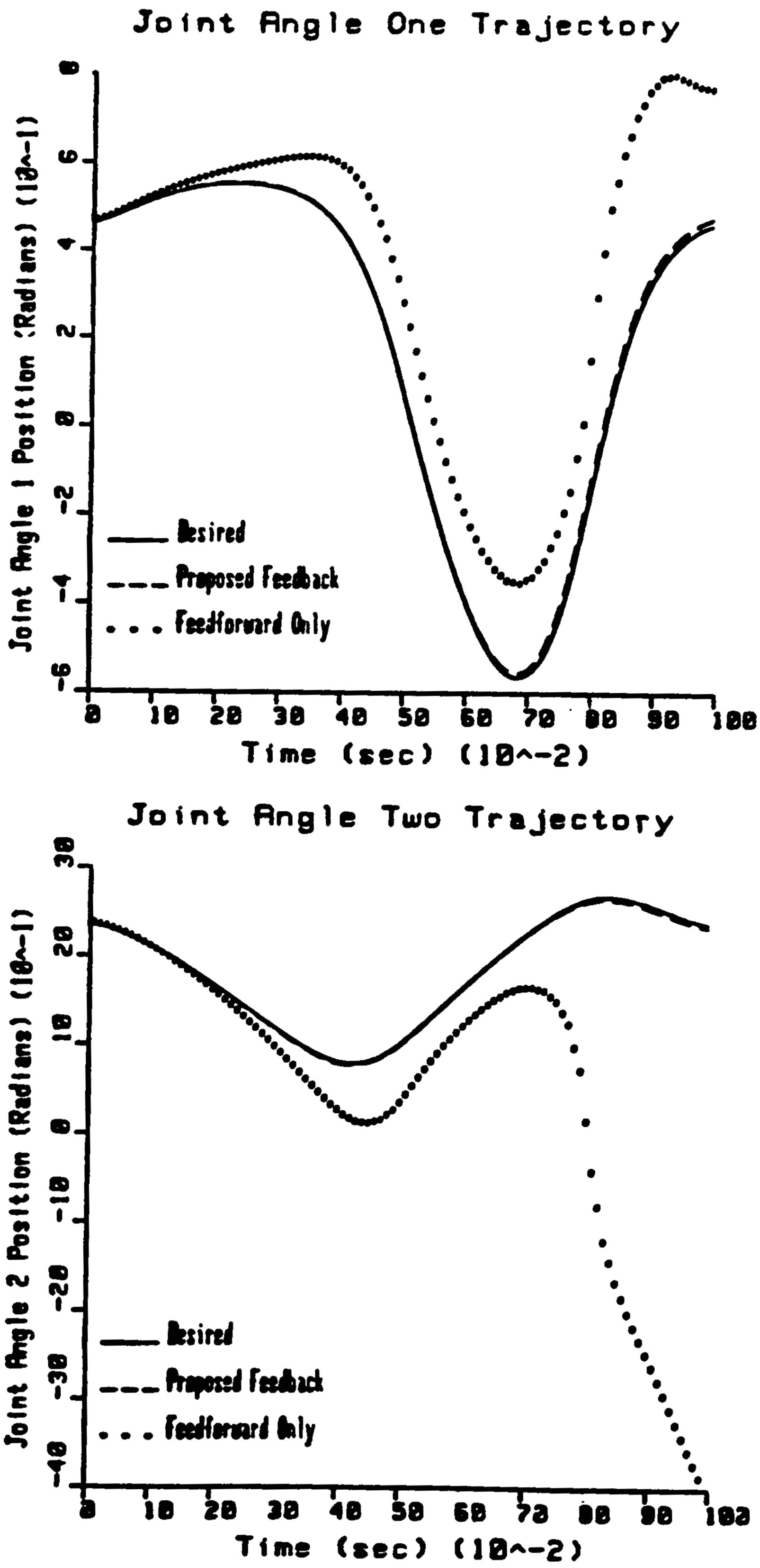
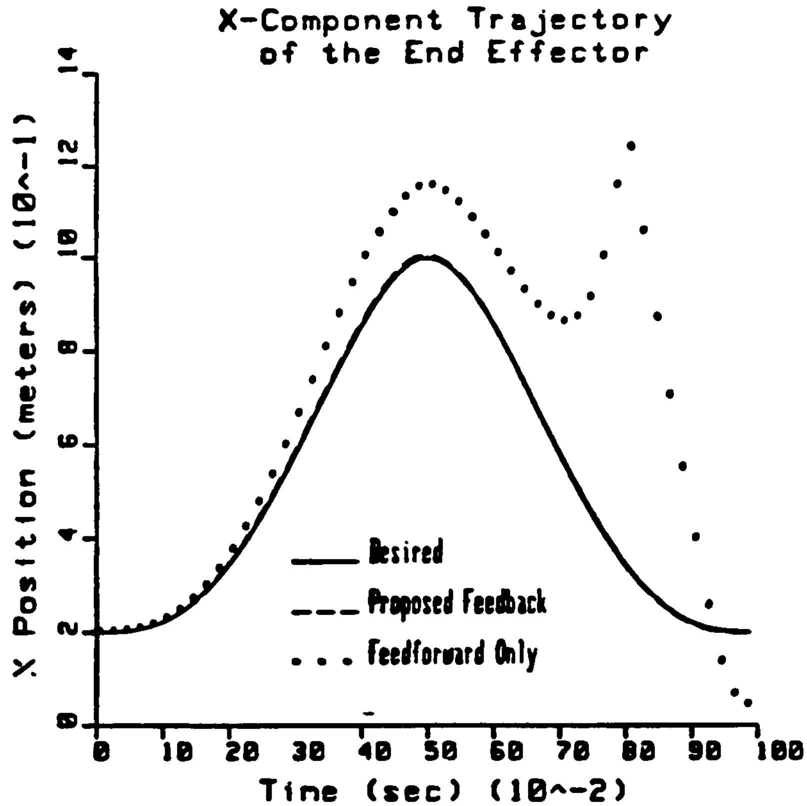


Figure 3.2 Joint trajectories with input disturbances



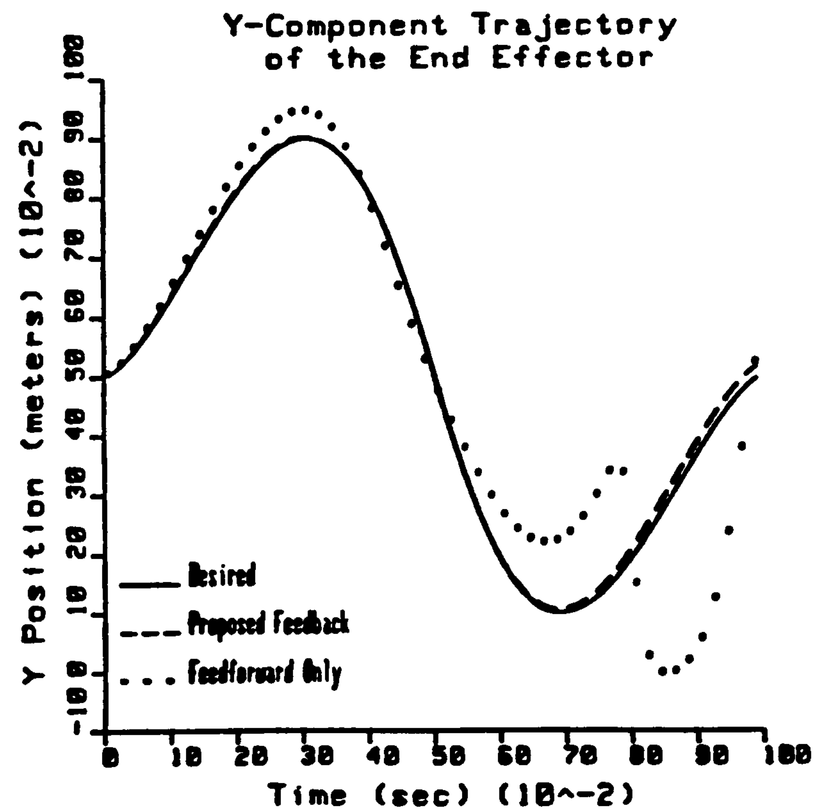


Figure 3.3 X and Y position trajectories of the end-effector with input disturbances

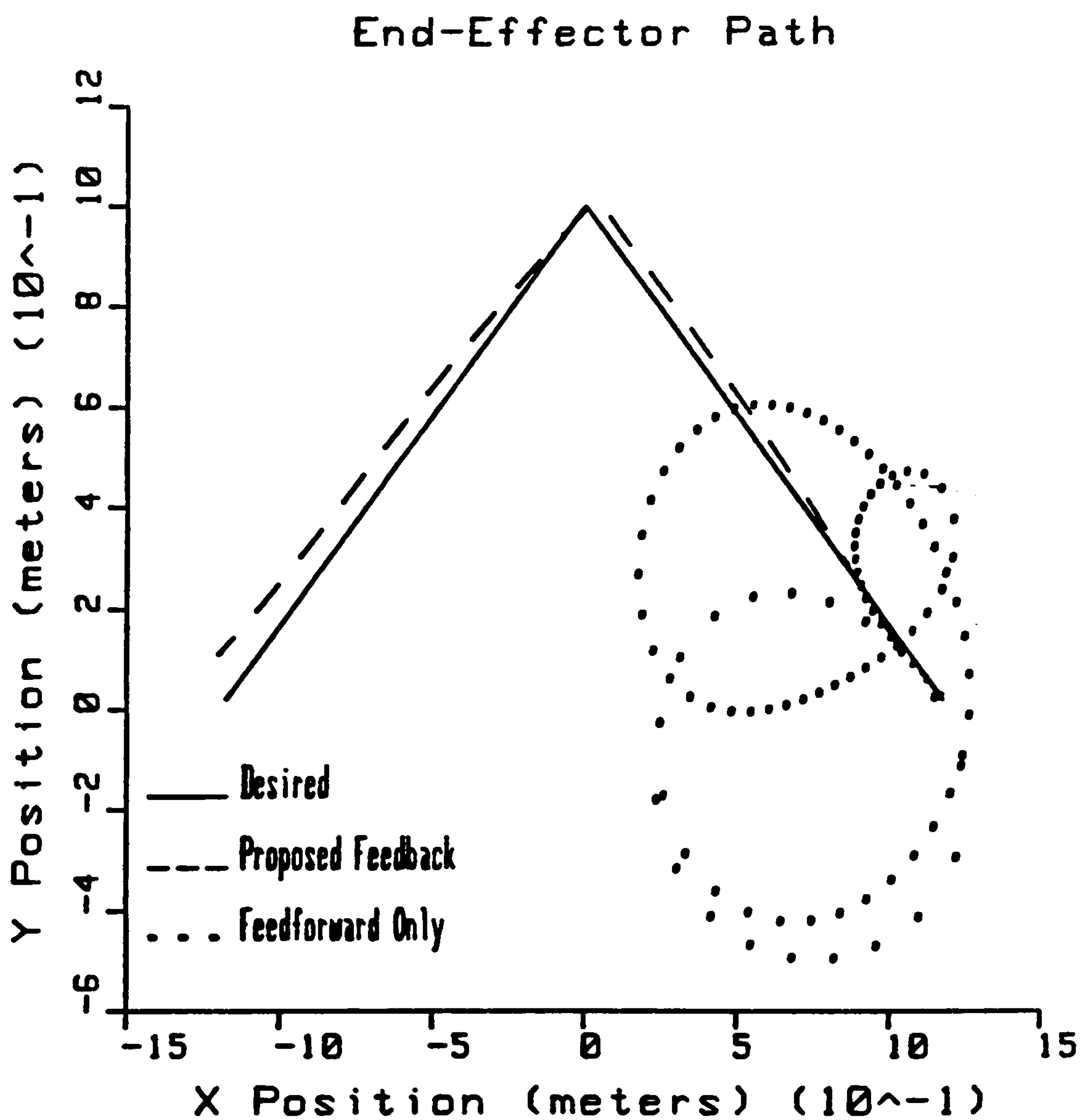
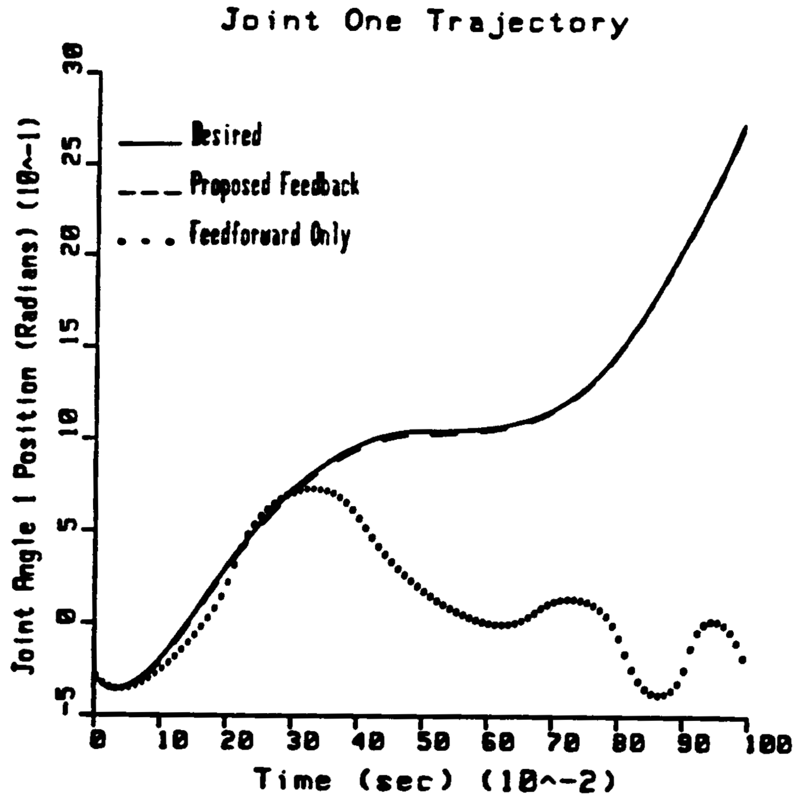


