Accurate Trajectory Control of Robotic Manipulators

J. C Van Winssen and C W. deSilva

CMU-RI-TR-85-15 (3)

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 robot nanipulators. The method uses feedforward control using model-based torques for fast operatic nd gross compensation, and adaptive feedback control for correcting deviations from the desire rajectory under feedforward control. The adaptive controller eliminates trajectory-following erro n the least squares sense. The control scheme takes into account dynamic nonlinearities (e.g oriolis and centrifugal accelerations and payload changes), geometric nonlinearities (e.g., nonline oordinate-transformation matrices) and physical nonlinearities (e.g., nonlinear damping) as well ynamic coupling in manipulators. Computer simulations are presented to indicate the effectivene nd robustness of the control scheme. When the desired trajectory is completely known before t ontrol scheme is implemented, then off-line computations can be used to generate the adapti eedback gains and the computational efficiency will not be a major limiting factor with th ontrol scheme. If real-time changes in the desired trajectory have to be accommodated, t omputational efficiency has to be improved using recursive relations to compute the adapti ains. The necessary recursive relations are derived and presented in this report.

.. INTRODUCTION

Many robot applications today and in the future will require accurate tracking of a prespecifie ontinuous path. Common examples of these tracking applications include seam tracking, ai 'elding, cutting (laser and water jet), spray painting, contours inspection, co-ordinated par •ansfer and assembly operations. These tracking paths are usually specified with respect to tl nd effector of the robotic manipulator and can specify trajectories with respect to time as we s position. The problem with achieving this objective of temporal path following is that strot onlinearities in the dynamics and geometry, unknown parameters, modeling errors, measuremei rrors, unplanned changes in operating conditions, and other disturbances are present in tl lanipulator 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.

dany control systems, which meet these requirements with different degrees of success, have bee roposed and some have been implemented. The control scheme developed in this report *cz* ccurately follow a prespecified trajectory while rejecting many classes of disturbances by using sedback control scheme that minimizes position and velocity deviation in the least squares sen: 'hile allowing for the changing of the feedback control parameters to account for unkno* hanges in payload or desired trajectory. A two-link manipulator simulation shows tl ffectiveness of this control scheme for trajectory following. However, the computational effo squired with this control scheme is high enough to limit the maximum sampling frequen< llowed for manipulator control in real time. Therefore the maximum trajectory-followii ccuracy that this control scheme can achieve is also limited by the computational effort, if tl esired trajectory is not known a priori, and is changing in real time.

Linear servo control is the most common type of control in commercial use today [3]. Thj :ontrol method involves having a separate feedback loop closed over each manipulator joint thz eedbacks the position (and sometimes velocity) of that joint. This control method has seven •roblems which limit its commercial usefulness. Since each control loop is closed independentl iver each manipulator joint, it has poor compensation for the dynamic coupling (i.e., particular] :oriolis forces and coordinate coupling) between joints because the effect of the motion of or oint on another is viewed as a disturbance which the feedback controller of the second joii aust compensate for. At low speeds, these "disturbance" forces are small and can be easil .ompensated for, but at high speeds, these forces are major components in the dynamics of tt manipulator, and the controller will fail to totally reject these "disturbances" and the end effectc /ill no longer be following the correct path [8]. Another factor is that the servo parametei isually are tuned for one set of operating conditions and can not be changed to meet changin onditions like payload variations during robot operation. Furthermore, classical servo contrc ssumes linear plants, which is not close to reality in the case of robotic manipulator

Other control schemes have been proposed that eliminate some of these problems but none ha\ >een commercially implemented. These methods include Model-Referenced Adaptive Contro Hiding Mode Control (a method of designing switching feedback regulators based on minimui ime, bang-bang control), optimal control, nonlinear feedback control and feedforward contro application of these control techniques, particularly for real-time control, is hindered by tl omplexity of the associated control algorithms, which increases the computation-cycle time an lecreases the control bandwidth.

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

Controller parameters are adjusted according to a differential law that uses the error signal (tl iifference between response of the reference model and the actual robot) as the input. There exi everal 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.

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 t he presence of adaptive feedback loops, there is no guarantee that the required accuracy btained in a given situation of trajectory following.

A control technique that strives to obtain linear behavior from a nonlinear manipulator is know s sliding mode control [9]. In the generalized case of this method (only the two dimension ase is presented by Klein and & Maney [9]), the state space is partitioned into several regio hat are bounded by a space trajectory conformal to the desired linear behavior. The objective he control would be to drive the manipulator along the desired trajectory. This is accomplishe y assigning a different control law for each region in the partitioned state space. If t anipulator deviates from the desired trajectory and enters a particular region of the state spa he corresponding control law is switched on. This will drive the manipulator back into t esired trajectory. If it overshoots, however, the control law of the new region which t anipulator entered will be automatically switched on to drive the the manipulator into t esired trajectory. If the alternative control laws that are assigned to the various regions can witched on at infinite frequency, which is of course not realistic, it is possible in theory, btain ideal behavior. In practice, however, the response will chatter about the desired trajector 'he amplitude of chatter will depend on the manipulator dynamics as well as control gains use n addition the switching frequency will depend on the deadband of control. These shortcomin f sliding mode control can be aggravated by the fact that the control laws are selected in eurístic manner, without even employing a model to represent the actual dynamics of t anipulator. At its best, sliding mode control usually brings about time delays (non-synchrono esponse) in addition to chatter. This technique too, has not been implemented in commerci obots.

In optimal control, the feedback control law is designed by optimizing a suitable performan ndex using a dynamic model for the manipulator. Control laws obtained in this manner can ighly complex except in a very few special cases. A nonlinear control approach that has bee roposed for robotic manipulator control is aimed at obtaining a desirable linear behavior fro he manipulator by employing a highly nonlinear feedback law [6, 1]. Unlike the mode eferenced adaptive control method, this control law is derived from an accurate nonlinear mod or the robot. The main disadvantage of the method, as has been warned by Asada & Hanafus [1] is the feedback law that is so complex, it is virtually impossible to compute the feedba arameters in real time for practical robots. Furthermore, performance of this nonlinear contr ystem is known to be quite sensitive to fidelity of the robot model that is employed.

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

LI Feedforward Control

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

$$M(q,W)\ddot{q} + f(q,\dot{q},W> = r(t) \qquad (1$$

where

W : payload

q : vector of generalized joint positions

M(q,W) : inertia matrix (n x n)

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

r(t) : input torques or forces at the
manipulator joints

In practical manipulators, input signals (e.g., field voltages, servovalve commands) produt aotor torques at the joints, with some dynamic delay. Motor torques are converted into tl arques that are actually applied to the links of the manipulator, with additional dynamic dela Manipulator displacements are a result of these joint torques. It is therefore clear that, by eith< leasuring or computing joint torques it is possible to eliminate part of the delay in lanipulator control system. Consequently, feedforward control has the advantage of speeding i tie manipulator response. Furthermore, torque disturbances can be calculated or measured, the an be completely rejected using feedforward control. A main disadvantage of feedforwai ontrol, in the present context, is that due to model errors and unknown disturbances, tl alculated torque is not the ideal torque and as a result errors can grow in an unstable mann< nless some form of feedback control is used.

Since in inverse dynamics a mathematical model of the manipulator is used to calculate the joii Drques required, when these torques are applied to the actual manipulator it might not follow tl esired trajectory accurately. This would be due to the cumulative effects of modelii riction. Therefore, for accurate tracking using feedforward control a precise dynamic model k o be employed and the manipulator must be made very rigid with strong structural links an decision gear trains and actuators. Another problem with this method is that the computations ffort required to accurately compute the necessary torques in a real-time situation can becoa 'ery significant if the desired trajectory is not known a priori and may not allow a sufficient] igh sampling rate for good control bandwidth.

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

1.2 Background Theory

In most instances, feedforward control needs a feedback controller to correct for unaccountc listurbances in the system. Since linear-servo control offers only$_f$ a limited ability to compensa or nonlinearities, model errors, measurement errors and disturbances a more adaptive feedbac ontroller was developed by R.P. Paul [2]. This controller is based on a nonlinear couple iynamic model of the manipulator, and therefore takes into account effects that linear contn isually neglects. It also allows for updating the control parameters to take care of unknow xternal disturbances and payload variations. The basic block diagram for the control system een in figure 1.

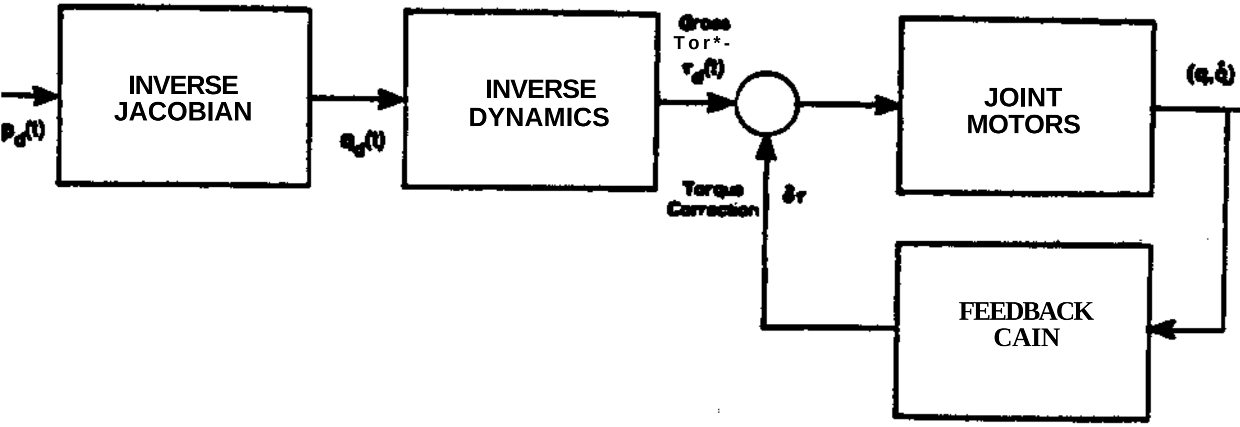


Figure 1. Basic control diagram for the manipulator

we can linearize me nonlinear set or dinerentiai equations u; wiin respect to sma
Perturbations, 5q, from the desired trajectory, $q_d(t)$, caused by small torque disturbances, 5r(t)

$$M(q_d,W)<3c+q_{d\wedge d}\frac{..\,..3M}{dq}(q\,,W)5q+\frac{3f}{dq}_{/v}<q_d,\dot{q}_d,W)5q+\frac{3f}{dq}_{*}(q_d,\dot{q}_d,W)5q \doteq \delta\tau(t) \qquad (:$$

where

$$\left[\ddot{q}_d\frac{\partial M}{\partial q}\right]_{ij}=\sum_{k=1}^{n}\frac{3M_{ik}}{\partial q_j}\ddot{q}_k$$

This equation can be rearranged in vector-matrix form

$$\begin{bmatrix} I & 0 \\ 0 & M_d \end{bmatrix}\begin{bmatrix} <3\dot{q} \\ *\ddot{q} \end{bmatrix} + \begin{bmatrix} * & 0 & & -I \\ 4 & \frac{Vi}{Oq}\Big| + \frac{f}{dq} & & \Big|\frac{f}{dq}_d \end{bmatrix}\begin{bmatrix} <5q \\ \delta\dot{q} \end{bmatrix} = \begin{bmatrix} 0 \\ I \end{bmatrix}\delta\tau(t) \qquad (.$$

where, $[\;]_{\dot a}$, denotes terms evaluated in terms of the desired trajectory, $q_{\dot a}(t)$.

