INCORPORATION OF PAYLOAD UNCERTAINTIES INTO ROBOT TRAJECTORY PLANNING

Kang G. Shin

Department of Electrical Engineering and Computer Science
The University of Michigan
Ann Arbor, Michigan 48109

Neil D. McKay

Computer Science Department
General Motors Research Laboratories
Warren Michigan 48090

ABSTRACT

A number of trajectory planning algorithms are available for determining a time history of the joint torques, positions, and velocities required to move a manipulator along a given geometric path in minimum time. These schemes require knowledge of the robot's dynamics, which in turn depend upon the characteristics of the payload which the robot is carrying. In practice, the dynamic properties of the payload will not be known exactly, so that the dynamics of the robot, and hence the required joint torques, must be calculated for a nominal set of payload characteristics. But since these trajectory planners generate nominal joint torques which are at the limits of the robot's capabilities, moving the robot along the desired geometric path at speeds calculated for the nominal payload may require torques which exceed the robot's capabilities. In this paper, bounds on joint torque uncertainties are derived in terms of payload uncertainties. This allows the trajectory planner to incorporate payload uncertainties into the trajectory planning process.

1. INTRODUCTION

Various algorithms are available for performing trajectory planning for robots, i.e., generating a time history of desired positions, velocities, accelerations and torques [1, 4]. These trajectory planners require knowledge of the robot's dynamics, which in turn depend upon the characteristics of the payload being carried. In practice, the exact characteristics of the payload will not be known; since the trajectory planners referenced above need to know the exact dynamics of the robot, the trajectory planning process must be carried out with dynamics which are calculated for a nominal payload. This practice can lead to difficulties. To see why this is the case, note that these trajectory planners generate nominal torques which are at the limits of the robot's capabilities *for the given dynamics* Moving the robot along the desired path at speeds calculated for the nominal payload may therefore require torques which are beyond the robot's capabilities if the payload differs from the nominal one. If the robot's joints are controlled by independent servoes, as is usually the case, then attempting to make the robot move along the nominal trajectory will result in one or more joints "falling behind", so that the robot strays from the desired geometric path. In other words, the trajectory generated by the trajectory planner is realizable for the nominal payload, but not for the actual payload.

There are a number of adaptive controllers which can compensate for the changes in load, provided that the plant (i.e. the robot joint drive) does not saturate [2]. However if the plant saturates, as may happen if the actual and nominal payloads differ too much, then these controllers cannot possibly compensate for load changes. It is the objective of this paper to present an analysis of the torque errors caused by payload changes, and incorporate the error information into the trajectory planning process so as to avoid saturation of the individual actuators.

Changes in payload characteristics will be expressed as errors in the *pseudo-inertia* of the payload; the pseudo-inertia is a matrix containing the mass and first and second moments of the payload. It will be shown that bounds on the joint torque errors can be calculated in terms of the norm of the error in the pseudo-inertia of the payload, given the robot's kinematics. The general trajectory planning algorithm given in [5] can then be modified to handle uncertainties in the dynamics caused by the payload. If the actual and nominal payloads are described by the pseudo-inertias I_A and I_N respectively, then for a given positive real number E, the algorithm generates a trajectory which is realizable for *all* payloads I_A for which $\|I_A - I_N\| \leq E$.

In order to determine errors in the torques, the dynamic equations of the robot are required. In tensor notation, the dynamic equations describing the behavior of a robot take the general form

$$\mathbf{u}_i = \mathbf{J}_{ij}\ddot{\mathbf{q}}^j + \mathbf{C}_{ijk}\dot{\mathbf{q}}^j\dot{\mathbf{q}}^k + \mathbf{R}_{ij}\dot{\mathbf{q}}^j + \mathbf{g}_i \qquad (1.1)$$

where $\mathbf{u}_i$ is the i^{th} generalized force, $\mathbf{q}^i$ is the i^{th} generalized coordinate, $\mathbf{J}_{ij}$ is the inertia matrix, $\mathbf{C}_{ijk}$ is an array of Coriolis coefficients, defined by

[1]The work reported here is supported in part by the NSF grant No. ECS-8409836 and the US AFOSR contract No. F33615-85-C-5105. Any opinions, findings, and conclusions or recommendations in this paper are those of the authors and do not necessarily reflect the view of the funding agencies.

$$\mathbf{C}_{ijk} = \frac{1}{2}\left(\frac{\partial \mathbf{J}_{ij}}{\partial \mathbf{q}^k} + \frac{\partial \mathbf{J}_{ik}}{\partial \mathbf{q}^j} - \frac{\partial \mathbf{J}_{jk}}{\partial \mathbf{q}^i} \right) \tag{1.2}$$

The matrix $\mathbf{R}_{ij}$ is the viscous friction matrix, and $\mathbf{g}_i$ is the gravitational force. The summation convention has been used here, so that all product terms in (1.1) are summed from 1 to N over repeated indices, where N is the number of degrees of freedom of the robot. The inertia matrix $\mathbf{J}_{ij}$, the Coriolis array $\mathbf{C}_{ijk}$ and the gravitational loading vector $\mathbf{g}_i$ are all functions of the position of the robot and the payload pseudo-inertia.