Figure 4.1 End-effector path with model errors



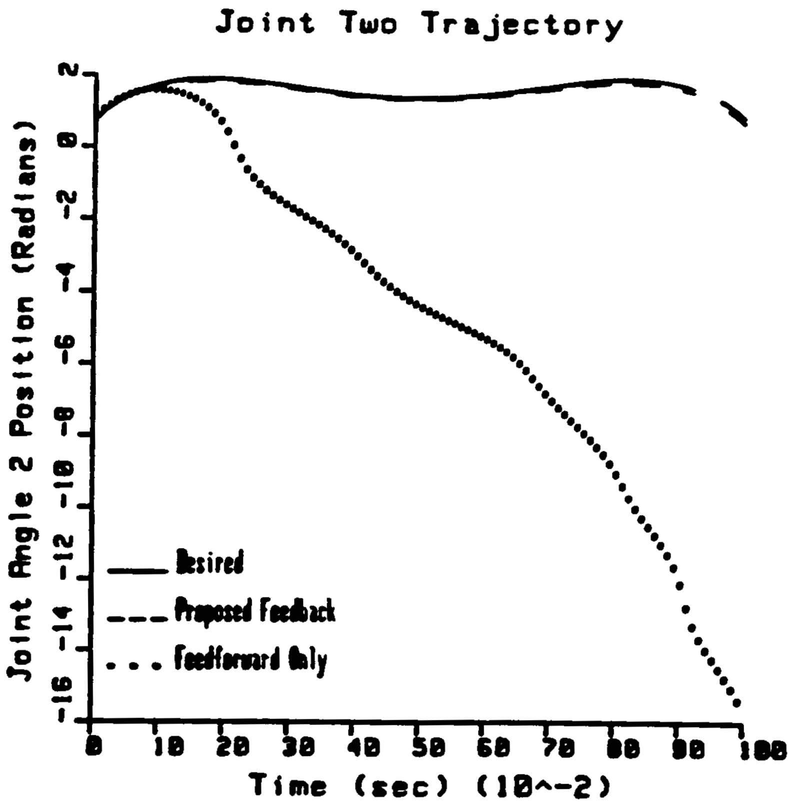


Figure 4.2 Joint trajectories with model errors

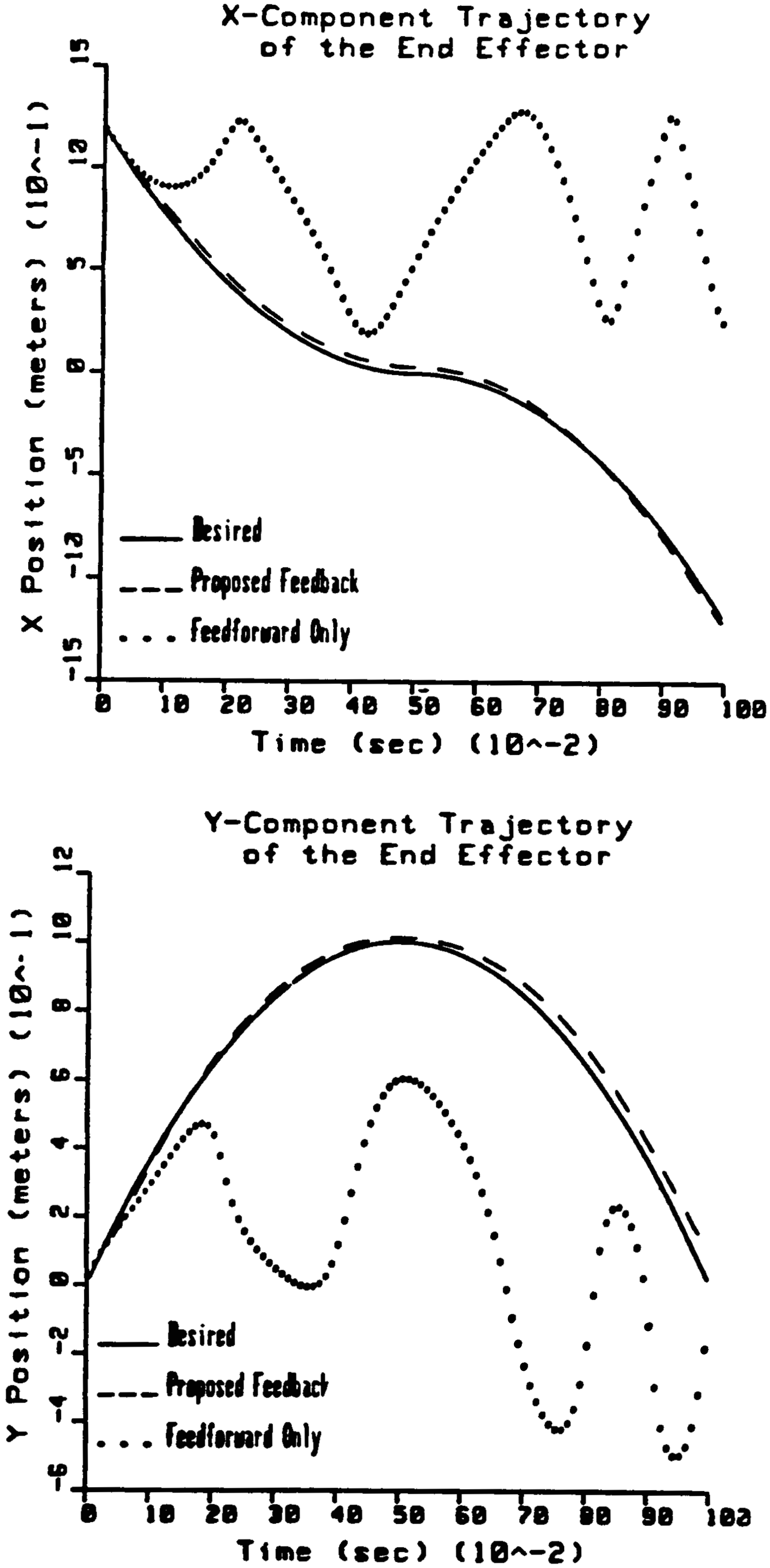


Figure 4.3 X and Y position trajectories of the end-effector with model errors

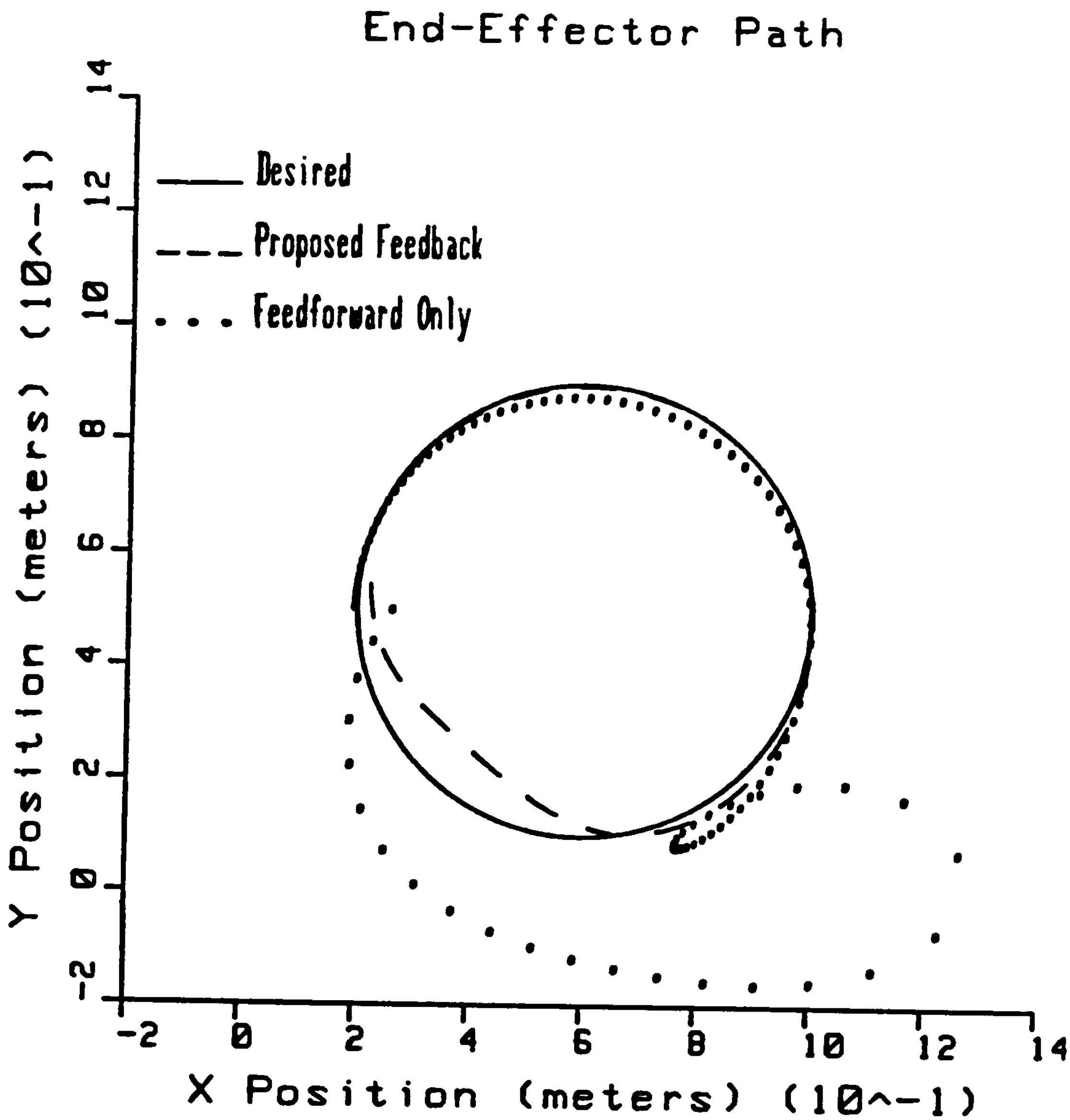


Figure 5.1 End-effector path with model errors

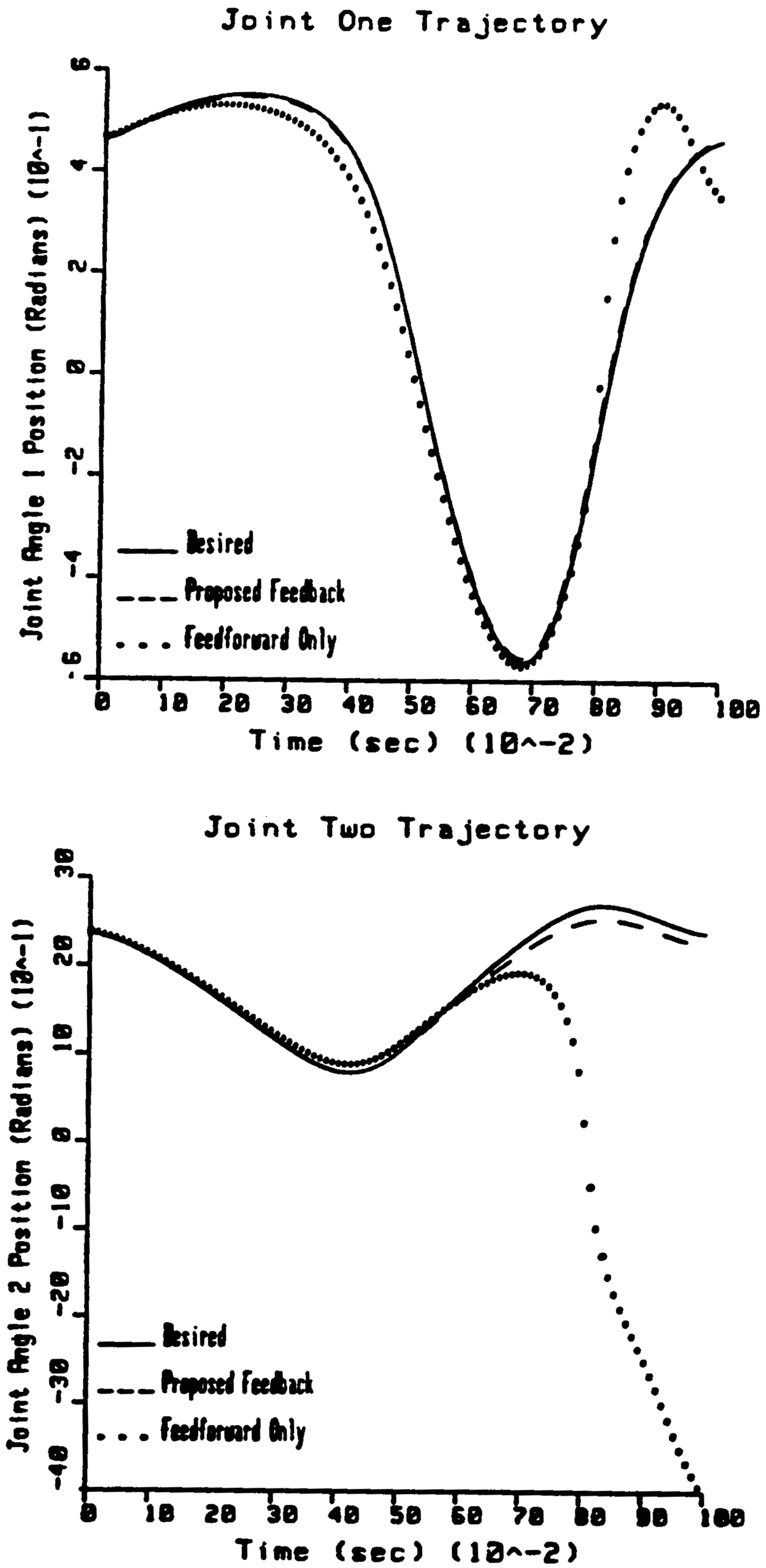
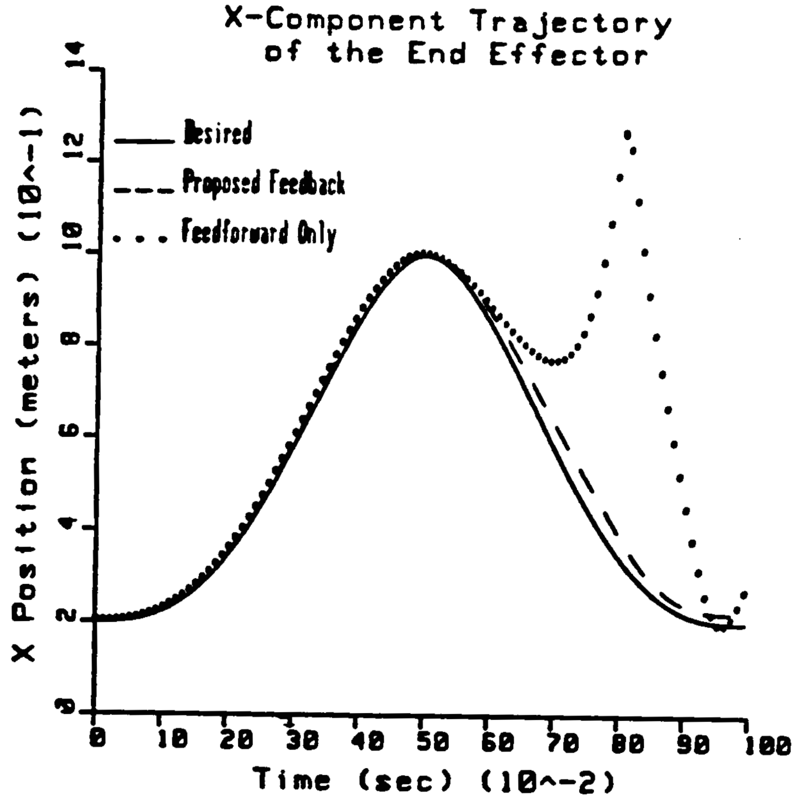


Figure 5.2 Joint trajectories with model errors



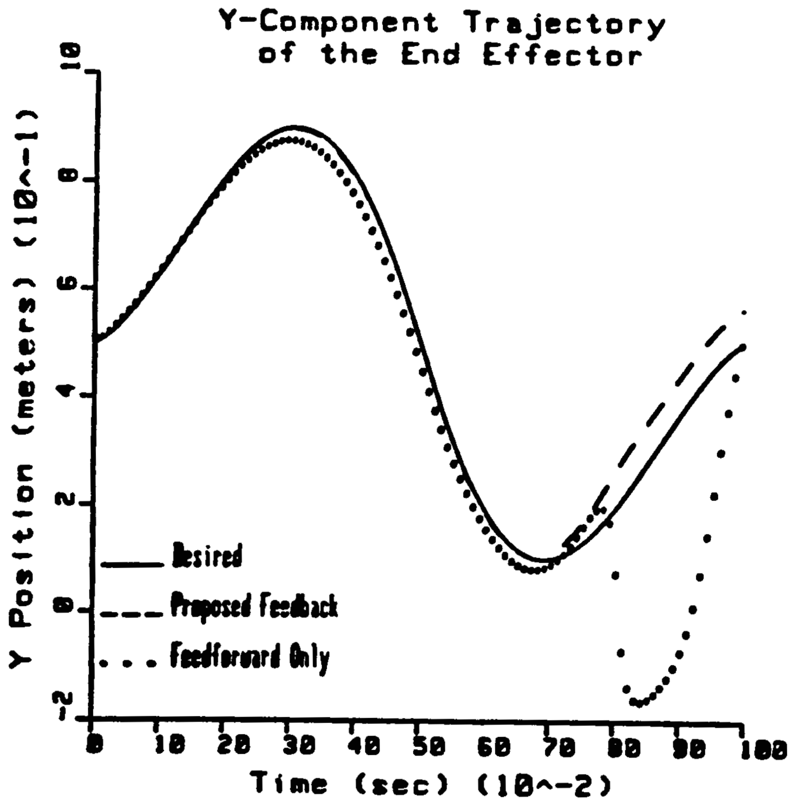


Figure 5.3 X and Y position trajectories of the end-effector with model errors

4. COMPUTATIONAL CONSIDERATIONS

The computational time that is required to update Φ, Γ, Q and K, will determine the minimum error, ϵ_1, that can be used in the control strategy and therefore determine the accuracy of the trajectory following. This update time will therefore affect the maximum sampling rate that can be used in the feedback loop when on-line trajectory generation is necessary. In many high accuracy applications, the update time will be the minimum sampling period allowed, while in other less critical situations, the use of the old gain matrix, K, during the time needed to calculate the new gain matrix, K_{new}, will not greatly affect the trajectory error. It is obvious that we want to minimize the update time so that the maximum sampling frequency is increased enough to permit good control bandwidth for the robotic manipulator.

The total computation time can be divided into three main computations:

- the feedforward, gross torque calculation,
- the calculation of the **A** and **B** matrices, and
- the updating of Φ, Γ, Q and **K**.

4.1 Feedback Controller Parameter Calculations

In the two link manipulator simulation, Sylvester's theorem [13] was used in the calculation of Φ. This theorem requires the calculation [11] of the eigenvalues of the system, and then the calculation of Φ by use of $\Phi = F_1 e^{\lambda_1 T} + F_2 e^{\lambda_2 T} + \ldots + F_N e^{\lambda_N T}$. For complex eigenvalues, Φ is written as damped sine and cosine terms, and Γ is calculated by a simple integration of these sine and cosine terms. An alternate method of Φ and Γ calculation is the use of the series expansion method. Specifically,

$$\Phi = \sum_{k=0}^{\infty} A^k T^k / k! = I + AT + \frac{1}{2!} A^2 T^2 + \ldots \tag{13}$$

$$\Gamma = \Big[\sum_{k=0}^{\infty} A^k T^{k+1} / (k+1)!\Big] B = \Big[T + \frac{AT^2}{2!} + \frac{A^2 T^3}{3!} \ldots \Big] B \tag{14}$$

This method is found to be computationally faster because the sampling period, T, is comparatively small so the higher order terms are negligible. Using an m^{th} order expansion for calculating Φ and Γ then the number of multiplications for each parametric matrix is $\sum_{i=1}^{m+1} (2n)^i$. Because the computational expense is increasing exponentially when the number of terms in the expansion is increased, so a small data sampling period, T, is very beneficial computationally.

The calculation of K

$$K = (\Gamma^T Q \Gamma)^{-1} \Gamma^T Q \Phi$$

involves matrix multiplications, transposes, and the inversion of the matrix, $(\Gamma^T Q \Gamma)$. The inversion of the matrix takes the longest to compute, and using the Gaussian elimination method, the number of operations is $O(n^3)$ for an n x n matrix. All these are standard matrix operations and codes are available to accomplish these operations in a computationally efficient manner.

The update calculation of Q is done by changing the weights of the diagonal elements in proportion to $|x_i|_{max}$, which represents the maximum deviation of any joint's position or velocity from the desired motion. It is found that in most cases, the updating of Q does not significantly affect the feedback gain matrix, K, so updating Q can be ignored if computational time is very critical.

4.2 Feedforward Computation

Many new robot applications require on-line decision making, database access, and interaction with other machines. Therefore the inverse dynamics need to be computed in real-time to obtain the gross torques of the manipulator joints, which need to be provided by the joint motors. The standard method used to derive the inverse dynamics is the standard Lagrangian formulation. Luh, Walker and Paul [10] have shown that this method would require about 7.9 seconds on the PDP 11/45 to calculate the gross torques for one position of the Stanford Arm using an efficient Fortran program. This formulation requires a computational effort of $O(n^4)$ because the method is doubly recursive with many redundant operations. The standard Lagrangian method computes the torques directly using