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

$$x \gg [\;<5q,\;3\dot{q}\;]^T, \qquad u \ll 5r$$

thus,

$$\dot{x} \gg Ax(O + Butt) \qquad (\text{á}$$

where, the system matrix

$$A(q,\overset{\ll\,\bullet\bullet}{q,q},W)_d = -\begin{bmatrix} 1 & 0 & 0 & -I \\ 0 & M^{-1}_d & \ddot{q}\frac{\partial M}{\partial q}+\frac{\partial f}{\partial q} & \frac{\partial f}{\partial \dot{q}}_d \end{bmatrix} \qquad \backslash.$$

and the input gain*:* matrix is

$$B(q_d.\;W) = \begin{bmatrix} 0 \\ M'^1_d \end{bmatrix} \qquad ($$

Since what is developed would be implemented as a digital control scheme, we need the discre
ƒbrm of the state space representation

for $\frac{dx}{\quad} = Ax +$ *Suit)*

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

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

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

x(*+l) = <£>x(*) + *Fu(k)* *k* » 0,1,2,3,...

in which

4> = $_e{}^{AT}$ s $_s$tate transition matrix

F ^{s} $\int$ I J *c*^{A}*^dj3B* = input gain matrix

T = data sampling period

.2.2 Minimization

Since the state vector x represents the deviation in position and velocity, from the desire •ajectory, then the objective of the minimization is to drive x to zero as fast as possible. Th ill be accomplished in the least squares sense by using the following objective index

Least Squares Minimization Performance Index :

$$J \bullet 21_{k-1}^{N} [4>x(*) + ru(*)]^T Q [<£>x(A-) + ru<\#)] \qquad (i$$

where Q is a diagonal weighting matrix. Q is used to weight the relative importance of eac aint position or velocity. This allows the motions of critical joints to be more heavily weight* aan the motions of other joints.

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

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

$u(k)$ = -Kx<*) (9

where K = (T^T Q IV^1 r^T Q ∢ *x(k)* (10

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

rank(T^T Q D = n (11

In particular, if

Q is positive definite, we must have

rank(D = n (12

where, n = degrees of freedom of manipulator

.3 Control Strategy

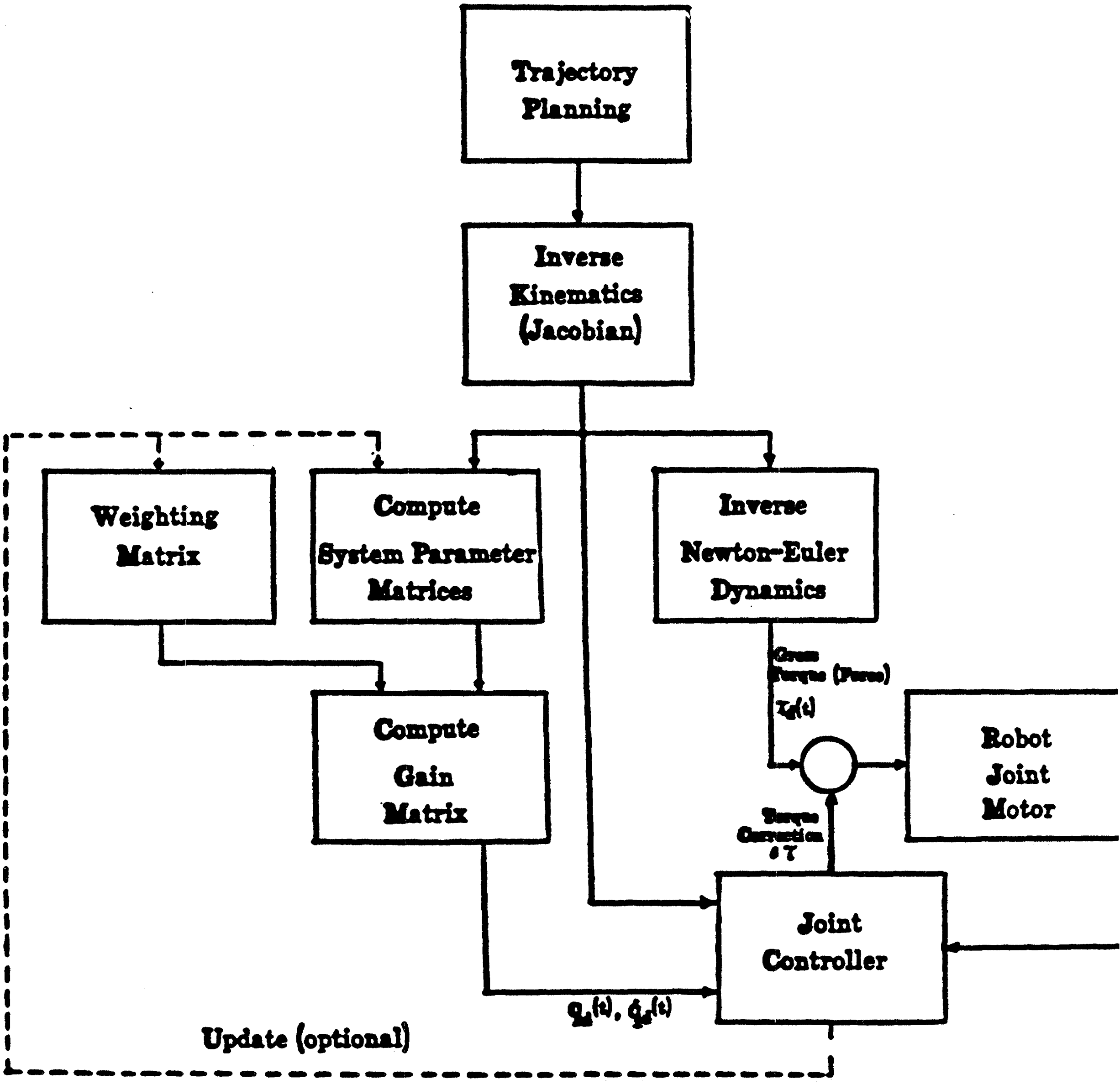
The complete control strategy for the manipulator is shown in figure 2. First the desired end ffector trajectory of the manipulator is ^generated. Then, using some inverse kinematics schem< ach incremental displacement, velocity and acceleration of the end-effector is translated into tt orresponding motions of the n joints. With the inverse dynamics of the manipulator, the desire ross torques for each joint can be calculated. These torques are applied to the actu lanipulator in a feedforward manner. The actual joint positions and velocities are then measure nee every period, T_s using resolvers or encoders. The difference between the actual and tt esired joint motions is then multiplied by the optimal feedback gain matrix, K, to produce tl ector of torque corrections that need to be added to the gross torque vector for proper contro k suitable criterion is needed to decide when to update the feedback gain matrix, K. In tl resent work the following criterion is used:

- Initially specify the weighting matrix Q and calculate, 4\ and ,I\
- Compute the initial feedback gain matrix, K using equation (10).
- Update the feedback gain matrix, K, according to the criterion

1. If ||x|| < e_Q Skip torque error feedback
2. If ||x|| > $€_x$ Update 4>,r,Q, and K
3. If ||x|| > 6_2 Excessive Error, terminate operation

Note that * < *e* < € . The error norm is defined as i I x 1 I = $\sum$ *y* ^{n} *a* *\x*.\
0 1 2 1 1 1 1 *Jmmmtl*^{u}\ \ * *t* *

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


Trajectory Planning
Inverse Kinematics (Jacobian)
Weighting Matrix
Compute System Parameter Matrices
Inverse Newton-Euler Dynamics
Gross Torque (Force)
$\tau_d(t)$
Compute Gain Matrix
Robot Joint Motor
Torque Correction
$\delta\tau$
Joint Controller
$q_d(t)$, $\dot{q}_d(t)$
Update (optional)

2.3.1 Stability

If the manipulator model is significantly different from the actual robot, then the feedback could cause instability in our control system. Stability is guaranteed if the closed-loop 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

*. SIMULATION RESULTS

The effectiveness of the control strategy presented in this report, *is* examined using a twc
legree-of-freedom manipulator. The manipulator equations are given in Appendix A. Two types c
listurbances 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 a
hree cases the feedforward control alone produces an unstable trajectory following. By addir
lie adaptive optimal feedback controller the actual trajectory was brought very close (8$^{(}$
laximum position error) to the desired trajectory.

It appears that our control scheme satisfies three of the four design goals for the controlle
ccurately tracks the end effector, rejects a wide class of disturbances, and is very reliable. Tl
ist goal is minimal complexity, or making the scheme computationally fast enough to allow *z*
dequate sampling rate for on-line trajectory generation and control.