$$\tau_i = \sum_{j=1}^{n} \Big[\sum_{k=1}^{j} \Big(tr\Big(\frac{\partial W_j}{\partial q_i} J_j \frac{\partial W_j^T}{\partial q_k}\Big)\Big)\ddot{q}_k + \sum_{k=1}^{j}\sum_{l=1}^{j} \Big(tr\Big(\frac{\partial W_j}{\partial q_i} J_j \frac{\partial^2 W_j^T}{\partial q_k \partial q_l}\Big)\dot{q}_k \dot{q}_l\Big) - m_j g^T \frac{\partial W_j}{\partial q_i} {}^j r_j \Big]$$

The computational time for this is obviously too long, so various methods of reducing the number of computations have been tried. Since most of the computational effort is devoted to calculating the triple sums involved in the coriolis and centrifugal forces, many computation schemes ignore these terms. The problem with this is that at high speeds, the coriolis and centrifugal forces dominate in the manipulator dynamics and therefore the burden of compensation is increasingly placed on the feedback controller. While this method can work at low speeds, at high speeds this approximation could mean that excessive torques must be applied. The controller might not be capable of doing this and sometimes burnout of equipment could result. Alternative methods are available using the Newton-Euler [10] or Lagrangian [7] recursive relations. These methods yield the same torques as the standard Lagrangian approach, but are computationally faster because the standard Lagrangian approach involves redundant operations. These recursive relations reduce the computational effort required to O(n). Luh's Newton-Euler formulation in floating point assembly has been shown to take 4.5 milliseconds on the PDP 11/45 for the torque calculation of one position of the Stanford Arm. This will allow a sampling rate for the manipulator of greater than 60 Hz which insures good control bandwidth for the manipulator. The Lagrangian recursive relations are presented here because the computational formulation for the feedback gain matrix, K, is based on this approach.

4.2.1 Recursive Lagrangian Dynamics

In the following, the recursive Lagrangian dynamics procedure [7] is used to calculate the joint torques. First, all the W_i^T terms are calculated using equations (17) and going from i=1 to i=n. Then the D_i and c_i terms are computed from i=n to i=1 using the forward recursive relations (16). Finally, the torques are computed using equation (15). This formulation has 830n - 592 multiplications and 675n - 464 additions which result in 4388 multiplications and 3586 additions for n=6.

$$\tau_i = \left[tr(\frac{\partial W_i}{\partial q_i} D_i) - g^T \frac{\partial W_i}{\partial q_i} c_i \right] \quad i = 1,\dots,n \tag{15}$$

where

Forward Recursion

For i = n,...,1

$$D_i = J_i W_i^T + A_{i+1} D_{i+1} \tag{16}$$
$$c_i = m_i\, {}^i r_i + A_{i+1} c_{i+1}$$

Backwards Recursion

For i = 1,...,n

$$W_i = W_{i-1} A_i$$
$$\dot{W}_i = \dot{W}_{i-1} A_i + W_{i-1} \frac{\partial A_i}{\partial q_i} \dot{q}_i \tag{17}$$
$$\ddot{W}_i = \ddot{W}_{i-1} A_i + 2\dot{W}_{i-1} \frac{\partial A_i}{\partial q_i} \dot{q}_i + W_{i-1} \frac{\partial^2 A_i}{\partial q_i^2} \dot{q}_i^2 + W_{i-1} \frac{\partial A_i}{\partial q_i} \ddot{q}_i$$

4.3 A and B Matrix Calculations

Since the **A** and **B** matrices are based on the linearization of the manipulator dynamics about a desired trajectory, it is suggested that an efficient formulation for their computations may be based on the Lagrangian or Newton-Euler recursive relations for the solution of manipulator dynamics.

4.3.1 Derivation

Looking at the structure of the **A** and **B** matrices it is seen that three submatrices need to be calculated: $\mathbf{M}^{-1}$, $\frac{\partial \mathbf{M}}{\partial \mathbf{q}}\ddot{\mathbf{q}} + \frac{\partial \mathbf{f}}{\partial \mathbf{q}}$, and $\frac{\partial \mathbf{f}}{\partial \dot{\mathbf{q}}}$. The Lagrangian approach will be used because the formulation is much clearer and the most efficient Lagrangian relations are of the same order of computational effort as the Newton-Euler method.