.1 Two-Link Manipulator Results

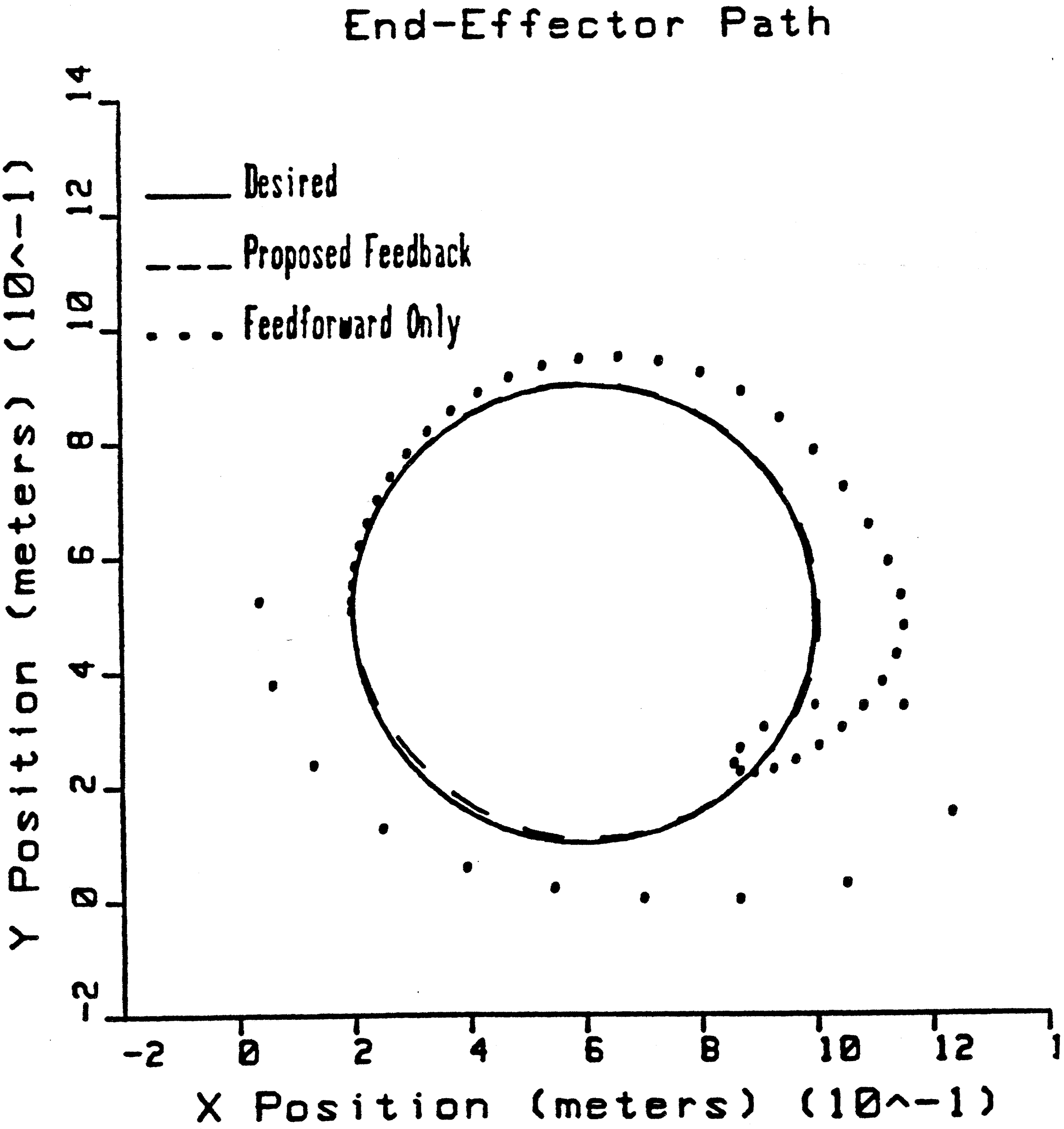


Figure 3.1 End-effector path with input disturbances

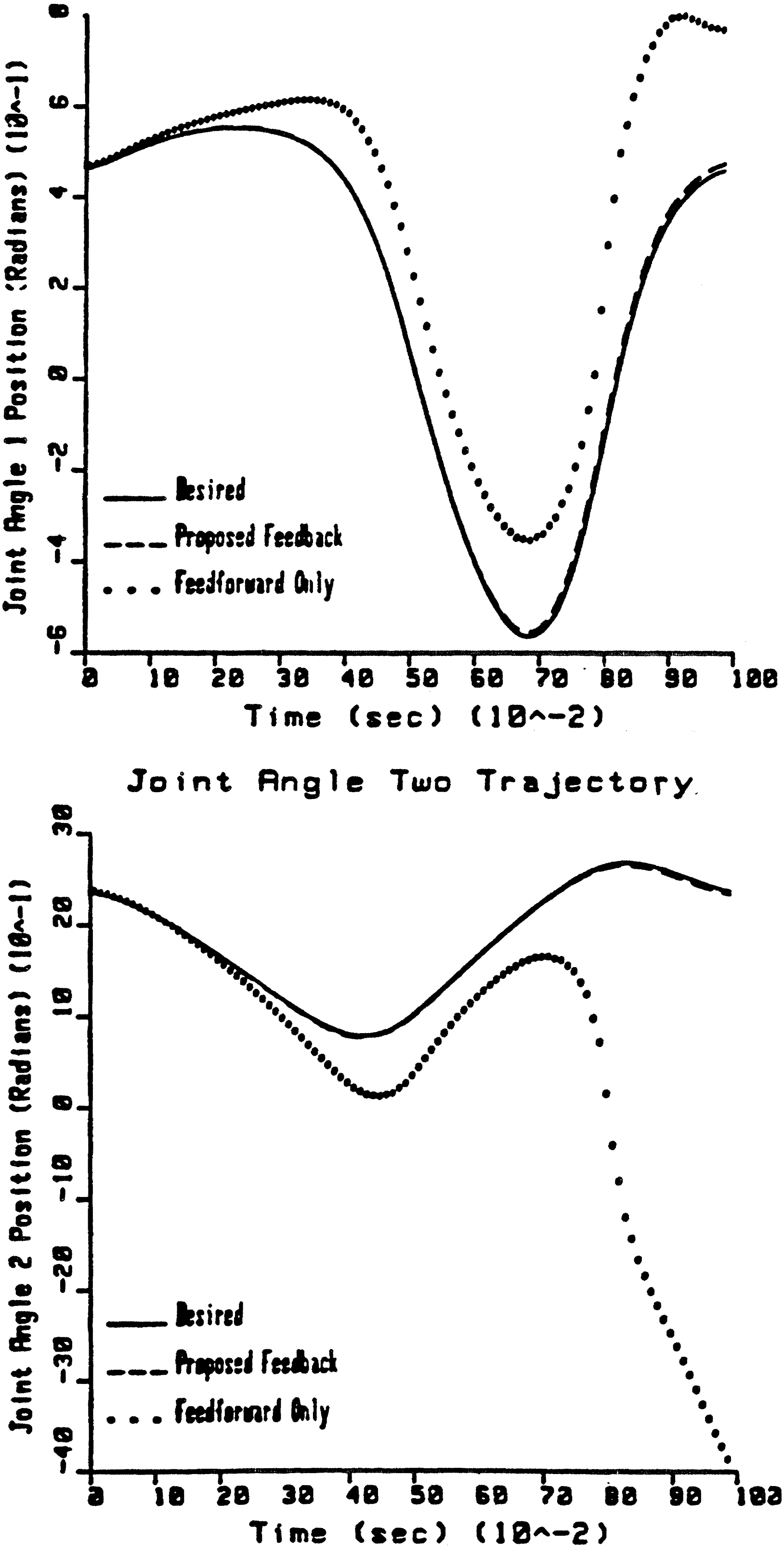


Figure 3.2 Joint trajectories with input disturbances

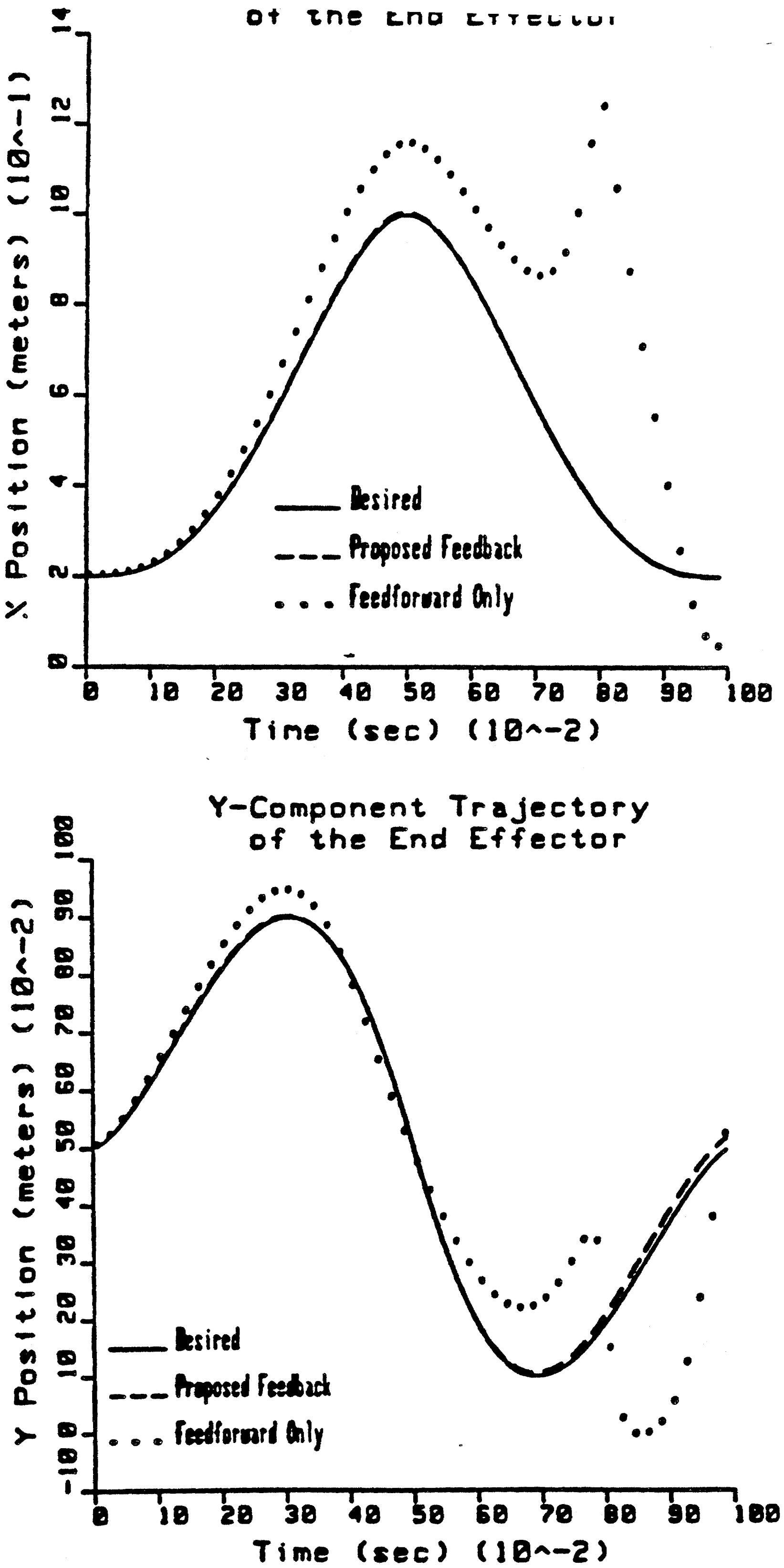
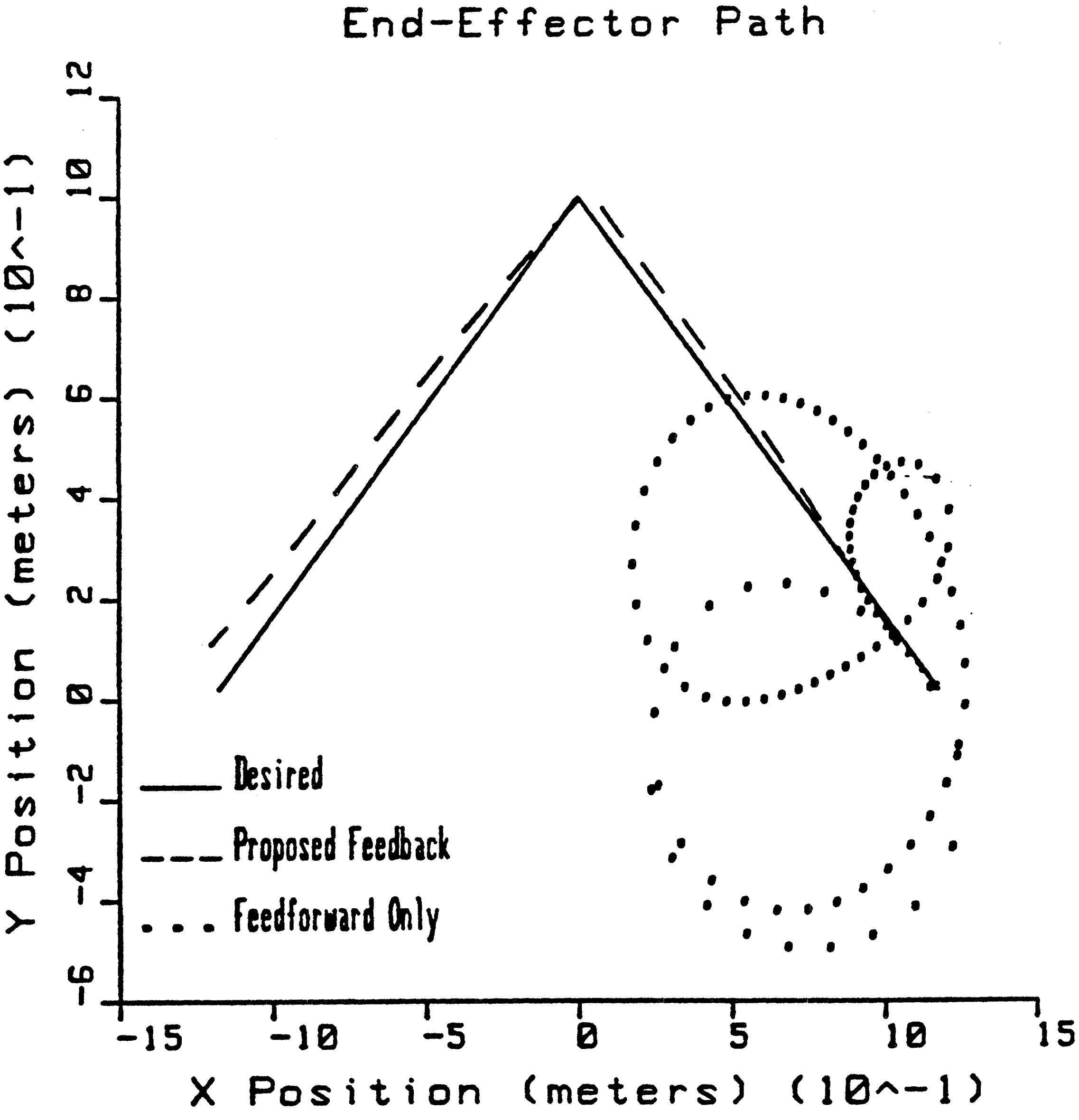


Figure 3.3 X and Y position trajectories of the end-effector with input disturbances


End-Effector Path
Y Position (meters) (10^-1)
X Position (meters) (10^-1)
12
10
8
6
4
2
0
-2
-4
-6
-15
-10
-5
0
5
10
15
Desired
Proposed Feedback
Feedforward Only