The general Lagrangian formulation for the generalized forces, τ_i, for and n-link manipulator is

$$\tau_i = \sum_{j=1}^{n}\Big[\sum_{k=1}^{j} \Big(tr\Big(\frac{\partial W_j}{\partial q_i} \mathbf{J}_j \frac{\partial W_j^T}{\partial q_k}\Big)\Big)\ddot{\mathbf{q}}_k + \sum_{k=1}^{j}\sum_{l=1}^{j} \Big(tr\Big(\frac{\partial W_j}{\partial q_i} \mathbf{J}_j \frac{\partial^2 W_j^T}{\partial q_k \partial q_l}\Big)\dot{\mathbf{q}}_k\dot{\mathbf{q}}_l\Big) - m_j g^T \frac{\partial W_j}{\partial q_i}\, {}^j r_j\Big] \tag{18}$$

which also can be written in the form [12]

$$\tau_i = \sum_{j=1}^{n} D_{ij}\mathbf{q}_j + \sum_{j=1}^{n}\sum_{k=1}^{n} D_{ijk}\mathbf{q}_j\mathbf{q}_k + D_i \tag{19}$$

where

$$D_{ij} = \sum_{p=\max i,j}^{n} tr\Big(\frac{\partial W_p}{\partial q_i} J_p \frac{\partial W_p^T}{\partial q_j}\Big) \quad = \quad \textit{inertia forces}$$

$$D_{ijk} = \sum_{p=\max i,j,k}^{n} tr\Big(\frac{\partial^2 W_p}{\partial q_j \partial q_k} J_p \frac{\partial W_p^T}{\partial q_i}\Big) \quad = \quad \textit{coriolis and centripetal forces}$$

$$D_i = \sum_{p=i}^{n} -m_p g^T \frac{\partial W_p}{\partial q_i}\, {}^p r_p \quad = \quad \textit{gravity forces}$$

andwhere

$$W_j = {}^0W_j = A_1 A_2 \dots A_j$$

$${}^iW_j = A_{i+1} A_{i+2} \dots A_j \qquad\qquad i<j$$

4.3.2 Linearized Matrices

The three matrices, $\mathbf{M}^{-1}$, $\frac{\partial \mathbf{M}}{\partial \mathbf{q}}\ddot{\mathbf{q}} + \frac{\partial \mathbf{f}}{\partial \mathbf{q}}$, and $\frac{\partial \mathbf{f}}{\partial \dot{\mathbf{q}}}$, are necessary to compute results from the linearization of the inverse dynamics with respect to small perturbations, $\delta \mathbf{q}$.

(a) $\left[\frac{\partial \mathbf{M}}{\partial \mathbf{q}}\ddot{\mathbf{q}}+\frac{\partial \mathbf{f}}{\partial \mathbf{q}}\right]$ term:

The first matrix computation formulation is $\left[\frac{\partial \mathbf{M}}{\partial \mathbf{q}}\ddot{\mathbf{q}}+\frac{\partial \mathbf{f}}{\partial \mathbf{q}}\right]$. This matrix is derived by taking the partial derivative of the generalized forces with respect to the joints' position vector. So

$$\left[\frac{\partial M}{\partial q}\ddot{\mathbf{q}}+\frac{\partial f}{\partial q}\right]_{ij}=\frac{\partial}{\partial q_j}\tau_i \qquad i = 1,\dots,n, \quad j=1,\dots n \tag{20}$$

But Waters [14] proved that instead of the standard Lagrangian, the generalized forces can be expressed in a form that will permit several backward recursive relations to be derived that will reduce the computational effort to $O(n^2)$.

$$\tau_i = \sum_{p=i}^{n}\left[tr\left(\frac{\partial W_p}{\partial q_i}\mathbf{J}_p \ddot{W}_p^{T}\right) - m_p g^{T}\frac{\partial W_p}{\partial q_i}\,{}^{p}r_p\right] \qquad i=1,\dots,n \tag{21}$$

where the backward recursive relations for velocities $\dot{W}_p$ and accelerations $\ddot{W}_p$ are :

$$\begin{aligned} W_p &= W_{p-1} A_p \\ \dot{W}_p &= \dot{W}_{p-1} A_p + W_{p-1}\frac{\partial A_p}{\partial q_p}\dot{q}_p \\ \ddot{W}_p &= \ddot{W}_{p-1} A_p + 2\dot{W}_{p-1}\frac{\partial A_p}{\partial q_p}\dot{q}_p + W_{p-1}\frac{\partial^2 A_p}{\partial q_p^2}\dot{q}_p^2 + W_{p-1}\frac{\partial A_p}{\partial q_p}\ddot{q}_p \end{aligned} \tag{22}$$

Using the same formulation for the generalized forces, the derivative of the generalized forces can be expressed as

$$\left[\frac{\partial M}{\partial q}\ddot{\mathbf{q}}+\frac{\partial f}{\partial q}\right]_{ij}=\frac{\partial}{\partial q_j}\sum_{p=i}^{n}\left[tr\left(\frac{\partial W_p}{\partial q_i}\mathbf{J}_p \ddot{W}_p^{T}\right) - m_p g^{T}\frac{\partial W_p}{\partial q_i}\,{}^{p}r_p\right] \tag{23}$$

$$= \sum_{p=i}^{n}\left[tr\left(\frac{\partial^2 W_p}{\partial q_i \partial q_l}\mathbf{J}_p \ddot{W}_p^{T}\right)+tr\left(\frac{\partial W_p}{\partial q_i}\mathbf{J}_p \frac{\partial \ddot{W}_p^{T}}{\partial q_j}\right)-m_p g^{T}\frac{\partial^2 W_p}{\partial q_i \partial q_j}\,{}^{p}r_p\right] \tag{24}$$

Now

$$\textit{if} \quad j < p \quad \textit{then} \quad \frac{\partial}{\partial q_j}\left(\frac{\partial W_p}{\partial q_i}\right) = 0$$

$$\textit{and} \qquad \frac{\partial \ddot{W}_p}{\partial q_j} = 0$$

$$\textit{since} \qquad W_p = {}^0W_p = A_1 A_2 \ldots A_p$$

Consequently, the matrix formulation can be written as

$$\left[\frac{\partial M}{\partial q}\ddot{q} + \frac{\partial f}{\partial q}\right]_{ij} = \sum_{p=\max i,j}^{n} \left[tr\left(\frac{\partial^2 W_p}{\partial q_i \partial q_j} J_p \ddot{W}_p^T\right) + tr\left(\frac{\partial W_p}{\partial q_i} J_p \frac{\partial \ddot{W}_p^T}{\partial q_j}\right) - m_p g^T \frac{\partial^2 W_p}{\partial q_i \partial q_j} {}^p r_p \right] \tag{25}$$

for i = 1,...,n and j = 1,...,n

Now a forward recursive relation can be developed by noting that

$$\frac{\partial W_p}{\partial q_i} = \frac{\partial W_i}{\partial q_i} {}^iW_p$$

$$\textit{where} \quad {}^iW_p = A_{i+1} A_{i+2} \ldots A_p \qquad\qquad i \le p$$

Therefore for the two cases of the double derivative we obtain

if i > j

$$\frac{\partial^2 W_p}{\partial q_i \partial q_j} = \frac{\partial}{\partial q_j}\left(\frac{\partial W_i}{\partial q_i} {}^iW_p\right)$$

$$= \frac{\partial^2 W_i}{\partial q_i \partial q_j} {}^iW_p + \frac{\partial W_i}{\partial q_i} \frac{\partial {}^iW_p}{\partial q_j}$$

$$= \frac{\partial^2 W_i}{\partial q_i \partial q_j} {}^iW_p$$

Similarly for j > i

$$\frac{\partial^2 W_p}{\partial q_i \partial q_j} = \frac{\partial^2 W_j}{\partial q_i \partial q_j} {}^jW_p$$

Because of the symmetry of the equations of the double derivative, only the case i ≥ j will be considered in what follows.

Rewriting the matrix formulation as

$$\left[\frac{\partial M}{\partial q}\ddot{q}+\frac{\partial f}{\partial q}\right]_{ij}=\sum_{p=i}^{n}\left[tr\left(\frac{\partial^2 W_i}{\partial q_i \partial q_j}\,{}^iW_p J_p\, \ddot{W}_p^T\right)+tr\left(\frac{\partial W_i}{\partial q_i}\,{}^iW_p J_p\,\frac{\partial \ddot{W}_p^T}{\partial q_j}\right)-m_p g^T \frac{\partial^2 W_i}{\partial q_i \partial q_j}\,{}^iW_p\,{}^pr_p\right] \qquad (26)$$

$$\left[\frac{\partial M}{\partial q}\ddot{q}+\frac{\partial f}{\partial q}\right]_{ij}=\sum_{p=i}^{n}\left[tr\left(\frac{\partial^2 W_i}{\partial q_i \partial q_j}\,{}^iW_p J_p\, \ddot{W}_p^T\right)+tr\left(\frac{\partial W_i}{\partial q_i}\,{}^iW_p J_p\,\frac{\partial \ddot{W}_p^T}{\partial q_j}\right)-m_p g^T \frac{\partial^2 W_i}{\partial q_i \partial q_j}\,{}^iW_p\,{}^pr_p\right] \qquad (27)$$

then the reformulation can be written as

$$\left[\frac{\partial M}{\partial q}\ddot{q}+\frac{\partial f}{\partial q}\right]_{ij}=$$

$$\left[tr\left(\frac{\partial^2 W_i}{\partial q_i \partial q_j}\sum_{p=i}^{n}{}^iW_p J_p \ddot{W}_p^T\right)+tr\left(\frac{\partial W_i}{\partial q_i}\sum_{p=i}^{n}{}^iW_p J_p\,\frac{\partial \ddot{W}_p^T}{\partial q_j}\right)-g^T \frac{\partial^2 W_i}{\partial q_i \partial q_j}\sum_{p=i}^{n} m_p\,{}^iW_p\,{}^pr_p\right] \qquad (28)$$

Let

$$D_i = \sum_{p=i}^{n} {}^iW_p J_p \ddot{W}_p^T$$

$$= {}^iW_i J_i \ddot{W}_i^T + \sum_{p=i+1}^{n} A_{i+1}\ {}^{i+1}W_p J_p \ddot{W}_p^T$$

Now since $\quad {}^iW_i = I$

$$\text{we get } D_i = J_i \ddot{W}_i^T + A_{i+1}\ D_{i+1} \qquad (29)$$

Also, let

$$c_i = \sum_{p=i}^{n} m_p\ {}^iW_p\ {}^pr_p$$

$$c_i = m_i\ {}^ir_i + A_{i+1}\ c_{i+1} \qquad (30)$$

and

$$N_i = \sum_{p=i}^{n} {}^iW_p J_p \frac{\partial \ddot{W}_p^T}{\partial q_j}$$

$$N_i = J_i \frac{\partial \ddot{W}_i^T}{\partial q_j} + A_{i+1}\ N_{i+1} \qquad (31)$$

Now for $i \geq j$ the matrix is simply written as

$$\left[\frac{\partial M}{\partial q}\ddot{q}+\frac{\partial f}{\partial q}\right]_{ij}=\left[tr\left(\frac{\partial^2 W_i}{\partial q_i \partial q_j} D_i\right)+ tr\left(\frac{\partial W_i}{\partial q_i} N_i\right) - g^T \frac{\partial^2 W_i}{\partial q_i \partial q_j} c_i\right] \qquad (32)$$

By a similar procedure we get

for $j \geq i$

$$\left[\frac{\partial M}{\partial q}\ddot{q}+\frac{\partial f}{\partial q}\right]_{ij}=\left[tr\left(\frac{\partial^2 W_j}{\partial q_i \partial q_j} D_j\right)+ tr\left(\frac{\partial W_j}{\partial q_i} N_j\right) - g^T \frac{\partial^2 W_j}{\partial q_i \partial q_j} c_j\right] \qquad (33)$$

where

$$D_j = J_j \ddot{W}_j^T + A_{j+1} D_{j+1} \tag{34}$$

$$c_j = m_j {}^j r_j + A_{j+1} c_{j+1} \tag{35}$$

$$N_j = J_j \frac{\partial \ddot{w}_j^T}{\partial q_j} + A_{j+1} N_{j+1} \tag{36}$$

(b) $\left[\frac{\partial f}{\partial \dot{q}}\right]$ term:

Using a procedure similar to what is given in the previous section, the $\left[\frac{\partial f}{\partial \dot{q}}\right]$ term can be simply formulated as a set of linear recursive backward and forward relations. This matrix term is derived by taking the partial derivative of the generalized forces with respect to the joints' velocity vector. So

$$\left[\frac{\partial f}{\partial \dot{q}}\right]_{ij} = \frac{\partial}{\partial \dot{q}_j} \tau_i \qquad i = 1,\dots,n, \qquad j=1,\dots n \tag{37}$$