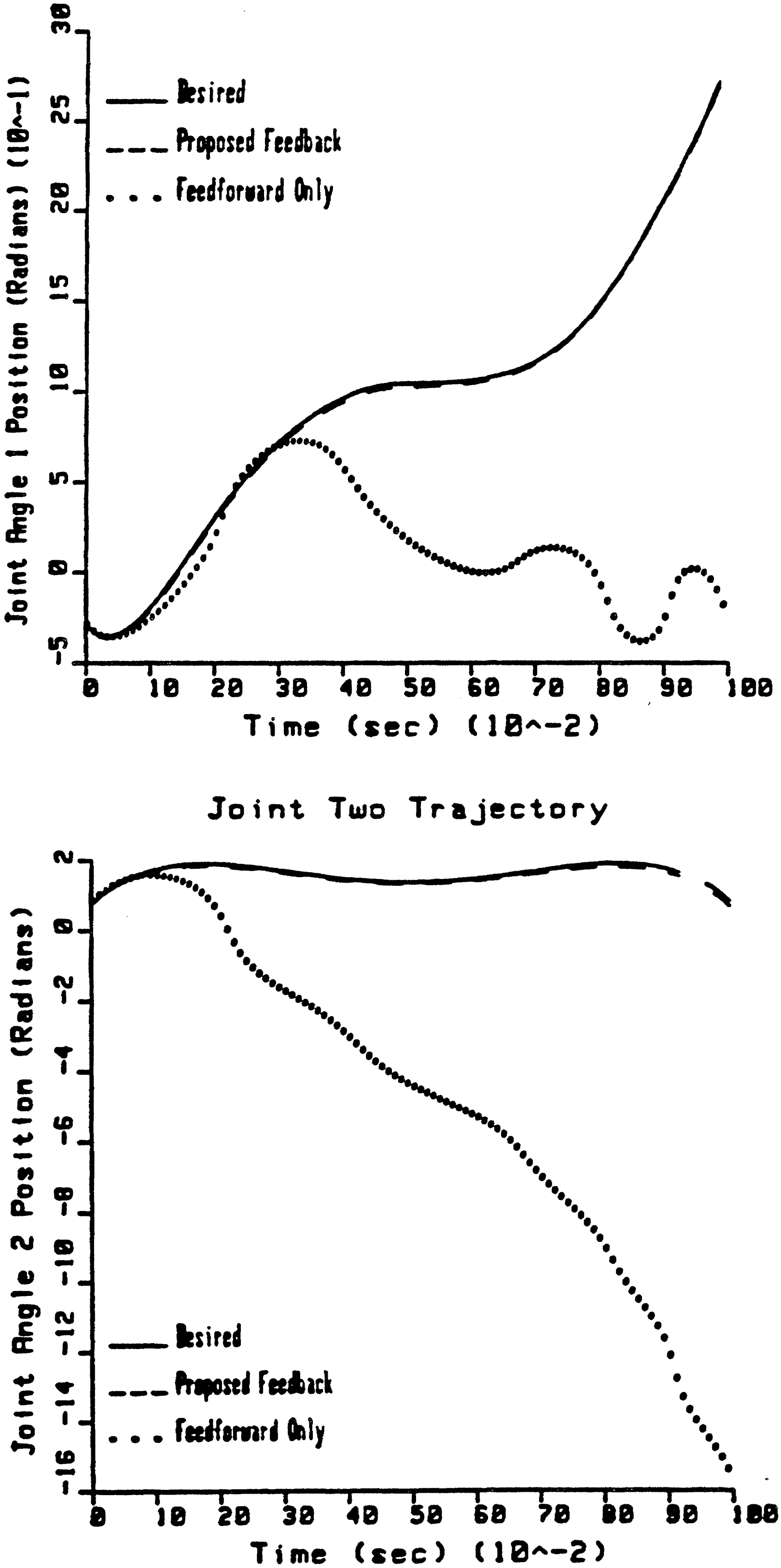
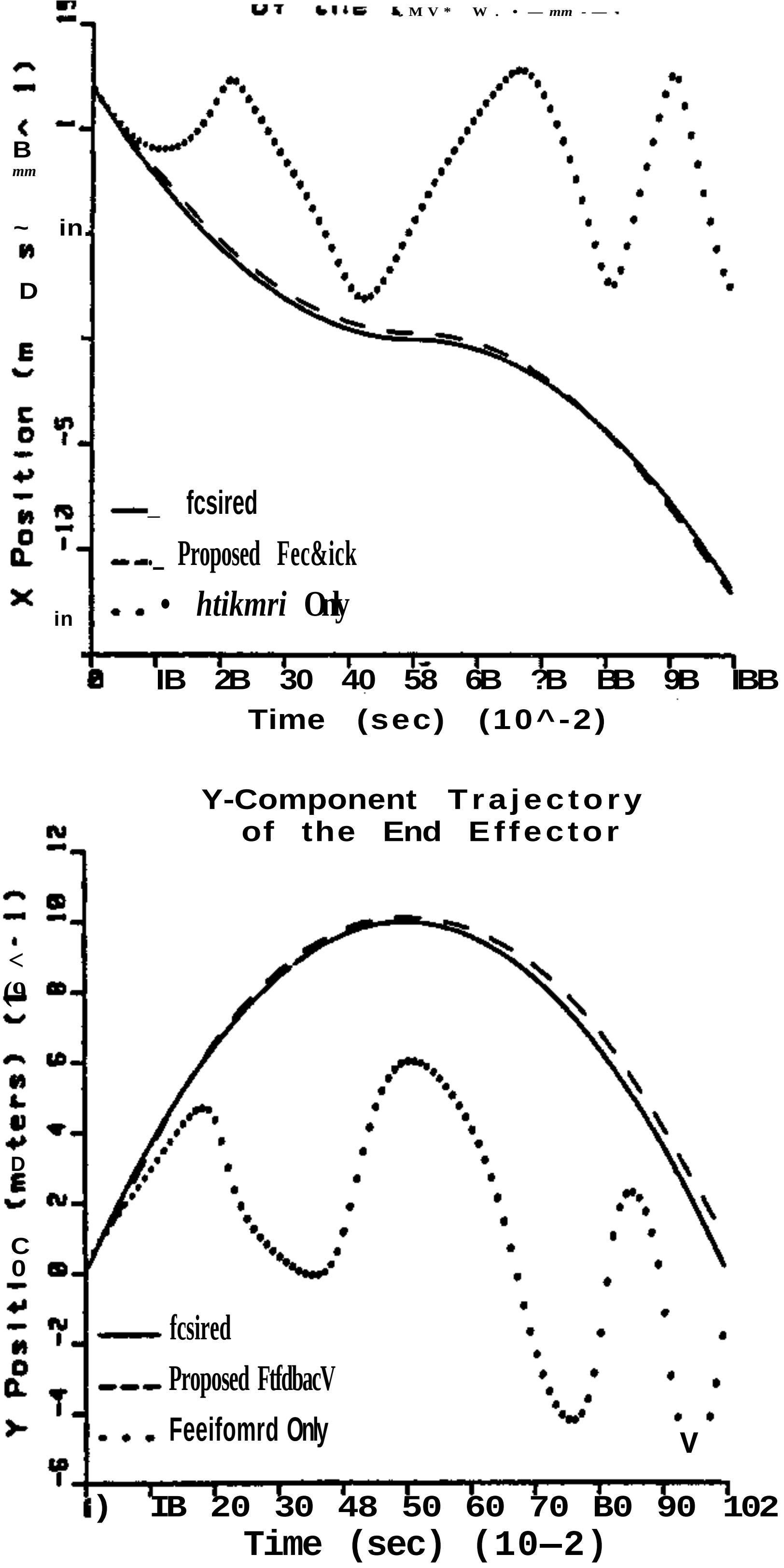
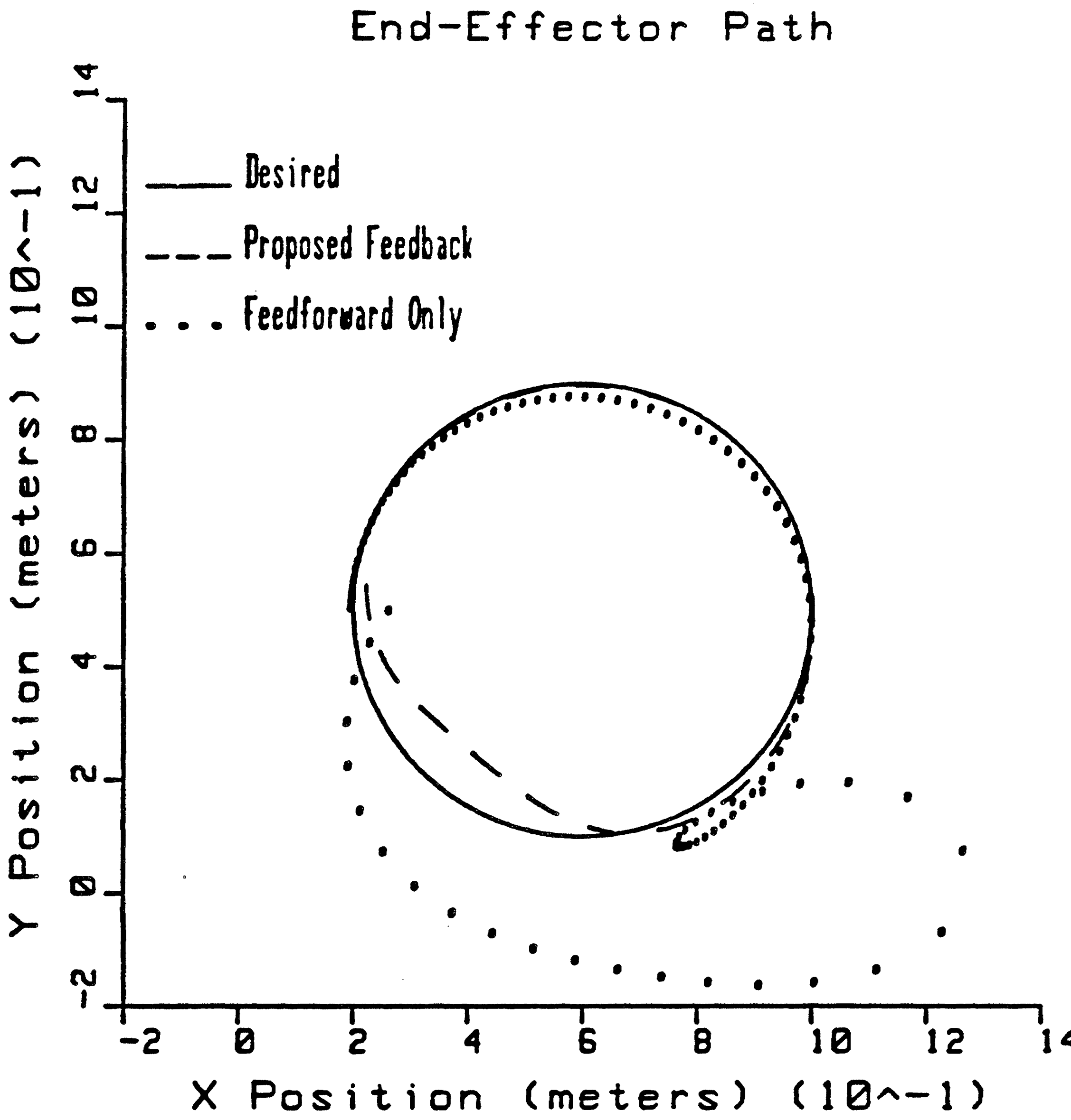