Now using Waters generalized forces formulation, the matrix becomes

$$\left[\frac{\partial f}{\partial \dot{q}}\right]_{ij} = \sum_{p=i}^{n} tr\left(\frac{\partial W_p}{\partial q_i} J_p \frac{\partial \ddot{W}_p^T}{\partial \dot{q}_j}\right) \tag{38}$$

$$If \; j > p \; then \; \frac{\partial \ddot{W}_p}{\partial q_j} = 0$$

Consequently the matrix equations are written as

$$\left[\frac{\partial f}{\partial \dot{q}}\right]_{ij} = \sum_{p=\max i,j}^{n} tr\left(\frac{\partial W_p}{\partial q_i} J_p \frac{\partial \ddot{W}_p^T}{\partial \dot{q}_j}\right) \tag{39}$$

Consider first the case of

If $i > j$

$$\left[\frac{\partial f}{\partial \dot{q}}\right]_{ij} = \sum_{p=i}^{n} tr\left(\frac{\partial W_i}{\partial q_i} {}^iW_p J_p \frac{\partial \ddot{W}_p^T}{\partial \dot{q}_j}\right) \tag{40}$$

$$\left[\frac{\partial f}{\partial \dot{q}}\right]_{ij} = tr\left(\frac{\partial W_i^T}{\partial q_i} \sum_{p=i}^{n} {}^iW_p J_p \frac{\partial \ddot{W}_p^T}{\partial \dot{q}_j}\right) \tag{41}$$

Now, it can be shown that

$$\frac{\partial \ddot{W}_p^T}{\partial \dot{q}_j} = \frac{\partial \dot{W}_p^T}{\partial q_j} \tag{42}$$

Which leads to the reformulation

$$\left[\frac{\partial f}{\partial \dot{q}}\right]_{ij} = tr\left(\frac{\partial W_i^T}{\partial q_i} \sum_{p=i}^{n} {}^{i}W_p \, \mathbf{J}_p \, \frac{\partial \dot{W}_p^T}{\partial q_j}\right) \tag{43}$$

that produces the forward recursive relation by letting

$$Q_i = \sum_{p=i}^{n} {}^{i}W_p \, \mathbf{J}_p \, \frac{\partial \dot{W}_p^T}{\partial q_j}$$

$$Q_i = A_{i+1} \, Q_{i+1} + \mathbf{J}_i \, \frac{\partial \dot{W}_i^T}{\partial q_j} \tag{44}$$

So the matrix compuation is simply formulated as

$$\left[\frac{\partial f}{\partial \dot{q}}\right]_{ij} = tr\left(\frac{\partial W_i}{\partial q_i} \, Q_i\right) \tag{45}$$

Considering the other case and by applying similar arguments we get

for $j \geq i$

$$\left[\frac{\partial f}{\partial \dot{q}}\right]_{ij} = \sum_{p=j}^{n} tr\left(\frac{\partial W_p}{\partial q_i} \, \mathbf{J}_p \, \frac{\partial \ddot{W}_p^T}{\partial \dot{q}_j}\right)$$

$$\left[\frac{\partial f}{\partial \dot{q}}\right]_{ij} = tr\left(\frac{\partial W_i^T}{\partial q_i} \, {}^{i}W_j \sum_{p=j}^{n} {}^{j}W_p \, \mathbf{J}_p \, \frac{\partial \dot{W}_p^T}{\partial q_j}\right) \tag{46}$$

Then the matrix is formulated as

$$\left[\frac{\partial f}{\partial \dot{q}}\right]_{ij} = tr\left(\frac{\partial W_i}{\partial q_i} \, {}^{i}W_j \, Q_j\right) \tag{47}$$

$$Q_j = A_{j+1} \, Q_{j+1} + \mathbf{J}_j \, \frac{\partial \dot{W}_j^T}{\partial q_j} \tag{48}$$

(c) $\mathbf{M}_{ij}$ term:

The next matrix to be calculated is the inertia matrix, **M**. The recursive relations are derived in the same manner as the other matices. Specifically,

$$M_{ij} = D_{ij} = \sum_{p=\max i,j}^{n} tr(\frac{\partial W_p}{\partial q_i} J_p \frac{\partial W^T_p}{\partial q_j}) \tag{49}$$

For $i \geq j$

$$M_{ij} = \sum_{p=i}^{n} tr(\frac{\partial W_p}{\partial q_i} {}^iW_p \mathbf{J}_p \frac{\partial W^T_p}{\partial q_j}) \tag{50}$$

$$M_{ij} = tr(\frac{\partial W_i}{\partial q_i} \sum_{p=i}^{n} {}^iW_p \mathbf{J}_p \frac{\partial W^T_p}{\partial q_j}) \tag{51}$$

the forward recursive relation is

$$P_i = \sum_{p=i}^{n} {}^iW_p \mathbf{J}_p \frac{\partial W^T_p}{\partial q_j})$$

$$P_i = A_{i+1} P_{i+1} + \mathbf{J}_i \frac{\partial W_i^T}{\partial q_j} \tag{52}$$

and the matrix is computed simply by

$$M_{ij} = tr(\frac{\partial W_i}{\partial q_i} P_i) \tag{53}$$

for $j \geq i$

$$M_{ij} = \sum_{p=j}^{n} tr(\frac{\partial W_i}{\partial q_i} {}^iW_j {}^iW_p \mathbf{J}_p \frac{\partial W^T_p}{\partial q_j}) \tag{54}$$

In this case the matrix formulation and forward recursive relations are

$$M_{ij} = tr(\frac{\partial W_i}{\partial q_i} {}^iW_j P_j) \tag{55}$$

$$P_j = A_{j+1} P_{j+1} + \mathbf{J}_j \frac{\partial W_j^T}{\partial q_j} \tag{56}$$

The last terms that need to be calculated are the $\frac{\partial \dot{w}_p}{\partial q_j}$ and $\frac{\partial \ddot{w}_p}{\partial q_j}$ terms. The backward linear recursive relations needed to calculate these terms are now presented.

(d) $\frac{\partial \mathbf{w}_p}{\partial \mathbf{q}_j}$ term:

For $p \geq j$

$$\frac{\partial W_p}{\partial q_j} = \frac{\partial W_j}{\partial q_j}\,{}^{j}W_p$$
$$\frac{\partial \dot{W}_p}{\partial q_j} = \frac{\partial \dot{W}_j}{\partial q_j}\,{}^{j}W_p + \frac{\partial W_j}{\partial q_j}\,{}^{j}\dot{W}_p \tag{57}$$
$$\frac{\partial \ddot{W}_p}{\partial q_j} = \frac{\partial \ddot{W}_j}{\partial q_j}\,{}^{j}W_p + 2\frac{\partial \dot{W}_j}{\partial q_j}\,{}^{j}W_p + \frac{\partial \ddot{W}_j}{\partial q_j}\,{}^{j}W_p$$

and for $j \geq p$

$${}^{j}W_p = {}^{j}W_{p-1}\,A_p$$
$${}^{j}\dot{W}_p = {}^{j}\dot{W}_{p-1}\,A_p + {}^{j}W_{p-1}\frac{\partial A_p}{\partial q_p}\,\dot{q}_p \tag{58}$$
$${}^{j}\ddot{W}_p = {}^{j}\ddot{W}_{p-1}\,A_p + 2\,{}^{j}\dot{W}_{p-1}\frac{\partial A_p}{\partial q_p}\,\dot{q}_p + {}^{j}W_{p-1}\frac{\partial^2 A_p}{\partial q_p^{\,2}}\,\dot{q}_p^{\,2} + {}^{j}W_{p-1}\frac{\partial A_p}{\partial q_p}\,\ddot{q}_p$$

For $j = 1,\ldots,n$

$$\dot{W}_j = \dot{W}_{j-1}\,A_j + W_{j-1}\frac{\partial A_j}{\partial q_j}\,\dot{q}_j$$
$$\frac{\partial \dot{W}_j}{\partial q_j} = \dot{W}_{j-1}\,\frac{\partial A_j}{\partial q_j} + W_{j-1}\frac{\partial^2 A_j}{\partial^2 q_j}\,\dot{q}_j \tag{59}$$

$$\ddot{W}_j = \ddot{W}_{j-1}\,A_j + 2\dot{W}_{j-1}\frac{\partial A_j}{\partial q_j}\,q_j + W_{j-1}\frac{\partial^2 A_j}{\partial q_j^2}\,\dot{q}_j^{\,2} + W_{j-1}\frac{\partial A_j}{\partial q_j}\,\ddot{q}_j$$
$$\frac{\partial \ddot{W}_j}{\partial q_j} = \ddot{W}_{j-1}\,\frac{\partial A_j}{\partial q_j} + 2\dot{W}_{j-1}\frac{\partial^2 A_j}{\partial q_j^2}\,\dot{q}_j + W_{j-1}\frac{\partial^3 A_j}{\partial q_j^3}\,\dot{q}_j^{\,2} + W_{j-1}\frac{\partial^2 A_j}{\partial q_j^2}\,\ddot{q}_j \tag{60}$$

Note also that

$$\frac{\partial W_p}{\partial q_p} = W_{p-1}\frac{\partial A_p}{\partial q_p} \tag{61}$$

$j = 1,\ldots,p-1$

$$\frac{\partial^2 W_p}{\partial q_p \partial q_j} = W_{j-1}\frac{\partial A_j}{\partial q_j}\,{}^{j}W_{p-1}\frac{\partial A_p}{\partial q_p} \tag{62}$$

$$\frac{\partial^2 W_p}{\partial q_p^{\,2}} = W_{p-1}\frac{\partial^2 A_p}{\partial q_p^{\,2}} \qquad \textit{for } j = p \tag{63}$$

4.4 The Summary of Recursive Relations

Now, to summarize the procedure for computing the $\mathbf{M}^{-1}$, $\frac{\partial \mathbf{M}}{\partial \mathbf{q}}\ddot{\mathbf{q}} + \frac{\partial \mathbf{f}}{\partial \mathbf{q}}$, and $\frac{\partial \mathbf{f}}{\partial \dot{\mathbf{q}}}$, matrices. First, the backward recursive relations (64) are used to compute all the W_i^T terms from i = 1 to i=n. Then all the $\frac{\partial w_i^T}{\partial q_j}$, $\frac{\partial \dot{w}_i^T}{\partial q_j}$, $\frac{\partial \ddot{w}_i^T}{\partial q_j}$ terms are computed by the recursive relations (65), (66) and (67) for i=1 to i=n and j=1 to j=n, but only for the cases of $i \geq j$. Next the forward recursive relations (68) and (69) are used to calculate D_i, and c_i for i=n to i=1, and relations (70), (71) and (72) are used to calculate P_{ij}, Q_{ij}, N_{ij} for j=1 to j=i. Finally, the necessary control matrices, $\mathbf{M}^{-1}$, $\frac{\partial \mathbf{M}}{\partial \mathbf{q}}\ddot{\mathbf{q}} + \frac{\partial \mathbf{f}}{\partial \mathbf{q}}$, and $\frac{\partial \mathbf{f}}{\partial \dot{\mathbf{q}}}$ are computed by (73), (74), (75), (76), (77) and (78) for i=1 to i=n and j=1 to j=n. Noting that many of the terms are the same as those calculated for the feedforward computations if the feedforward calculation is incorporated in the control loop, then many of these computations need not be repeated.

4.4.1 Backwards Recursion

For i = 1,...,n

$$W_i = W_{i-1}\,A_i$$

$$\dot{W}_i = \dot{W}_{i-1}\,A_i + W_{i-1}\frac{\partial A_i}{\partial q_i}\dot{q}_i \tag{64}$$

$$\ddot{W}_i = \ddot{W}_{i-1}\,A_i + 2\dot{W}_{i-1}\frac{\partial A_i}{\partial q_i}\dot{q}_i + W_{i-1}\frac{\partial^2 A_i}{\partial q_i^{\,2}}\dot{q}_i^{\,2} + W_{i-1}\frac{\partial A_i}{\partial q_i}\ddot{q}_i$$

For <u>j ≤ i</u>

$$
\begin{aligned}
{}^{j}W_{i} &= {}^{j}W_{i-1}\, A_{i} \\
{}^{j}\dot{W}_{i} &= {}^{j}\dot{W}_{i-1}\, A_{i} + {}^{j}W_{i-1}\frac{\partial A_{i}}{\partial q_{i}}\,\dot{q}_{i} \\
{}^{j}\ddot{W}_{i} &= {}^{j}\ddot{W}_{i-1}\, A_{i} + 2\,{}^{j}\dot{W}_{i-1}\frac{\partial A_{i}}{\partial q_{i}}\,\dot{q}_{i} + {}^{j}W_{i-1}\frac{\partial^{2} A_{i}}{\partial q_{i}^{2}}\,\dot{q}_{i}^{2} + {}^{j}W_{i-1}\frac{\partial A_{i}}{\partial q_{i}}\,\ddot{q}_{i}
\end{aligned}
\tag{65}
$$

For <u>j = 1,...,n</u>

$$
\begin{aligned}
\frac{\partial \dot{W}_{j}}{\partial q_{j}} &= \dot{W}_{j-1}\,\frac{\partial A_{j}}{\partial q_{j}} + W_{j-1}\frac{\partial^{2} A_{j}}{\partial^{2} q_{j}}\,\dot{q}_{j} \\
\frac{\partial \dot{W}_{j}}{\partial q_{j}} &= \ddot{W}_{j-1}\,\frac{\partial A_{j}}{\partial q_{j}} + 2\dot{W}_{j-1}\frac{\partial^{2} A_{j}}{\partial q_{j}^{2}}\,\dot{q}_{j} + W_{j-1}\frac{\partial^{3} A_{j}}{\partial q_{j}^{3}}\,\dot{q}_{j}^{2} + W_{j-1}\frac{\partial^{2} A_{j}}{\partial q_{j}^{2}}\,\ddot{q}_{j}
\end{aligned}
\tag{66}
$$