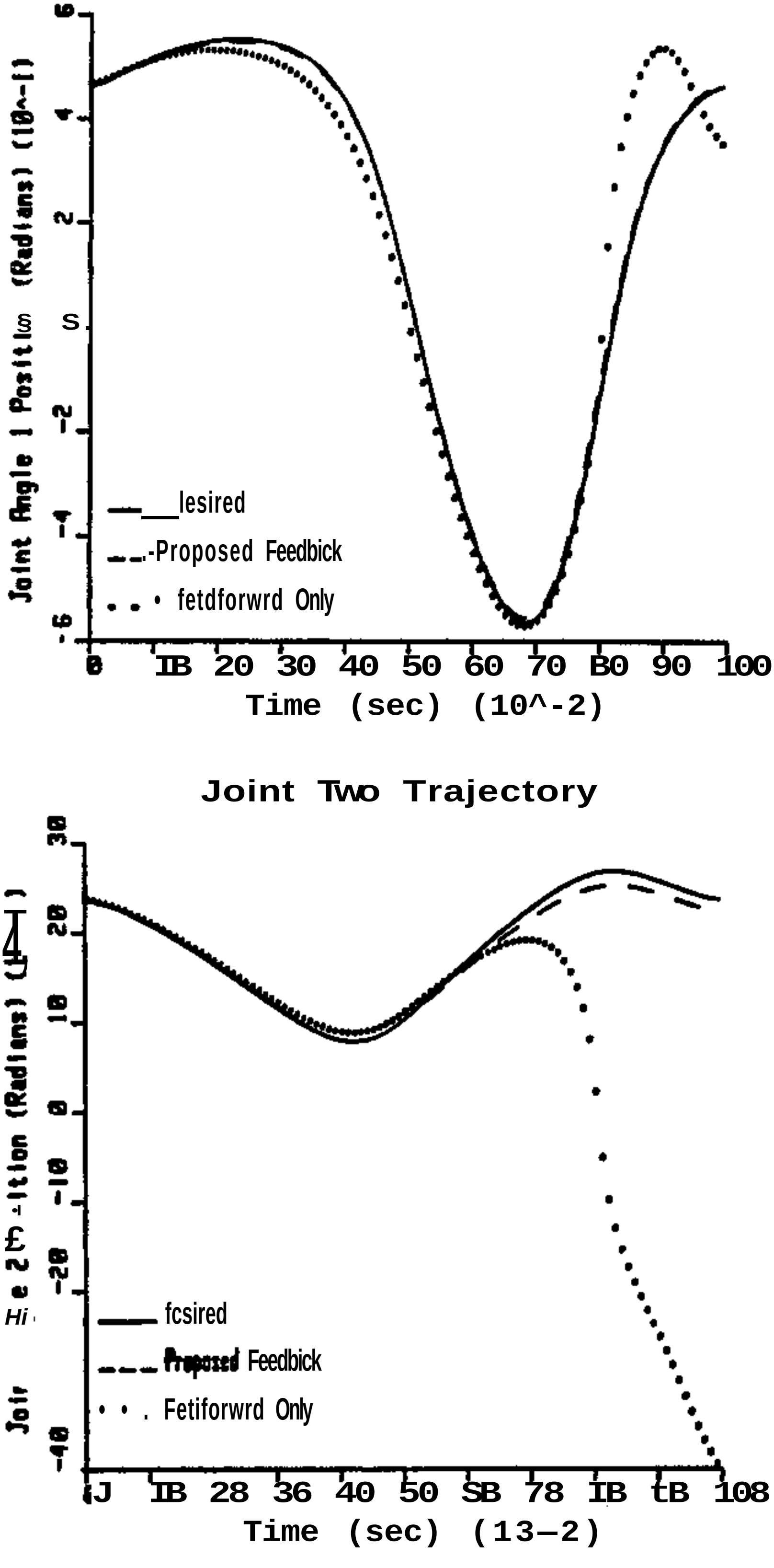
Figure 4.2 Joint trajectories with model errors



Y-Component Trajectory of the End Effector

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


End-Effector Path
Desired
Proposed Feedback
Feedforward Only
Y Position (meters) (10^-1)
X Position (meters) (10^-1)
-2
0
2
4
6
8
10
12
14
-2
0
2
4
6
8
10
12
14



Joint Two Trajectory

Figure 5.2 Joint trajectories with model errors

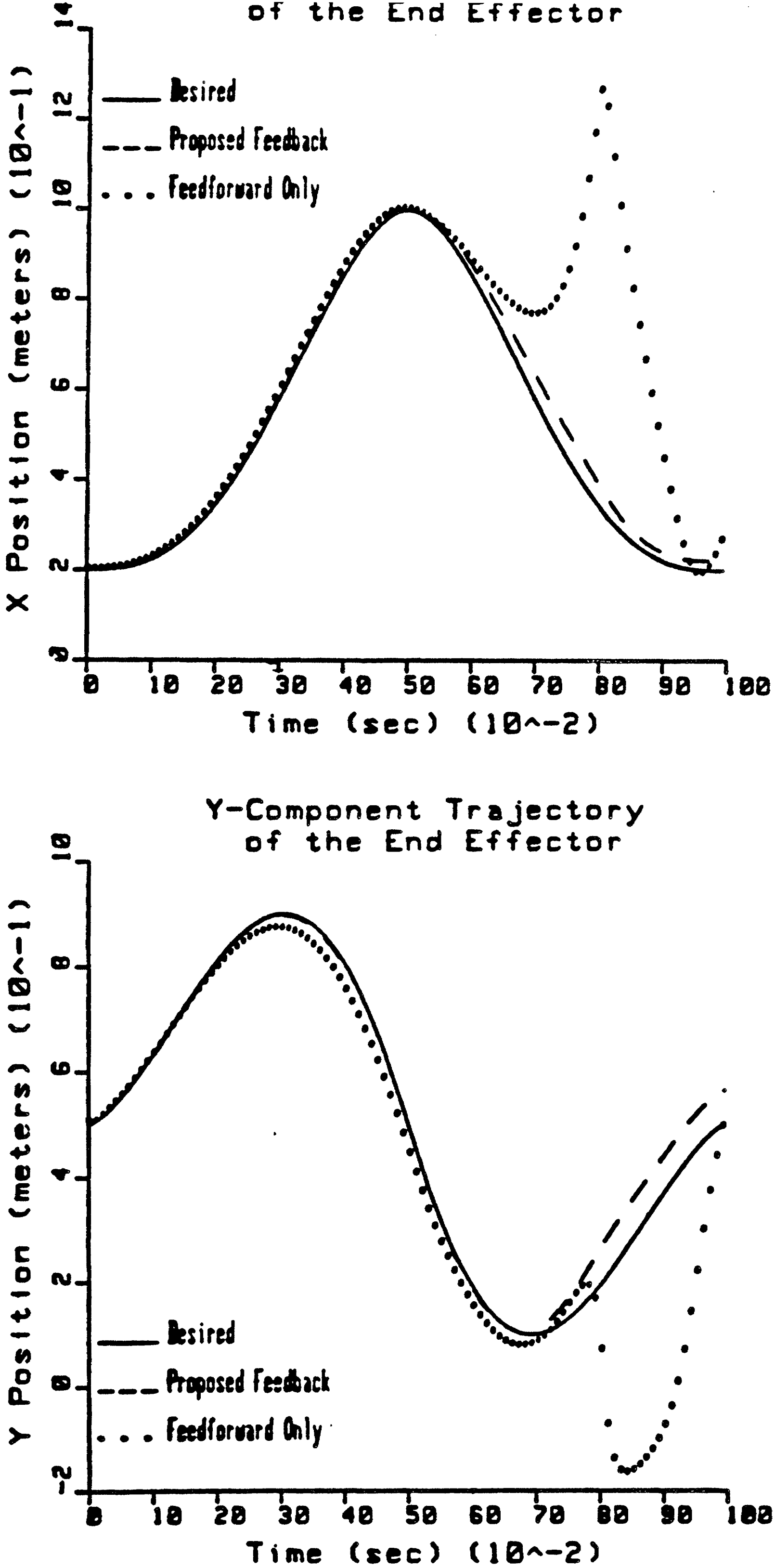


Figure 5.3 X and Y position trajectories of the end-effector with model errors

. COMPUTATIONAL CONSIDERATIONS

The computational time that is required to update Φ, Γ, Q and K, will determine the minimun rror, ϵ_1, that can be used in the control strategy and therefore determine the accuracy of th ajectory following. This update time will therefore affect the maximum sampling rate that ca e used in the feedback loop when on-line trajectory generation is necessary. In many hig ccuracy applications, the update time will be the minimum sampling period allowed, while i ther less critical situations, the use of the old gain matrix, K, during the time needed t alculate the new gain matrix, K_{new}, will not greatly affect the trajectory error. It is obvious tha e want to minimize the update time so that the maximum sampling frequency is increase nough 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**.

.1 Feedback Controller Parameter Calculations

In the two link manipulator simulation, Sylvester's theorem [13] was used in the calculation o . This theorem requires the calculation [11] of the eigenvalues of the system, and then th alculation 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, s written as damped sine and cosine terms, and Γ is calculated by a simple integration of the ine and cosine terms. An alternate method of Φ and Γ calculation is the use of the seri xpansion method. Specifically,

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

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

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

ivolves matrix multiplications, transposes, and the inversion of the matrix, $(F^TQD$. The inversi f the matrix takes the longest to compute, and using the Gaussian elimination method, t umber of operations is $O(n^3)$ for an n x n matrix. All these are standard matrix operations a 3des 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 roportion to $|x_i|_{max}$, which represents the maximum deviation of any joint's position or veloci 'om the desired motion. It is found that in most cases, the updating of Q does not significant ffect the feedback gain matrix, K, so updating Q can be ignored if computational time is ve ritical.

2 Feedforward Computation

Many new robot applications require on-line decision making, database access, and interact! rith other machines. Therefore the inverse dynamics need to be computed in real-time to obta le gross torques of the manipulator joints, which need to be provided by the joint motors. T andard method used to derive the inverse dynamics is the standard Lagrangian formulation. Li ^alker and Paul [10] have shown that this method would require about 7.9 seconds on the PI 1/45 to calculate the gross torques for one position of the Stanford Arm using an efficie Drtran program. This formulation requires a computational effort of $O(n^4)$ because the method subly recursive with many redundant operations. The standard Lagrangian method computes t irques directly using

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

The computational time for this is obviously too long, so various methods of reducing t imber of computations have been tried. Since most of the computational effort is devoted ilculating the triple sums involved in the coriolis and centrifugal forces, many computati ;hemes ignore these terms. The problem with this is that at high speeds, the coriolis a mtrifugal forces dominate in the manipulator dynamics and therefore the burden of compensati increasingly placed on the feedback controller. While this method can work at low speeds, gh speeds this approximation could mean that excessive torques must be applied The control! ight not be capable of doing this and sometimes burnout of equipment could result Alternati .ethods are available using the Newton-Euler [10] or Lagrangian [7] recursive relations. The ethods yield the same torques as the standard Lagrangian approach, but are computational LSter because the standard Lagrangian approach involves redundant operations. These recursi lations reduce the computational effort required to O(n). Luh's Newton-Euler formulation i oating point assembly has been shown to take 4.5 milliseconds on the PDP 11/45 for the torqu alculation of one position of the Stanford Arm. This will allow a sampling rate for th anipulator of greater than 60 Hz which insures good control bandwidth for the manipulator. Th agrangian recursive relations are presented here because the computational formulation for th edback gain matrix, K, is based on this approach.

.2.1 Recursive Lagrangian Dynamics

In the following, the recursive Lagrangian dynamics procedure [7] is used to calculate the joi orques. First, all the W_i^T terms are calculated using equations (17) and going from i=1 to i= hen the D_i and c_i terms are computed from i=n to i=1 using the forward recursive relatio 16). Finally, the torques are computed using equation (15). This formulation has 830n - 59 multiplications and 675n - 464 additions which result in 4388 multiplications and 3586 additio or n=6.