For <u>i ≥ j</u>

$$
\begin{aligned}
\frac{\partial W_{i}}{\partial q_{j}} &= \frac{\partial W_{j}}{\partial q_{j}}\,{}^{j}W_{i} \\
\frac{\partial \dot{W}_{i}}{\partial q_{j}} &= \frac{\partial \dot{W}_{j}}{\partial q_{j}}\,{}^{j}W_{i} + \frac{\partial W_{j}}{\partial q_{j}}\,{}^{j}\dot{W}_{i} \\
\frac{\partial \ddot{W}_{i}}{\partial q_{j}} &= \frac{\partial \ddot{W}_{j}}{\partial q_{j}}\,{}^{j}W_{i} + \frac{\partial \dot{W}_{j}}{\partial q_{j}}\,{}^{j}\dot{W}_{i} + \frac{\partial W_{j}}{\partial q_{j}}\,{}^{j}\ddot{W}_{i}
\end{aligned}
\tag{67}
$$

4.4.2 Forward Recursion

For i = n,...,1

$$D_i = J_i \ddot{W}_i^T + A_{i+1} D_{i+1} \tag{68}$$

$$c_i = m_i \, {}^i r_i + A_{i+1} c_{i+1} \tag{69}$$

For j = 1,...,i

$$P_{ij} = A_{i+1} P_{i+1j} + J_i \frac{\partial W_i^T}{\partial q_j} \tag{70}$$

$$Q_{ij} = A_{i+1} Q_{i+1j} + J_i \frac{\partial \dot{W}_i^T}{\partial q_j} \tag{71}$$

$$N_{ij} = A_{i+1} N_{i+1j} + J_i \frac{\partial \ddot{W}_i^T}{\partial q_j} \tag{72}$$

For i=1,...,n , j=1,...,n

(a) $\mathbf{M}_{ij}$ term:

For $i \geq j$

$$M_{ij} = tr(\frac{\partial W_i}{\partial q_i} P_{ij}) \tag{73}$$

For $j \geq i$

$$M_{ij} = tr(\frac{\partial W_i}{\partial q_i} \, {}^i W_j \, P_{ii}) \tag{74}$$

(*b*) $\left[\frac{\partial \mathbf{M}}{\partial \mathbf{q}} \ddot{\mathbf{q}} + \frac{\partial \mathbf{f}}{\partial \mathbf{q}}\right]$ *term:*

If $i > j$

$$\left[\frac{\partial M}{\partial q}\ddot{q}+\frac{\partial f}{\partial q}\right]_{ij}=\left[tr\left(\frac{\partial^2 W_i}{\partial q_i \partial q_j} D_i\right)+ tr\left(\frac{\partial W_i}{\partial q_i} N_{ij}\right) - g^T \frac{\partial^2 W_i}{\partial q_i \partial q_j} c_i\right] \tag{75}$$

If $j \geq i$

$$\left[\frac{\partial M}{\partial q}\ddot{q}+\frac{\partial f}{\partial q}\right]_{ij}=\left[tr\left(\frac{\partial^2 W_i}{\partial q_i \partial q_j} D_i\right)+ tr\left(\frac{\partial W_i}{\partial q_i} N_{ii}\right) - g^T \frac{\partial^2 W_i}{\partial q_i \partial q_j} c_i\right] \tag{76}$$

(c) $\left[\frac{\partial f}{\partial \dot{q}}\right]$ *term:*

for $i \geq j$

$$\left[\frac{\partial f}{\partial \dot{q}}\right]_{ij} = tr\left(\frac{\partial W_i}{\partial q_i} Q_{ij}\right) \tag{77}$$

for $j > i$

$$\left[\frac{\partial f}{\partial \dot{q}}\right]_{ij} = tr\left(\frac{\partial W_i}{\partial q_i} {}^{i}W_j Q_{ii}\right) \tag{78}$$

The number of multiplications involved with the matrix calculations is $1062n^2 - 1021n - 128$ and the number of additions is $1037n^2 - 621n - 96$. This means that for n=6, the number of multiplications is 40,594 and the number of additions is 37,926 for each update of the **A** and **B** matrices. Therefore, the number of multiplications and additions is of n^2 dependence and for n=6 the number of operations is 10 times the number of operations involved in the recursive Lagrangian dynamics relations.

4.5 Recursive Parametric Matrices Using 3 x 3 Matrices

The previous formulation reduces the computational effort to $O(n^2)$ for each matrix, which is the lowest order that can be achieved. The only way to further reduce the computational cost is to use 3 x 3 rotation matrices instead of 4 x 4 rotation-translation matrices. The 4 x 4 matrices are inefficient because of some sparseness and because of the combination of translation with rotation [7]. The 4 x 4 matrices require 64 multiplications for each matrix multiplication, while 3 x 3 matrices only require 27 multiplications, so a 58% reduction in coefficient multiplications is effected.

The 3 x 3 rotation matrix A_j relates the orientations of coordinate systems j-1 and j, and W_j

and ${}^{i}W_j$ are defined as before except the A matrices are 3 x 3. Using these relations, the derivation of the formulations for computing $\mathbf{M}^{-1}$, $\frac{\partial \mathbf{M}}{\partial \mathbf{q}}\ddot{\mathbf{q}} + \frac{\partial \mathbf{f}}{\partial \mathbf{q}}$, and $\frac{\partial \mathbf{f}}{\partial \dot{\mathbf{q}}}$, using 3x3 matrices is presented in Appendix B. The procedure for calculating the $\mathbf{M}^{-1}$, $\frac{\partial \mathbf{M}}{\partial \mathbf{q}}\ddot{\mathbf{q}} + \frac{\partial \mathbf{f}}{\partial \mathbf{q}}$, and $\frac{\partial \mathbf{f}}{\partial \dot{\mathbf{q}}}$, matrices using 3 x 3 rotation matrices is now summarized. First, the backward relations (64), (65), (66), (67) and (79) are used to compute all the $\frac{\partial w_i^T}{\partial q_j}$, $\frac{\partial \dot{w}_i^T}{\partial q_j}$, $\frac{\partial \ddot{w}_i^T}{\partial q_j}$, and the $\frac{\partial p_i^T}{\partial q_j}$, $\frac{\partial \dot{p}_i^T}{\partial q_j}$, $\frac{\partial \ddot{p}_i^T}{\partial q_j}$, terms for i=1 to i=n and j=1 to j=1. Next, the forward recursive relations (80), (81) and (82) are used to calculate D_i, e_i and c_i for i=n to i=1, and relations (83), (84), (85), (86), (87) and (88) are used to calculate P_{ij}, k_{ij}, Q_{ij}, b_{ij}, N_{ij}, I_{ij}, for j=1 to j=i. Finally, the necessary control matrices, $\mathbf{M}^{-1}$, $\frac{\partial \mathbf{M}}{\partial \mathbf{q}}\ddot{q} + \frac{\partial \mathbf{f}}{\partial \mathbf{q}}$, and $\frac{\partial \mathbf{f}}{\partial \dot{\mathbf{q}}}$ are computed by (89), (90), (91), (92), (93) and (94) for i=1 to i=n and j=1 to j=n.

4.5.1 Backwards Recursion

The $\frac{\partial W_i^T}{\partial q_j}$, $\frac{\partial \dot{W}_i^T}{\partial q_j}$, $\frac{\partial \ddot{W}_i^T}{\partial q_j}$ terms are calculated with the same recurrence relations (64), (65), (66) and (67) as before except the matrices are now 3 x 3.

For i = 1,..,n

$$p_i = p_{i-1} - W_i\, {}^{i}p_i^*$$

For j = 1,...,i

$$\frac{\partial p_i}{\partial q_j} = \frac{\partial p_{i-1}}{\partial q_j} - \frac{\partial W_i}{\partial q_j}\, {}^{i}p_i^*$$

$$\frac{\partial \dot{p}_i}{\partial q_j} = \frac{\partial \dot{p}_{i-1}}{\partial q_j} - \frac{\partial \dot{W}_i}{\partial q_j}\, {}^{i}p_i^*$$

$$\frac{\partial \ddot{p}_i}{\partial q_j} = \frac{\partial \ddot{p}_{i-1}}{\partial q_j} - \frac{\partial \ddot{W}_i}{\partial q_j}\, {}^{i}p_i^* \tag{79}$$

4.5.2 Forward Recursion

For i = n,...,1

$$D_i = J_i \ddot{W}_i^T + {}^i n_i^T p_i^T + A_{i+1} D_{i+1} + {}^i p_{i+1} e_{i+1} \tag{80}$$

$$e_i = e_{i+1} + m_i \ddot{p}_i^T + {}^i n_i^T \ddot{W}_i^T \tag{81}$$

$$c_i = m_i \, {}^i r_i + A_{i+1} c_{i+1} \tag{82}$$

For j = 1,...,i

$$P_{ij} = A_{i+1} P_{i+1j} + {}^i p_{i+1} k_{i+1j} + {}^i n_i^T \frac{\partial p_i^T}{\partial q_j} + J_i \frac{\partial W_i^T}{\partial q_j} \tag{83}$$

$$k_{ij} = k_{i+1j} + m_i \frac{\partial p_i^T}{\partial q_j} + {}^i n_i^T {}_i \frac{\partial W_i^T}{\partial q_j} \tag{84}$$

$$Q_{ij} = A_{i+1} Q_{i+1j} + {}^i p_{i+1} b_{i+1j} + {}^i n_i^T \frac{\partial \dot{p}_i^T}{\partial q_j} + J_i \frac{\partial \dot{W}_i^T}{\partial q_j} \tag{85}$$

$$b_{ij} = b_{i+1j} + m_i \frac{\partial \dot{p}_i^T}{\partial q_j} + {}^i n_i^T {}_i \frac{\partial \dot{W}_i^T}{\partial q_j} \tag{86}$$

$$N_{ij} = A_{i+1} N_{i+1j} + {}^i p_{i+1} l_{i+1j} + {}^i n_i^T \frac{\partial \ddot{p}_i^T}{\partial q_j} + J_i \frac{\partial \ddot{W}_i^T}{\partial q_j} \tag{87}$$

$$l_{ij} = l_{i+1j} + m_i \frac{\partial \ddot{p}_i^T}{\partial q_j} + {}^i n_i^T {}_i \frac{\partial \ddot{W}_i^T}{\partial q_j} \tag{88}$$

For i=1,...,n , j=1,..,n

(a) $\mathbf{M}_{ij}$ term:

For $i \geq j$

$$M_{ij} = tr\left(\frac{\partial W_i}{\partial q_i} P_{ij}\right) \tag{89}$$

For $j \geq i$

$$M_{ij} = tr(\frac{\partial W_i}{\partial q_i} \, {}^iW_j \, P_{ii}) \tag{90}$$

(b) $\left[\frac{\partial \mathbf{M}}{\partial \mathbf{q}} \ddot{\mathbf{q}} + \frac{\partial \mathbf{f}}{\partial \mathbf{q}}\right]$ *term:*

If $i > j$

$$\left[\frac{\partial M}{\partial q} \ddot{q} + \frac{\partial f}{\partial q}\right]_{ij} = \left[tr(\frac{\partial^2 W_i}{\partial q_i \partial q_j} D_i) + tr(\frac{\partial W_i}{\partial q_i} N_{ij}) - g^T \frac{\partial^2 W_i}{\partial q_i \partial q_j} c_i\right] \tag{91}$$

If $j \geq i$

$$\left[\frac{\partial M}{\partial q} \ddot{q} + \frac{\partial f}{\partial q}\right]_{ij} = \left[tr(\frac{\partial^2 W_i}{\partial q_i \partial q_j} D_i) + tr(\frac{\partial W_i}{\partial q_i} N_{ii}) - g^T \frac{\partial^2 W_i}{\partial q_i \partial q_j} c_i\right] \tag{92}$$

(c) $\left[\frac{\partial \mathbf{f}}{\partial \dot{\mathbf{q}}}\right]$ *term:*

for $i \geq j$

$$\left[\frac{\partial f}{\partial \dot{q}}\right]_{ij} = tr(\frac{\partial W_i}{\partial q_i} Q_{ij}) \tag{93}$$

for $j > i$

$$\left[\frac{\partial f}{\partial \dot{q}}\right]_{ij} = tr(\frac{\partial W_i}{\partial q_i} \, {}^iW_j Q_{ii}) \tag{94}$$

The number of multiplications involved with the recursive 3 x 3 relations is $739n^2 + 62n - 54$ and the number of additions is $(1161/2)n^2 - (19/2)n - 36$. For n=6 the number of multiplications for each update of **A** and **B** is 26922 and the number of additions is 20805. This is a greater than 40% reduction in the number of operations over using 4 x 4 rotation-translation matrices.

5. CONCLUSION

This report has presented a control scheme for accurate trajectory following with robotic manipulators. The technique has been based on the use of measured joint displacements and velocities to generate corrective torques through an adaptive controller that eliminates deviations of the manipulator from the desired trajectory under feedforward control, in the least squares sense. The controller has taken into account dynamic nonlinearities (coriolis and centrifugal accelerations, pay-load change, etc.), geometric nonlinearities (nonlinear transformation matrices), physical nonlinearities (e.g., coulomb damping), dynamic coupling between joints, and real-time changes in the desired trajectory. Simulation results have been presented for a two-degree-of-freedom manipulator. These results have indicated the effectiveness and robustness of the controller. The stability issue has been addressed. Recursive relations have been developed to compute the adaptive feedback gains, thereby improving the computational efficiency of the scheme that makes the controller feasible under real-time changes in the desired trajectory. Two methods of deriving the recursive relations based on Lagrangian dynamics have been presented: (i) using 4 x 4 rotation-translation matrices, and (ii) using 3 x 3 rotation matrices. For a six degree-of-freedom manipulator, the 3x3 Lagrangian recursive relations involve 47,727 operations, which is 41% more efficient than the alternative method of using 4 x 4 rotation-translation matrices. The number of operations involved in updating the feedback gain matrix would limit the maximum update frequency to about 3 Hz when used with computers like the PDP 11 for six degree-of-freedom manipulators.

REFERENCES

[1] Asada, H., and Hanafusa, H.
An Adaptive Tracing Control of Robots and Its Application to Automatic Welding.
Proc. 1980 Joint Automatic Control Conference :FA7-D, August, 1980.

[2] DeSilva, C.W.
A Motion Control Scheme for Robotic Manipulators.
Proceedings of the Third Canadian CAD/CAM and Robotics Conference , June, 1984.

[3] DeSilva, C.W. and Aronson, M.H.
Reset and Rate Control.
Measurements and Control Journal 107:133-145, October, 1984.

[4] Dubowsky, S., and Des Forges, D.T.
The Application of Model-Referenced Adaptive Control to Robotic Manipulators.
ASME Journal of Dynamic Systems Measurement, and Control 101:193-200, September, 1979.

[5] Dubowsky, S.
On the Adaptive Control of Robotic Manipulators : The Discrete-Time Case.
Proc. 1981 Joint Automatic Control Conference 1:1A-2B, June, 1981.

[6] Hemami, H. and Camana, P.C.
Nonlinear Feedback in Simple Locomotion Systems.
IEEE Transactions on Automatic Control AC-21 No. 6:855-860, December, 1976.

[7] Hollerbach, J.M.
A Recursive Lagrangian Formulation of Manipulator Dynamics and A Comparative Study of Dynamics Formulation Complexity.
IEEE Transactions on Systems, Man, and Cybernetics SMC-10(11):730-736, Nov., 1980.

[8] Hou,F., deSilva, C., and Wright, P.
Mechanical Structural Analysis and Design Optimization of Industrial Robots.
Report No CMU-RI-TR-4, The Robotics Institute, Carnegie-Mellon University, Pittsburgh, PA , November, 1980.

[9] Klein, C.A., and Maney, J.J.
Real-Time Control of a Multiple-Element Mechanical Linkage with a Microcomputer.
IEEE Transactions on Industrial Electronics and Control Instrumentation IECI-26 No. 4:227-234, November, 1979.

[10] Luh, J.Y.S., Walker, M.W., and Paul, R.P.C.
On-Line Computational Scheme for Mechanical Manipulators.
Journal of Dynamic Systems, Measurement, and Control, Trans. ASME 102:66-76, June, 1980.

[11] Melsa, J.L., and Jones, S.K.
Computer Programs for Computational Assistance in the Study of Linear Control Theory.
McGraw-Hill, 1973.

[12] Paul, R.P.
Robot Manipulators: Mathematics, Programming, and Control.
MIT Press, Cambridge, Mass., 1981.

[13] Schultz, D.G. and Melsa, J.L.
State Functions and Linear Control Systems.
McGraw-Hill, 1967.

[14] Waters, R.C.
Mechanical arm control.
Artificial Intelligence Laboratory, M.I.T AIM 549, October, 1979.

APPENDIX A. TWO-LINK MANIPULATOR

In this appendix we formulated a dynamic model for a two-link manipulator.

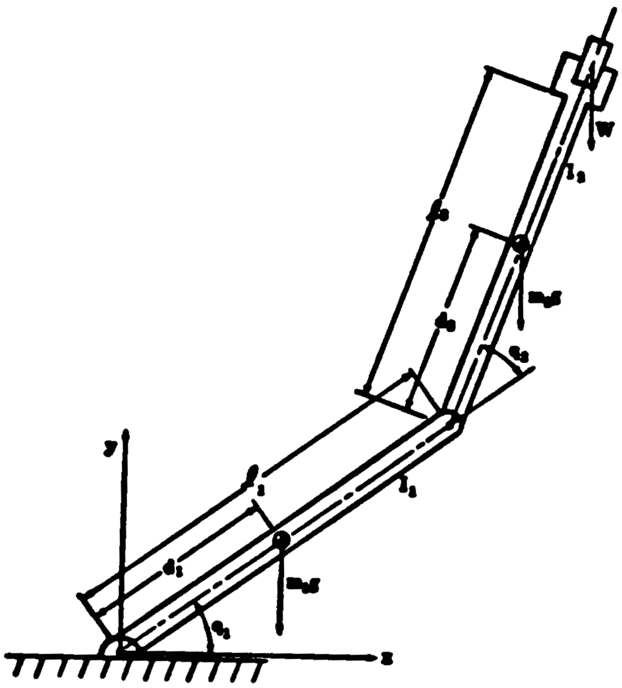


Figure A.1 Nomenclature for the two-link manipulator

$$\mathbf{q} = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} \tag{A.1}$$

A.1 Kinematics

$$\mathbf{p} = \begin{bmatrix} u_x \\ u_y \end{bmatrix} = \begin{bmatrix} l_1\cos(\theta_1) + l_2\cos(\theta_1+\theta_2) \\ l_1\sin(\theta_1) + l_2\sin(\theta_1+\theta_2) \end{bmatrix} \tag{A.2}$$

$$\begin{bmatrix} \delta u_x \\ \delta u_y \end{bmatrix} = \begin{bmatrix} -l_1\sin\theta_1 - l_2\sin(\theta_1+\theta_2) & -l_2\sin(\theta_1+\theta_2) \\ l_1\cos\theta_1 + l_2\cos(\theta_1+\theta_2) & l_2\cos(\theta_1+\theta_2) \end{bmatrix} \begin{bmatrix} \delta\theta_1 \\ \delta\theta_2 \end{bmatrix} \tag{A.3}$$

$$\begin{bmatrix} \delta u_x \\ \delta u_y \end{bmatrix} = \mathbf{J}\,\delta\mathbf{q} \tag{A.4}$$

Velocity

$$\dot{q} = J^{-1} [\, v_x \quad v_y \,]^T \tag{A.5}$$

Joint Accelerations

$$[\, a_x \; a_y \,]^T = \frac{\partial J}{\partial_t} \mathbf{q} + \mathbf{J}\,\ddot{\mathbf{q}}$$

$$\ddot{\mathbf{q}} = \mathbf{J}^{-1} \left[\, [\, a_x \; a_y]^T - \frac{\partial J}{\partial_t} \dot{\mathbf{q}} \, \right] \tag{A.6}$$

A.2 Dynamics

Define :

$$\begin{aligned}
I_1^* &= I_1 + (\, m_2 + W/g)\, l_1^2 \\
I_2^* &= m_2 d_2^2 + W/g\, l_2^2 + I_2 \\
I_3^* &= 2(\, m_2 d_2 + W/g\, l_2^2)\, l_1 \\
W_1^* &= m_1 g d_1 + m_2 g l_1 + W l_1 \\
W_2^* &= m_2 g d_2 + W\, l_2
\end{aligned}$$

Now for

$$\mathbf{M}(q,\, W)\ddot{\mathbf{q}} + \mathbf{f}(q,\, \dot{q},\, W) = \tau(t)$$

we have :

$$\begin{aligned}
M_{11} &= I_1^* + I_2^* + I_3^* \cos\theta_2 \\
M_{12} &= I_2^* + 1/2\; I_3^* \cos\theta_2 \\
M_{21} &= I_2^* + 1/2\; I_3^* \cos\theta_2 \\
M_{22} &= I_2^* \\
f_1 &= -1/2\; I_3^* \,(2\theta_1 + \theta)\; \theta\, \sin\theta_2 + W_1^* \cos\theta^* + W_2^* \cos(\theta_1 + \theta_2) \\
f_2 &= -1/2\; I_3^*\; \theta_1\; \theta_2 \sin\theta_2 + W_2^* \cos(\theta_1 + \theta_2)
\end{aligned}$$

$$\mathbf{A} = -\begin{matrix} 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & I_2^* & Minv_{12} & AM_{11} & AM_{12} & Af_{11} & Af_{12} \\ 0 & 0 & Minv_{12} & Minv_{22} & AM_{21} & AM_{12} & Af_{21} & Af_{22} \end{matrix}$$

$$\mathbf{B} = \begin{matrix} 0 & 0 \\ 0 & 0 \\ I_2^* & -(I_2^*+I_3^*/2\ \cos\theta_2) \\ -(I_2^*+I_3^*/2\ \cos\theta_2) & (I_1^*+I_2^*+I_3^*\cos\theta_2) \end{matrix}$$

where

$$Minv_{12} = -(I_2^*+I_3^*/2\ \cos\theta_2)$$
$$Minv_{22} = (I_1^*+I_2^*+I_3^*\cos\theta_2$$
$$AM_{11} = I_2^*\sin\theta_2[(W_2^*+I_3^*)\sin\theta_1+W_2^*\sin(\theta_1+\theta_2]$$
$$AM_{12} = -I_2^*\sin\theta_2[2\theta_1+\theta_2]+I_2^*\cos\theta_2[2\theta_1+\theta_2^2]+I_2^*\sin\theta_2W_1^*\sin(\theta_1+\theta_2$$
$$AM_{21} = I_2^*\sin\theta_2W_1^*\sin(\theta_1+\theta_2)$$
$$AM_{22} = -I_2^*\sin\theta_2\theta_1+I_2^*\cos\theta_1\theta_2^2$$
$$Af_{11} = -2I_2^*\sin\theta_{1q}$$
$$Af_{12} = -2I_2^*\sin\theta_2(\theta_1+\theta_2)$$
$$Af_{21} = 2I_2^*\theta_1$$
$$Af_{22} = 0$$

APPENDIX B. RECURSIVE CONTROL PARAMETERS WITH 3 X 3 MATRICES

In this appendix the formulation for the three matrices, $\mathbf{M}^{-1}$, $\frac{\partial \mathbf{M}}{\partial \mathbf{q}}\ddot{\mathbf{q}} + \frac{\partial \mathbf{f}}{\partial \mathbf{q}}$, and $\frac{\partial \mathbf{f}}{\partial \dot{\mathbf{q}}}$, is developed using 3 x 3 rotation matrices.

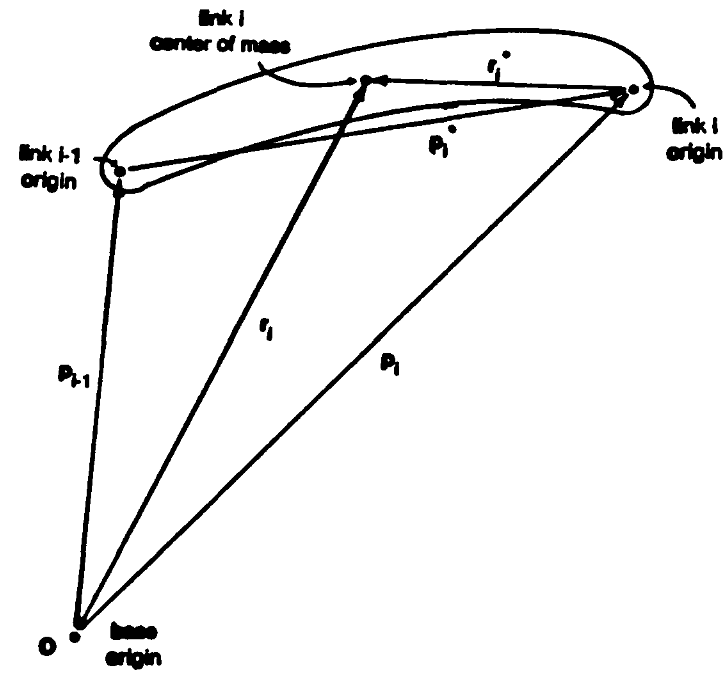


Figure B.1 3x3 Vector definitions

p_i: vector from base coordinate origin to the joint i coordinate origin

p_i^*: vector from the origin i-1 to coordinate origin i.

r_i: vector from the base coordinate origin to the link i center of mass

r_i^*: vector from coordinate origin i to the link i center of mass

n_i: r_i^* / m

${}^jW_k$: defined as before except it is composed of 3 x 3 rotation matrices.