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

where

<u>Forward Recursion</u>

For i = n,...,1

$$D_i = J_i W_i^T + A_{i+1} D_{i+1} \qquad (1$$

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

<u>Backwards Recursion</u>

For i = 1,...,n

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

$$\dot{W}_i = \dot{W}_{i-1} A_i + W_{i-1} \frac{\partial A_i}{\partial q_i} \dot{q}_i \qquad ($$

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

1.3 A and B Matrix Calculations

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

1.3.1 Derivation

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

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

$$\tau_i = \sum_{j=1}^{n}\left[\sum_{k=1}^{j}\left(tr\left(\frac{\partial W_j}{\partial q_i}J_j\frac{\partial W_j}{\partial q_k}\right)\right)\ddot{q}_k+\sum_{k=1}^{j}\sum_{l=1}^{j}\left(tr\left(\frac{\partial W_j}{\partial q_i}J_j\frac{\partial^2 W_j}{\partial q_k\partial q_l}\right)\dot{q}_k\dot{q}_l\right)-m_j g^T\frac{\partial W_j}{\partial q_i}{}^j r_j\right] \quad (18$$

which also can be written in the form [12]

$$\tau_i = \sum_{j=1}^{n} D_{ij}\ddot{q}_j + \sum_{j=1}^{n}\sum_{k=1}^{n} D_{ijk}\dot{q}_j\dot{q}_k + D_i \quad (19$$

where

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

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

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

and where

$$W_j = {}^0V_j = A_1A_2\ldots A_j$$

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

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 th nearization of the inverse dynamics with respect to small perturbations, $\delta\mathbf{q}$.

ı) $\left[\frac{\partial \mathbf{M}}{\partial \mathbf{q}}\ddot{\mathbf{q}} + \frac{\partial \mathbf{f}}{\partial \mathbf{q}}\right]$ 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 th artial derivative of the generalized forces with respect to the joints' position vector. So

$$\left[\frac{\partial M}{\partial q}\ddot{\mathbf{q}} + \frac{\partial f}{\partial q}\right]_{ij} = \frac{\partial}{\partial q_j}\tau_i \qquad i = 1,\ldots,n, \quad j=1,\ldots n \qquad \text{(2C}$$

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

$$\tau_i = \sum_{p=i}^{n}\left[tr(\frac{\partial W_p}{\partial q_i}\mathbf{J}_p\ddot{W}_p^{T}) - m_p g^{T}\frac{\partial W_p}{\partial q_i}{}^{p}r_p\right] \qquad i=1,\ldots,n \qquad \text{(2}$$

where the backward recursive relations for velocities W_p and accelerations W_p are :

$$W_p = W_{p-1}A_p$$

$$\dot{W}_p = \dot{W}_{p-1}A_p + W_{p-1}\frac{\partial A_p}{\partial q_p}\dot{q}_p \qquad \text{(2}$$

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

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

$$\left[\frac{\partial M}{\partial q}\ddot{\mathbf{q}} + \frac{\partial f}{\partial q}\right]_{ij} = \frac{\partial}{\partial q_j}\sum_{p=i}^{n}\left[tr(\frac{\partial W_p}{\partial q_i}\mathbf{J}_p\ddot{W}_p^{T}) - m_p g^{T}\frac{\partial W_p}{\partial q_i}{}^{p}r_p\right] \qquad \text{(2}$$

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

Now

if $j < p$ *then* $\frac{\partial}{\partial q_j}(\frac{\partial W_p}{\partial q_i}) = 0$

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

since $W_p = {}^0W_p = A_1 A_2 \dots A_p$

Consequently, the matrix formulation can be written as

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

for $i = 1,\dots,n$ and $j = 1,\dots,n$

Now a forward recursive relation can be developed by noting that

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

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

Therefore for the two cases of the double derivative we obtain

if $i > j$

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

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

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

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 $i \geq$ considered in what follows.

Rewriting the matrix formulation as

$$\partial q \quad \partial q \quad {}_{p=i} \quad \partial q_i \partial q_j \qquad \partial q_i \qquad \partial q_j \qquad \partial q_i \partial q_j$$

$$\frac{\partial M}{\partial q}\ \ddot{\mathbf{q}}+\frac{\partial f}{\partial q}\ \Big]_{ij}= \sum_{p=i}^{n}\Big[tr(\frac{\partial^2 W_i}{\partial q_i \partial q_j}\ {}^{i}W_p \mathbf{J}_p\ \ddot{W}_p^{\ T})+tr(\frac{\partial W_i}{\partial q_i}\ {}^{i}W_p \mathbf{J}_p\ \frac{\partial \ddot{W}_p^{\ T}}{\partial q_j})-m_p g^{T}\frac{\partial^2 W_i}{\partial q_i \partial q_j}{}^{i}W_p\ {}^{p}r_p\Big] \qquad (27$$

then the reformulation can be written as

$$\frac{\partial M}{\partial q}\ \ddot{\mathbf{q}}+\frac{\partial f}{\partial q}\ \Big]_{ij}=$$

$$\Big[tr(\frac{\partial^2 W_i}{\partial q_i \partial q_j}\sum_{p=i}^{n}{}^{i}W_p \mathbf{J}_p \ddot{W}_p^{\ T})+tr(\frac{\partial W_i}{\partial q_i}\sum_{p=i}^{n}{}^{i}W_p \mathbf{J}_p\ \frac{\partial \ddot{W}_p^{\ T}}{\partial q_j})-g^{T}\frac{\partial^2 W_i}{\partial q_i \partial q_j}\sum_{p=i}^{n} m_p{}^{i}W_p\ {}^{p}r_p\Big] \qquad (28$$

Let

$$\mathrm{D}_i = \sum\nolimits_{p=i}^{n} {}^{i}\mathrm{W}_p \mathbf{J}_p \ddot{\mathrm{W}}_p^{\ T}$$

$$= {}^{i}\mathrm{W}_i \mathbf{J}_i \ddot{\mathrm{W}}_i^{\ T} + \sum\nolimits_{p=i+1}^{n} \mathrm{A}_{i+1}\ {}^{i+1}\mathrm{W}_p \mathbf{J}_p \ddot{\mathrm{W}}_p^{\ T}$$

Now since $\quad {}^{i}\mathrm{W}_i = \mathbf{I}$

we get $\mathrm{D}_i = \mathbf{J}_i \ddot{\mathrm{W}}_i^{\ T} + \mathrm{A}_{i+1}\ \mathrm{D}_{i+1}$ (29

Also, let

$$\mathrm{c}_i = \sum\nolimits_{p=i}^{n} \mathrm{m}_p\ {}^{i}\mathrm{W}_p\ {}^{p}\mathrm{r}_p$$

$$\mathrm{c}_i = \mathrm{m}_i\ {}^{i}\mathrm{r}_i + \mathrm{A}_{i+1}\ \mathrm{c}_{i+1} \qquad (30$$

and

$$\mathrm{N}_i = \sum\nolimits_{p=i}^{n} {}^{i}\mathrm{W}_p \mathbf{J}_p \frac{\partial \ddot{\mathrm{w}}_p^{\ T}}{\partial \mathrm{q}_j}$$

$$\mathrm{N}_i = \mathbf{J}_i \frac{\partial \ddot{\mathrm{w}}_i^{\ T}}{\partial \mathrm{q}_j} + \mathrm{A}_{i+1}\ \mathrm{N}_{i+1} \qquad (31$$

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

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

By a similar procedure we get

for $j \geq i$

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

$$D_j = J_j W_j + A_{j+1} D_{j+1} \qquad (34)$$

$$c_j = m_j \, {}^j r_j + A_{j+1} c_{j+1} \qquad (35)$$

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

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 joint velocity vector. So

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

Now using Waters generalized forces formulation, the matrix becomes

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

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

Consequently the matrix equations are written as

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

Consider first the case of

If i > j

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

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

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

ich leads to the reformulation

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

produces the forward recursive relation by letting

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

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

the matrix compuation is simply formulated as

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

onsidering the other case and by applying similar arguments we get

for $j \geq i$

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

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

n the matrix is formulated as

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

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

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

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

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

For i ≥ j

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

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

the forward recursive relation is

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

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

and the matrix is computed simply by

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

for j ≥ i

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

In this case the matrix formulation and forward recursive relations are

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

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

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 recursive relations needed to calculate these terms are now presented.

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

or p $\geq$ j

$$\begin{aligned}
\frac{\partial W_p}{\partial q_j} &= \frac{\partial W_j}{\partial q_j}\, {}^jW_p \\
\frac{\partial \dot{W}_p}{\partial q_j} &= \frac{\partial \dot{W}_j}{\partial q_j}\, {}^jW_p + \frac{\partial W_j}{\partial q_j}\, {}^j\dot{W}_p \qquad (57) \\
\frac{\partial \ddot{W}_p}{\partial q_j} &= \frac{\partial \ddot{W}_j}{\partial q_j}\, {}^jW_p + 2\frac{\partial \dot{W}_j}{\partial q_j}\, {}^jW_p + \frac{\partial \ddot{W}_j}{\partial q_j}\, {}^jW_p
\end{aligned}$$

and for j $\geq$ p

$$\begin{aligned}
{}^jW_p &= {}^jW_{p-1}\, A_p \\
{}^j\dot{W}_p &= {}^j\dot{W}_{p-1}\, A_p + {}^jW_{p-1}\frac{\partial A_p}{\partial q_p}\, \dot{q}_p \qquad (58 \\
{}^j\ddot{W}_p &= {}^j\ddot{W}_{p-1}\, A_p + 2\,{}^j\dot{W}_{p-1}\frac{\partial A_p}{\partial q_p}\, \dot{q}_p + {}^jW_{p-1}\frac{\partial^2 A_p}{\partial q_p^{\,2}}\, \dot{q}_p^{\,2} + {}^jW_{p-1}\frac{\partial A_p}{\partial q_p}\, \ddot{q}_p
\end{aligned}$$

For j = 1,...,n

$$\begin{aligned}
\dot{W}_j &= \dot{W}_{j-1}\, A_j + W_{j-1}\frac{\partial A_j}{\partial q_j}\, \dot{q}_j \\
\frac{\partial \dot{W}_j}{\partial q_j} &= \dot{W}_{j-1}\, \frac{\partial A_j}{\partial q_j} + W_{j-1}\frac{\partial^2 A_j}{\partial^2 q_j}\, \dot{q}_j \qquad (5
\end{aligned}$$

$$\begin{aligned}
\ddot{W}_j &= \ddot{W}_{j-1}\, A_j + 2\dot{W}_{j-1}\frac{\partial A_j}{\partial q_j}\, q_j + W_{j-1}\frac{\partial^2 A_j}{\partial q_j^2}\, \dot{q}_j^{\,2} + W_{j-1}\frac{\partial A_j}{\partial q_j}\, \ddot{q}_j \\
\frac{\partial \ddot{W}_j}{\partial q_j} &= \ddot{W}_{j-1}\, \frac{\partial A_j}{\partial q_j} + 2\dot{W}_{j-1}\frac{\partial^2 A_j}{\partial q_j^2}\, \dot{q}_j + W_{j-1}\frac{\partial^3 A_j}{\partial q_j^3}\, \dot{q}_j^{\,2} + W_{j-1}\frac{\partial^2 A_j}{\partial q_j^2}\, \ddot{q}_j \qquad (6
\end{aligned}$$

Note also that

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

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

$$\frac{8^2 w_P}{\partial q_p \partial q_j} = \text{iv!.,} \frac{3/4_J}{\partial q_j} \wedge w \wedge \frac{84_P}{\partial \bar{q}_p}$$ **(6;**

$$\frac{\partial^2 W_p}{\partial q_p^{\,2}} = w_{p-1}\frac{B^2 A}{\wedge_p} \quad \text{for } j = p$$ (6:

1.4 The Summary of Recursive Relations

Now, to summarize the procedure for computing the M" , $\frac{3M.}{Oq}$™r~q + $\frac{?}{Oq}$, and $\frac{df}{dq}$7, matrices. Firs he backward recursive relations (64) are used to compute all the W_i^T terms from i· = 1 to i^s: "hen all the $\frac{3w_i^T}{Oq_j}$> $\frac{3\dot{w}_i^T}{Oq_j}$, $\frac{8\ddot{w}_i^T}{Oq_j}$ -r— terms are computed by the recursive relations (65), (66) and (61 or i=l to i=n and j=l to j=n, but only for the cases of i £ j. Next the forward recursn elations (68) and (69) are used to calculate D_i and c_i for i=n to i=l, and relations (70), (7i nd (72) are used to calculate P , Q , N for j=l to j=i. Finally, the necessary contn aatrices, M"\ ^$\ddot{q}$ + $\frac{\partial f}{dq}$ and 21 $\dot{d}q$ are computed by (73), (74), (75), (76), (77) and (78) f(^{r}l to i=n and j=l to j=n. Noting that many of the terms are the same as those calculated f< he feedforward computations if the feedforward calculation is incorporated in the control loo] hen many of these computations need not be repeated.