Then the generalized force as derived by Hollerbach [7] is

$$f_i = \sum_{r=1}^{n} \left[tr\left(m_r \frac{\partial p_r}{\partial q_i} \ddot{p}_r^T + \frac{\partial p_r}{\partial q_i} {}^r n_r^T \ddot{W}_r^T + \frac{\partial W_r}{\partial q_i} {}^r n_p^T \ddot{p}_p^T + \frac{\partial W_r}{\partial q_i} {}^r J_p \ddot{W}_r \right) - m_p g^T \frac{\partial W_p}{\partial q_i} {}^p r_p \right] \quad (B.1)$$

(a) $\left[\frac{\partial \mathbf{M}}{\partial \mathbf{q}} \ddot{\mathbf{q}} + \frac{\partial \mathbf{f}}{\partial \mathbf{q}}\right]$ *term:*

$$\left[\frac{\partial M}{\partial q}\ddot{\mathbf{q}}+\frac{\partial f}{\partial q}\right]_{ij} = \sum_{p=\max i,j}^{n}\Big[tr\Big(m_p\frac{\partial^2 p_p}{\partial q_i \partial q_j}\ddot{p}_p^{\,T} + m_p\frac{\partial p_p \partial \ddot{p}_p^{\,T}}{\partial q_i\,\partial q_j} + \frac{\partial^2 p_p}{\partial q_i \partial q_j}\,{}^p n_p^{\,T}\ddot{W}_p^{\,T} + \frac{\partial p_p}{\partial q_i}\,{}^p n_p^{\,T}\frac{\partial \ddot{W}_p^{\,T}}{\partial q_j} + \frac{\partial^2 W_p}{\partial q_i \partial q_j}\,{}^p n_p^{\,T}\ddot{p}_p^{\,T} + \frac{\partial W_p}{\partial q_i}\,{}^p n_p^{\,T}\frac{\partial \ddot{p}_p^{\,T}}{\partial q_j} + \frac{\partial^2 W_p}{\partial q_i \partial q_j} J_p \ddot{W}_p + \frac{\partial W_p}{\partial q_i} J_p \frac{\partial \ddot{W}_p}{\partial q_j}\Big) - m_p g^T\frac{\partial^2 W_p}{\partial q_i \partial q_j}\,{}^p r_p\Big] \tag{B.2}$$

Now for the case where $i \le j$

$$p_p = p_i + W_i\,{}^i p_i$$

$$\frac{\partial p_p}{\partial q_i} = \frac{\partial W_i}{\partial q_i}\,{}^i p_p \tag{B.3}$$

$$\frac{\partial W_p}{\partial q_i} = \frac{\partial W_i}{\partial q_i}\,{}^i W_p \tag{B.4}$$

For $i \ge j$

$$\frac{\partial^2 p_p}{\partial q_i \partial q_j} = \frac{\partial^2 W_i}{\partial q_i \partial q_j}\,{}^i p_p \tag{B.5}$$

$$\frac{\partial^2 W_p}{\partial q_i \partial q_j} = \frac{\partial^2 W_i}{\partial q_i \partial q_j}\,{}^i W_p \tag{B.6}$$

$$\left[\frac{\partial M}{\partial q}\ddot{\mathbf{q}}+\frac{\partial f}{\partial q}\right]_{ij} = tr\Big[\frac{\partial^2 W_i}{\partial q_i \partial q_j}\sum_{p=i}^{n}\big(m_p\,{}^i p_p\ddot{p}_p^{\,T} + {}^i p_p\,{}^p n_p^{\,T}\ddot{W}_p + {}^i W_p\,{}^p n_p^{\,T}\ddot{p}_p^{\,T} + {}^i W_p J\ddot{W}_p\big) + \frac{\partial W_i}{\partial q_i}\sum_{p=i}^{n}\Big(m_p\,{}^i p_p\frac{\partial \ddot{p}_p^{\,T}}{\partial q_j} + {}^i p_p\,{}^p n_p^{\,T}\frac{\partial \ddot{W}_p}{\partial q_j} + {}^i W_p\,{}^p n_p^{\,T}\frac{\partial \ddot{p}_p^{\,T}}{\partial q_j} + {}^i W_p J\frac{\partial \ddot{W}_p}{\partial q_j}\Big)\Big] - g^T\frac{\partial^2 W_i}{\partial q_i \partial q_j}\sum_{p=i}^{n} m_p\,{}^i W_p\,{}^p r_p \tag{B.7}$$

Let

$$D_i = \sum_{p=i}^{n} (m_p {}^ip_p \ddot{p}_p^T + {}^ip_p {}^pn_p^T \ddot{W}_p + {}^iW_p {}^pn_p^T \ddot{p}_p^T + {}^iW_p J \ddot{W}_p)$$

$$D_i = 0 + 0 + {}^in_i^T p_i^T + J_i W_i^T +$$

$$\sum_{p=i+1}^{n} [(A_{i+1} {}^{i+1}p_p + {}^ip_{i+1})(m_p \ddot{p}_p^T + {}^pn_p^T \ddot{W}_p^T) + (A_{i+1} {}^{i+1}W_p)({}^pn_p^T \ddot{p}_p^T + J_p \ddot{W}_p^T)]$$

$$D_i = A_{i+1} D_{i+1} + {}^ip_{i+1} e_{i+1} + {}^in_i^T \ddot{p}_i^T + J_i \ddot{W}_i^T \qquad (B.8)$$

where

$$e_i = \sum_{p=i}^{n} (m_p \ddot{p}_p^T + {}^pn_p^T \ddot{W}_p^T)$$

$$e_i = e_{i+1} + m_i \ddot{p}_i^T + {}^in_i^T \ddot{W}_i^T \qquad (B.9)$$

Similarly,

$$N_i = \sum_{p=i}^{n} (m_p {}^ip_p \frac{\partial \ddot{p}_p^T}{\partial q_j} + {}^ip_p {}^pn_p^T \frac{\partial \ddot{W}_p}{\partial q_j} + {}^iW_p {}^pn_p^T \frac{\partial \ddot{p}_p^T}{\partial q_j} + {}^iW_p J \frac{\partial \ddot{W}_p}{\partial q_j})$$

$$N_i = A_{i+1} N_{i+1} + {}^ip_{i+1} l_{i+1} + {}^in_i^T \frac{\partial \ddot{p}_i^T}{\partial q_j} + J_i \frac{\partial \ddot{W}_i^T}{\partial q_j} \qquad (B.10)$$

where

$$l_i = \sum_{p=i}^{n} (m_p \frac{\partial \ddot{p}_p^T}{\partial q_j} + {}^pn_p^T \frac{\partial \ddot{W}_p^T}{\partial q_j})$$

$$l_i = l_{i+1} + m_i \frac{\partial \ddot{p}_i^T}{\partial q_j} + {}^in_i^T \frac{\partial \ddot{W}_i^T}{\partial q_j} \qquad (B.11)$$

The recurrence relation c_i for the gravity term is the same as equation (69).

For $i \geq j$

$$\left[\frac{\partial M}{\partial q} \ddot{\mathbf{q}} + \frac{\partial f}{\partial q}\right]_{ij} = \left[tr(\frac{\partial^2 W_p}{\partial q_i \partial q_j} D_p) + tr(\frac{\partial W_p}{\partial q_i} N_p) - g^T \frac{\partial^2 W_p}{\partial q_i \partial q_j} c_p\right] \qquad (B.12)$$

(*b*) $\left[\frac{\partial \mathbf{f}}{\partial \dot{\mathbf{q}}}\right]$ *term*:

$$\left[\frac{\partial f}{\partial \dot{q}}\right]_{ij} = \frac{\partial}{\partial \dot{q}_j}\tau_i \qquad i = 1,\dots,n, \quad j=1,\dots n$$

$$= \sum_{p=\max i,j}^{n} \left(m_p \frac{\partial p_p \partial \ddot{p}_p^T}{\partial q_i\, \partial \dot{q}_j} + \frac{\partial p_p}{\partial q_i}{}^p n_p^T \frac{\partial \ddot{W}_p}{\partial \dot{q}_j} + \frac{\partial W_p}{\partial q_i}{}^p n_p^T \frac{\partial \ddot{p}_p^T}{\partial \dot{q}_j} + \frac{\partial W_p}{\partial q_i}\mathbf{J}_i \frac{\partial \ddot{W}_p}{\partial \dot{q}_j}\right) \tag{B.13}$$

Now,

$$\frac{\partial \ddot{p}_p}{\partial \dot{q}_i} = \frac{\partial \dot{p}_p}{\partial q_i} \tag{B.14}$$

$$\frac{\partial \ddot{W}_p}{\partial \dot{q}_i} = \frac{\partial \dot{W}_p}{\partial q_i} \tag{B.15}$$

For $i \geq j$

$$\frac{\partial p_p}{\partial q_i} = \frac{\partial W_i}{\partial q_i}{}^i p_p \tag{B.16}$$

$$\frac{\partial W_p}{\partial q_i} = \frac{\partial W_i}{\partial q_i}{}^i W_p \tag{B.17}$$

Therefore

$$\left[\frac{\partial f}{\partial \dot{q}}\right]_{ij} = tr\left(\frac{\partial W_i}{\partial q_i}\sum_{p=i}^{n}\left(m_p{}^i p_p \frac{\partial \dot{p}_p^T}{\partial q_j} + {}^i p_p{}^p n_p^T \frac{\partial \dot{W}_p}{\partial q_j} + {}^i W_p{}^p n_p^T \frac{\partial \dot{p}_p^T}{\partial q_j} + {}^i W_p \mathbf{J} \frac{\partial \dot{W}_p}{\partial q_j}\right)\right. \tag{B.18}$$

Let

$$Q_i = \sum_{p=i}^{n}\left(m_p{}^i p_p \frac{\partial \dot{p}_p^T}{\partial q_j} + {}^i p_p{}^p n_p^T \frac{\partial \dot{W}_p}{\partial q_j} + {}^i W_p{}^p n_p^T \frac{\partial \dot{p}_p^T}{\partial q_j} + {}^i W_p \mathbf{J} \frac{\partial \dot{W}_p}{\partial q_j}\right)$$

$$\Rightarrow Q_i = A_{i+1}Q_{i+1} + {}^i p_{i+1} b_{i+1} + {}^i n_i^T \frac{\partial \dot{p}_i^T}{\partial q_j} + \mathbf{J}_i \frac{\partial \dot{W}_i^T}{\partial q_j} \tag{B.19}$$

where

$$b_i = \sum_{p=i}^{n} (m_p \frac{\partial \dot{p}_p^T}{\partial q_j} + {}^p n_p^T \frac{\partial \dot{W}_p^T}{\partial q_j})$$

$$b_i = b_{i+1} + m_i \frac{\partial \dot{p}_i^T}{\partial q_j} + {}^i n_i^T \frac{\partial \dot{W}_i^T}{\partial q_j} \qquad (B.20)$$

for i ≥ j

$$\left[\frac{\partial f}{\partial \dot{q}}\right]_{ij} = tr(\frac{\partial W_i}{\partial q_i} Q_i) \qquad (B.21)$$

Similarly for j > i

$$\left[\frac{\partial f}{\partial \dot{q}}\right]_{ij} = tr(\frac{\partial W_i}{\partial q_i} {}^i W_j Q_j) \qquad (B.22)$$

$$Q_j = A_{j+1} Q_{j+1} + {}^j p_{j+1} b_{j+1} + {}^j n_j^T \frac{\partial p_i^T}{\partial q_j} + \mathbf{J}_j \frac{\partial W_j^T}{\partial q_j} \qquad (B.23)$$

By a similar procedure we obtain

(c) $\mathbf{M}_{ij}$ term:

For i ≥ j

$$M_{ij} = tr(\frac{\partial W_i}{\partial q_i} P_i) \qquad (B.24)$$

$$P_i = A_{i+1} P_{i+1} + {}^i p_{i+1} k_{i+1} + {}^i n_i^T \frac{\partial p_i^T}{\partial q_j} + \mathbf{J}_i \frac{\partial W_i^T}{\partial q_j} \qquad (B.25)$$

for j ≥ i

$$M_{ij} = tr(\frac{\partial W_i}{\partial q_i} {}^i W_j P_j) \qquad (B.26)$$

$$P_j = A_{j+1} P_{j+1} + {}^j p_{j+1} k_{j+1} + {}^j n_j^T \frac{\partial p_i^T}{\partial q_j} + \mathbf{J}_j \frac{\partial W_j^T}{\partial q_j} \qquad (B.27)$$

where

$$k_i = k_{i+1} + m_i \frac{\partial p_i^T}{\partial q_j} + {}^i n_i^T \frac{\partial W_i^T}{\partial q_j} \qquad (B.28)$$

The last new terms that need to be calculated are the p terms.

Since

$$p_{i-1} = p_i + W_i \, {}^{i}p_i^*$$

Then

$$p_i = p_{i-1} - W_i \, {}^{i}p_i^*$$

and for $j \le i$

$$\frac{\partial p_i}{\partial q_j} = \frac{\partial p_{i-1}}{\partial q_j} - \frac{\partial W_i}{\partial q_j} \, {}^{i}p_i^*$$

$$\frac{\partial p_i}{\partial q_j} = \frac{\partial p_{i-1}}{\partial q_j} - \frac{\partial W_i}{\partial q_j} \, {}^{i}p_i^* \qquad (B.29)$$

$$\frac{\partial p_i}{\partial q_j} = \frac{\partial p_{i-1}}{\partial q_j} - \frac{\partial W_i}{\partial q_j} \, {}^{i}p_i^*$$