1.4.1 Backwards Recursion

For i s 1....n

$$W_1 = W_{1-1}^{\,t} A_1$$

$$\dot{W}_i = \dot{W}_{i-}, A_i + W_{.}, \frac{\partial A_i}{"a^q}\dot{q}_i$$ (6

$$\ddot{W}_1 = \ddot{W}_{1-1}, A_1 + 2\dot{W}_{1-1.}, \frac{BA_i}{3<7.}\dot{q}_{\wedge i} + W_{1-1}, \frac{d^2 A_i}{dq-}\dot{q}^2_{\wedge 1} + W_{1-1}, \frac{3A}{dq!}\ddot{q}_{\wedge 1}$$

For $\underline{j \leq i}$

$$ {}^{j}W_{i} = {}^{j}W_{i-1} \, A_{i} $$

$$ {}^{j}\dot{W}_{i} = {}^{j}\dot{W}_{i-1} \, A_{i} + {}^{j}W_{i-1} \frac{\partial A_{i}}{\partial q_{i}} \, \dot{q}_{i} $$

$$ {}^{j}\ddot{W}_{i} = {}^{j}\ddot{W}_{i-1} \, A_{i} + 2\,{}^{j}\dot{W}_{i-1} \frac{\partial A_{i}}{\partial q_{i}} \, \dot{q}_{i} + {}^{j}W_{i-1} \frac{\partial^{2} A_{i}}{\partial q_{i}^{2}} \, \dot{q}_{i}^{2} + {}^{j}W_{i-1} \frac{\partial A_{i}}{\partial q_{i}} \, \ddot{q}_{i} $$

For $\underline{j = 1,\ldots,n}$

$$ \frac{\partial \dot{W}_{j}}{\partial q_{j}} = \dot{W}_{j-1} \, \frac{\partial A_{j}}{\partial q_{j}} + W_{j-1} \frac{\partial^{2} A_{j}}{\partial^{2} q_{j}} \, \dot{q}_{j} $$

$$ \frac{\partial \ddot{W}_{j}}{\partial q_{j}} = \ddot{W}_{j-1} \, \frac{\partial A_{j}}{\partial q_{j}} + 2\dot{W}_{j-1} \frac{\partial^{2} A_{j}}{\partial q_{j}^{2}} \, \dot{q}_{j} + W_{j-1} \frac{\partial^{3} A_{j}}{\partial q_{j}^{3}} \, \dot{q}_{j}^{2} + W_{j-1} \frac{\partial^{2} A_{j}}{\partial q_{j}^{2}} \, \ddot{q}_{j} $$

For $\underline{i \geq j}$

$$ \frac{\partial W_{i}}{\partial q_{j}} = \frac{\partial W_{j}}{\partial q_{j}} \, {}^{j}W_{i} $$

$$ \frac{\partial \dot{W}_{i}}{\partial q_{j}} = \frac{\partial \dot{W}_{j}}{\partial q_{j}} \, {}^{j}W_{i} + \frac{\partial W_{j}}{\partial q_{j}} \, {}^{j}\dot{W}_{i} $$

$$ \frac{\partial \ddot{W}_{i}}{\partial q_{j}} = \frac{\partial \ddot{W}_{j}}{\partial q_{j}} \, {}^{j}W_{i} + \frac{\partial \dot{W}_{j}}{\partial q_{j}} \, {}^{j}\dot{W}_{i} + \frac{\partial W_{j}}{\partial q_{j}} \, {}^{j}\ddot{W}_{i} $$

4.4.2 Forward Recursion

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

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

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

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

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

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

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

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

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

For i ≥ j

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

For j ≥ i

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

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

L ∂q q+ ∂q J..=» I t r y U.)-T try /v..; - y c. I V /J
,^ , Sq.Sq '

If j £ i

L——q+ — I » \tr(———D.)+ tr(———N.) - g' ———c\ (76
LBq BBqq J,J L 3^ ' 3g " Bqfa 'J

c) $\left[\frac{3fT}{\partial q}\right]$ *term.*

for i ^ j

$$\left[\frac{3\wedge}{\partial q}\right] = tr(\frac{aw_i}{\partial q_j} Q.) \qquad (11$$

for j > i

$$\left[\frac{Bf,}{\partial q_{JIJ}}1\right] - tr(\frac{\partial W_i}{a<7_(}W.Q..)_{J} \qquad (78$$

The number of multiplications involved with the matrix calculations is $1062n^2$ - 102In - 12 nd the number of additions is $1037n^2$ - 621n -96. This means that for n=6, the number c multiplications is 40,594 and the number of additions *is* 37,926 for each update of the A and matrices. Therefore, the number of multiplications and additions is of n^2 dependence and for n= be number of operations *is* 10 times the number of operations involved in the recursi\ ,agrangian dynamics relations.

•5 Recursive Parametric Matrices Using 3 x 3 Matrices

The previous formulation reduces the computational effort to $O(n^2)$ for each matrix, which lie lowest order that can be achieved. The only way to further reduce the computational cost D use 3 x 3 rotation matrices instead of 4 x 4 rotation-translation matrices. The 4 x 4 matric€ re inefficient because of some sparseness and because of the combination of translation wit otation [7]. The 4 x 4 matrices require 64 multiplications for each matrix multiplication, whil · x 3 matrices only require 27 multiplications, so a 58% reduction in coefficient multiplication 5 effected.

The 3 x 3 rotation matrix A_j relates the orientations of coordinate systems j-1 and j, and V ierivation of the formulations for computing M^{-1}, $\frac{3M.\#}{8q}q$ + $\frac{3f}{3q}$, and $\frac{3f}{d\dot{q}}$, using 3x3 matrices is

•resented in Appendix B. The procedure for calculating the M , $\frac{\partial M..}{dq}q$ $^{+}$ $\frac{8f}{\partial q}$, and $\frac{8f}{d\dot{q}}$, matrices ising 3 x 3 rotation matrices is now summarized. First, the backward relations (64), (65

66), (67) and (79) are used to compute all the $\frac{3w_i^T}{oq_j}$, $\frac{3\dot{w}_i^T}{oq_j}$, $\frac{8\ddot{w}_i^T}{oq_j}$, and the $\frac{8p_i^T}{oq_j}$, $\frac{8\dot{p}_i^T}{oq_j}$, $\frac{\partial\ddot{p}_i^T}{dq_j}$, srms for i=l to i=n and j=l to j=l. Next, the forward recursive relations (80), (81) and (8^ re used to calculate D; e and c for i=n to i=l, and relations (83), (84), (85), (86), (87) ar 88) are used to calculate P_{ij}, k_{ij}, Q_{ij}, b_{ij}, N_{ij}, I_{ij}, for j=l to j=i. Finally, the necessary contn

latrices, M^{-1}, $\frac{3M..}{oq}q$ + $\frac{3f}{oq}$ and $\frac{3f}{o\dot{q}}$ are computed by (89), (90), (91), (92), (93) and (94) for

=1 to i=n and j=l to j=n.

***ISA* Backwards Recursion**

The $\frac{3w_i^T}{dq_j}$, $\frac{8\dot{w}_i^T}{dq_j}$, $\frac{8\ddot{w}_i^T}{dq_j}$ terms are calculated with the same recurrence relations (64), (65), (66 nd (67) as before except the matrices are now 3 x 3 .

For i = l,..,n

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

For j = 1,...,i

$$\frac{\partial p_i}{dq_j} = \frac{\partial p_{i-1}}{8q_j} - \frac{\partial w_i}{dq_j} \, {}^ip_i^*$$

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

$$\frac{8\ddot{p}_i}{dq_j} - \frac{3\ddot{p}_{i-1}}{dq_j} - \frac{3\ddot{W}_i}{dq_j} \, V \qquad (7S$$

! Forward Recursion

For i « n,...,l

$$D_i = J_i W_i^{\cdot\cdot T} + {}^i n_i^T p_{*i}^T + A_{i+1} D_{i+1} + {}^J p_{*i+1} e_{i+1} \quad (80)$$

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

$$c_i = m_i {}^{\cdot}r_i + A_{i+1} c_{i+1} \quad (82)$$

For i - 1....i

$$^pn \ * \ A+>^pi+u^+ \ \ ^i+.^+u^+ \ \ W^{\wedge}r^{i} \ + \ ^Jrr\sim^{i} \quad (83)$$

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

$$Q_{ij} = , \ \ Q \ \ + {}^j p \ 6 \ \ + {}^l n7—r_j^T + J,\frac{\partial \dot{W}_i^T}{\partial q_j} \quad (85)$$

$$*« \ - \ {}^6 i+u^+ \ {}^m r\frac{\partial \dot{p}^T}{\partial \hat{q}_j} \ + \ in.^{T'"}\frac{\partial \dot{W}_i^T}{\partial q_j} \quad (86)$$

$$N_{ij} = A_{i+1}N_{i+1j} + {}^i p_{i+1} I_{i+1j} + {}^i n_i^T\frac{\partial \ddot{p}_i^T}{\partial q_j} + {}^J{}_i\frac{\partial \ddot{W}_i^T}{\partial q_j} \quad (87)$$

$$I_{ij} = I_{i+1j} + »>,\frac{\partial \ddot{p}_i^T}{\partial q_j}\wedge"*r\frac{\partial \ddot{W}_i^T}{\partial q_j} \quad (88)$$

^{s}ori=1....n , j=1,...,n

[a] M_{ij} term:

For i ≥ j

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

For $j \geq i$

$$M_{ij} = tr(\frac{\partial W_i}{\partial q_i} {}^{i}W_j \; P_{ii}) \qquad (9$$

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

If $i > j$

$$\left[\frac{\partial M}{\partial q} \ddot{\mathbf{q}} + \frac{\partial f}{\partial q}\right]_{ij} = \left[tr(\frac{\partial^2 W_i}{\partial q_i \partial q_j} D_i) + tr(\frac{\partial W_i}{\partial q_i} N_{ij}) - g^T \frac{\partial^2 W_i}{\partial q_i \partial q_j} c_i\right] \qquad (9$$

If $j \geq i$

$$\left[\frac{\partial M}{\partial q} \ddot{\mathbf{q}} + \frac{\partial f}{\partial q}\right]_{ij} = \left[tr(\frac{\partial^2 W_i}{\partial q_i \partial q_j} D'_i) + tr(\frac{\partial W_i}{\partial q_i} N_{ii}) - g^T \frac{\partial^2 W_i}{\partial q_i \partial q_j} c_i\right] \qquad (9$$

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

for $i \geq j$

$$\left[\frac{\partial f}{\partial \dot{q}}\right]_{ij} = tr(\frac{\partial W_i}{\partial q_i} Q_{ij}) \qquad (9$$

for $j > i$

$$\left[\frac{\partial f}{\partial \dot{q}}\right]_{ij} = tr(\frac{\partial W_i}{\partial q_i} {}^{i}W_j Q_{ii}) \qquad (9$$

The number of multiplications involved with the recursive 3 x 3 relations is $739n^2 + 62n$ -5
nd the number of additions is $(1161/2)n^2 - (19/2)n - 36$. For n=6 the number
nultiplications for each update of **A** and **B** is 26922 and the number of additions is 20805. Th
s a greater than 40% reduction in the number of operations over using 4 x 4 rotation-translatio
natrices.

CONCLUSION

rhis report has presented a control scheme for accurate trajectory following with robotic anipulators. The technique has been based on the use of measured joint displacements anc slocities to generate corrective torques through an adaptive controller that eliminates deviation: f the manipulator from the desired trajectory under feedforward control, in the least square inse. The controller has taken into account dynamic nonlinearities (coriolis and centrifuga ccelerations, pay-load change, etc.), geometric nonlinearities (nonlinear transformation matrices; hysical nonlinearities (e.g., coulomb damping), dynamic coupling between joints, and real-tim hanges in the desired trajectory. Simulation results have been presented for a two-degree-oi reedom manipulator. These results have indicated the effectiveness and robustness of tl ;ontroller. The stability issue has been addressed. Recursive relations have been developed compute the adaptive feedback gains, thereby improving the computational efficiency of the scher hat makes the controller feasible under real-time changes in the desired trajectory. Two metho 3f 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-c freedom manipulator, the 3x3 Lagrangian recursive relations involve 47,727 operations, which 41% more efficient than the alternative method of using 4 x 4 rotation-translation matrices. 1 number of operations involved in updating the feedback gain matrix would limit the maxima update frequency to about 3 Hz when used with computers like the PDP 11 for six degree-, freedom manipulators.

REFERENCES

[I] **Asada, H., and Hanafusa, H.**
An Adaptive Tracing Control of Robots and Its Application to Automatic Weldin
***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 CADICAM and Robotics Conference , Ju

[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, 19**

[7] **Hollerbach, J.M.**
A Recursive Lagrangian Formulation of Manipulator Dynamics and A Comparativi of Dynamics Formulation Complexity.
IEEE Transactions on Systems, Man, and Cybernetics SMC-10(II):730-736, 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-Mel Ion Univers Pittsburgh, PA* , November, 1980.**

[9] **Klein, C.A., and Maney, J.J.**
Real-Time Control of a Multiple-Element Mechanical Linkage with a Microcompi
***IEEE Transactions on Industrial Electronics and Control Instrumentation I.* 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: June, 1980.**

[II] **Melsa, J.L., and Jones, S.K.**
Computer Programs for Computational Assistance in the Study of Linear G ***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.

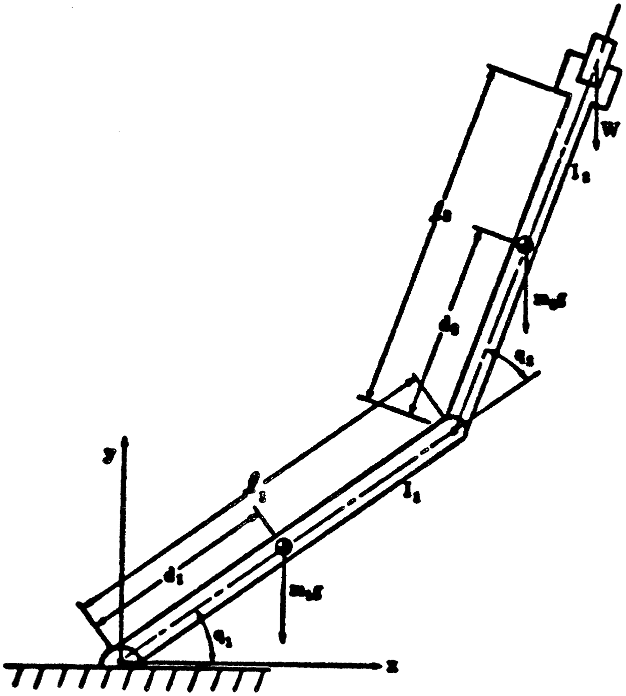


Figure A.1 Nomenclature for the two-link manipulator

$$\mathbf{q} = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}$$

A.1 Kinematics

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

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

$$\begin{bmatrix} \delta u_x \\ \delta u_y \end{bmatrix} = \mathbf{J}\ \delta\mathbf{q}$$

Velocity

$$\dot{q} = J^{-1} [\; v_x \quad v_y \;]^T \qquad \text{(A.5)}$$

Joint Accelerations

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

$$\ddot{\mathbf{q}} = \mathbf{J}^{-1} \left[\; [\; a_x \; a_y]^T - \frac{\partial J}{\partial t} \dot{\mathbf{q}} \; \right] \qquad \text{(A.6}$$

..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 :

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

$$= -\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & V & \mathrm{Minv}_{12} & AM_{11} & AM_{12} & Af_{11} & Af_{12} \\ 0 & 0 & \mathrm{Minv}_{12} & \mathrm{Minv}_{22} & AM_{21} & AM_{22} & Af_{21} & Af_{22} \end{bmatrix}$$

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

where

$$\mathrm{Minv}_{12} = -(I_2^*+I_3^*/2\ \cos\theta_2)$$

$$\mathrm{Minv}_{22} = (I_1^*+I_2^*+I_3^*\cos\theta_2$$

$$AM_{11} = I_2^*\sin\theta_2[(W_2^*+I_3^*)\sin\theta_1+W_2^*\sin(\theta_1+\theta_2]$$

$$AM_{12} = -I_2^*\sin\theta_2[2\theta_1+\theta_2]+I_2^*\cos\theta_2[2\theta_1+\theta_2^2]+I_2^*\sin\theta_2W_1^*\sin(\theta_1+\theta_2$$

$$AM_{21} = I_2^*\sin\theta_2W_1^*\sin(\theta_1+\theta_2)$$

$$AM_{22} = -I_2^*\sin\theta_2\theta_1+I_2^*\cos\theta_1\theta_2^2$$

$$Af_{11} = -2I_2^*\sin\theta_1$$

$$Af_{12} = -I_2^*\sin\theta_2(\theta_1+\theta_2)$$

$$Af_{21} = 2I_2^*\theta_1$$

$$Af_{22} = 0$$

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 develope using 3 x 3 rotation matrices.

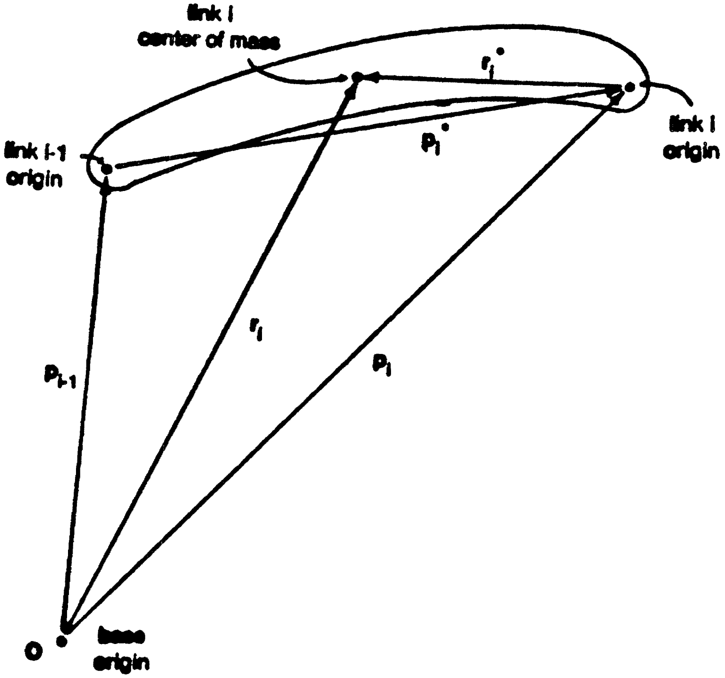


Figure B.1 3x3 Vector definitions

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

$$f_i = \sum_{p=1}^{n} \left[tr\left(m_p \frac{\partial p_p}{\partial q_i} \ddot{p}_p^T + \frac{\partial p_p}{\partial q_i} {}^pn_p^T \ddot{W}_p^T + \frac{\partial W_p}{\partial q_i} {}^pn_p^T \ddot{p}_p^T + \frac{\partial W_p}{\partial q_i} J_p \ddot{W}_p \right) - m_p g^T \frac{\partial W_p}{\partial q_i} {}^pr_p \right] \qquad ($$

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

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

Now for the case where i ≤ j

$$p_p = p_i + W_i{}^ip_i$$

$$\frac{\partial p_p}{\partial q_i} = \frac{\partial W_i}{\partial q_i}\,{}^ip_p$$

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

For $i \geq j$

$$\frac{\partial^2 p_p}{\partial q_i \partial q_j} = \frac{\partial^2 W_i}{\partial q_i \partial q_j}\,{}^ip_p$$

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

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

$$D_i = \sum_{p=i}^{n} (m_p {}^x p j p_p^{\cdot T} + {}^j p_{*p} {}^p n_p {}^T \ddot{W}_p + \dot{W}_P {}^p /7_p {}^T p_{*P}^{\cdot T} + \dot{W}\, J W_p \ddot{}_p)$$

$$D_1 \bullet 0 + 0 + {}^i /i_{1}{}^T p_{*1}^{T} + J.W_{ii}^T +$$

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

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

where

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

$$e_i = e_{i+1} + m_i \ddot{p}_i{}^1 + tnf\ddot{W}_i^* \qquad (5,9)$$

Similarly,

$$N_i = \sum_{p=i}^{n} (m_p {}^i p_p \frac{\partial \ddot{p}^T}{d<7_j} \quad \frac{\partial \ddot{W}}{da_j} \quad \frac{\partial \ddot{p}^T}{da_j} \quad \frac{\partial \ddot{W}}{da_j}$$

$$N_i = A_{i+1} N_{i+1} + {}^i p_{i+1} I_{i+1} + {}^i n_i{}^T \frac{\partial \ddot{p}_j^T}{} + J_i \frac{\partial \ddot{W}_i{}^T}{} \qquad (B.10)$$

where

$$', - Z_{p=i} (\frac{\partial \ddot{p}_p^T}{\hat{d}<7_j} + {}^p n_p{}^T \frac{\partial \ddot{W}_p{}^T}{\partial q_j})$$

$$I_i = I_{i+1} + m. \frac{\partial \dot{p};^T}{'a<7j} + {}^j /7.^T \frac{\partial \dot{W}_i{}^T}{' \; a\wedge} \qquad (5.11)$$

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

i ≥ j

$$r_{,}\frac{{}^{*M}_{O}}{\sigma q} {}^{M}\ddot{q} + \frac{A/ \; {}^{O}}{\sigma q}]^{f}..-{}^{T} ff/-(\frac{{}^r \hat{\partial}^2 W}{\sigma q_i \sigma q_j} {}^P D) + tr(\frac{dW}{\sigma q_i} {}^P N) - g^7 - \frac{\hat{6}^2 W}{\sigma q_i \sigma q_j} {}^{\top\, L\, P} c \; \backslash \qquad (BAD$$

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

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

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

Now,

$$\frac{\partial \ddot{p}_p}{\partial \mathring{q}_i} = \frac{\partial \mathring{p}_p}{\partial q_i} \qquad (B.14)$$

$$\frac{\partial \ddot{W}_p}{\partial \mathring{q}_i} = \frac{\partial \dot{W}_p}{\partial q_i} \qquad (B.15)$$

For $i \geq j$

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

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

Therefore

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

Let

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

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

where

$$b_i = \sum_{p=i} (m_p \frac{\partial p_p}{\partial q_j} + {}^p n_p{}^T \frac{\partial W_p}{\partial q_j})$$

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

r 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) **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

$$k_i = k_{i+1} + m_i \frac{\partial p_i{}^T}{\partial q_j} + {}^i n_i{}^T \frac{\partial W_i{}^T}{\partial q_j} \qquad (B.2$$

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

Since

$$p_{i-1} = p_i + w_i \quad \forall i$$

Then

$$p_i = p_{i-1} - w_i \quad \forall i$$

and for $j \neq i$

$$\frac{\partial p_i}{\partial q_j} = \frac{\partial p_{i-1}}{\partial q_j} - \frac{\partial w_i}{\partial q_j} \bigg|_{p_i^*}$$

$$\frac{\partial p_i}{\partial q_j} = \frac{\partial p_{i-1}}{\partial q_j} - \frac{\partial w_i}{\partial q_j} \bigg|_{p_i^*}$$

$$\frac{\partial p_i}{\partial q_i} = \frac{\partial p_{i-1}}{\partial q_i} - \frac{\partial w_i}{\partial q_i} \bigg|_{p_i^*}$$