It will be assumed that the path which the robot is expected to follow has been given as a parameterized curve in joint space, i.e., the joint coordinates $\mathbf{q}^i$ are given in terms of a single scalar λ by the equations

$$\mathbf{q}^i = f^i(\lambda), \quad 0 \le \lambda \le \lambda_{\max}. \tag{1.3}$$

This allows all of the joint positions, velocities, and accelerations to be expressed in terms of the scalar parameter λ and its time derivatives. Plugging these relations into the dynamic equations gives torque in terms of these quantities also. More specifically, we have

$$\mathbf{u}_i = \mathbf{J}_{ij}\frac{df^j}{d\lambda}\dot{\mu} + \left\{ \mathbf{J}_{ij}\frac{d^2 f^j}{d\lambda^2} + \mathbf{C}_{ijk}\frac{df^j}{d\lambda}\frac{df^k}{d\lambda} \right\}\mu^2 + \mathbf{R}_{ij}\frac{df^j}{d\lambda}\mu + \mathbf{g}_i \tag{1.4}$$

where $\mu \equiv \dot{\lambda}$ is the *pseudo-velocity*. The quantities $\mathbf{J}_{ij}$, $\mathbf{C}_{ijk}$, and $\mathbf{g}_i$ depend upon the masses and moments of inertia of the robot's links. The robot's payload is fixed to the last link, and hence must be regarded as part of the last link for purposes of calculating the dynamic coefficients. Therefore $\mathbf{J}_{ij}$, $\mathbf{C}_{ijk}$, and $\mathbf{g}_i$ will change as the characteristics of the payload vary.

We may write Eq. (1.4) as

$$\mathbf{u}_i = M_i(\lambda, I_N)\dot{\mu} + Q_i(\lambda, I_N)\mu^2 + R_i(\lambda)\mu + S_i(\lambda, I_N) \tag{1.5}$$

where the coefficients M_i, Q_i, R_i, and S_i are given by $Q_i \equiv \mathbf{J}_{ij}(I_N)\frac{d^2 f^j}{d\lambda^2} + \mathbf{C}_{ijk}(I_N)\frac{df^j}{d\lambda}\frac{df^k}{d\lambda}$, $M_i \equiv \mathbf{J}_{ij}(I_N)\frac{df^j}{d\lambda}$, $R_i \equiv \mathbf{R}_{ij}\frac{df^j}{d\lambda}$, and $S_i \equiv \mathbf{g}_i(I_N)$. The functional dependence of $\mathbf{J}_{ij}$, $\mathbf{C}_{ijk}$ and $\mathbf{g}_i$ on the payload I_N has been shown explicitly. Then the dynamics of the system after the payload has been perturbed may be written

$$\mathbf{u}'_i = M_i(\lambda, I_N + \Delta I_N)\dot{\mu} + Q_i(\lambda, I_N + \Delta I_N)\mu^2 + R_i(\lambda)\mu + S_i(\lambda, I_N + \Delta I_N) \tag{1.6}$$

In order to avoid excessive torque requirements, we wish to compute a set of velocities and accelerations μ and $\dot{\mu}$ such that if the nominal torques $\mathbf{u}_i$ are given by (1.5), then the actual torques $\mathbf{u}'_i$ given by (1.6) will be realizable, i.e.,

$$\mathbf{u}_i^{\min}(\lambda,\mu) \le \mathbf{u}'_i \le \mathbf{u}_i^{\max}(\lambda,\mu). \tag{1.7}$$

Formally, the *Robust Trajectory Planning* (**RTP**) problem may be stated as follows:

> Given a geometric path described as a parameterized curve, the torque limits $\mathbf{u}_i^{\min}$ and $\mathbf{u}_i^{\max}$ as functions of λ and μ, the dynamics of the robot when carrying the nominal payload I_N, and a bound E on the norm of the difference between the pseudo-inertias of the actual and nominal payloads, determine the fastest trajectory (sequence of (λ,μ) pairs) such that the torques $\mathbf{u}'_i$ given by (1.6) satisfy the constraints (1.7) for all points on the trajectory and for all payload errors ΔI_N such that $\|\Delta I_N\| \le E$.

We will solve this problem by calculating the worst-case torque error, as a function of λ, μ, and $\dot{\mu}$ for a given payload error, and decreasing the torque limits by this amount when doing trajectory planning.

The rest of the paper is organized as follows: in Section 2, torque errors are calculated in terms of changes to the payload pseudo-inertia. In Section 3 bounds on the joint torque errors are derived in terms of bounds on the norm of the pseudo-inertia error. Section 3 also discusses how these results can be incorporated into the trajectory planning process. Section 4 presents a numerical example, and the paper concludes with Section 5.

2. CALCULATION OF DYNAMIC COEFFICIENT ERRORS

For a given path, we need to know the changes to the coefficients M_i, Q_i, R_i, and S_i in Eq. (1.5) which result from changes in the dynamics of the robot. In the sequel, changes in dynamics will be assumed to come from changes in payload characteristics. While changes in friction coefficients, and hence changes to R_i, also contribute to changes in required torques, such changes are independent of changes in payload characteristics, and for the sake of simplicity will not be dealt with here.[2]

Determining the change to M_i, we have

$$M_i(I_N) = \mathbf{J}_{ij}(I_N)\frac{df^j}{d\lambda} \tag{2.1}$$

and

$$M_i(I_N + \Delta I_N) = \mathbf{J}_{ij}(I_N + \Delta I_N)\frac{df^j}{d\lambda} \tag{2.2}$$

so that

$$\begin{aligned} \delta M_i(I_N, \Delta I_N) &\equiv M_i(I_N + \Delta I_N) - M_i(I_N) \\ &= \left\{ \mathbf{J}_{ij}(I_N + \Delta I_N) - \mathbf{J}_{ij}(I_N) \right\}\frac{df^j}{d\lambda} \\ &\equiv \delta \mathbf{J}_{ij}(I_N, \Delta I_N)\frac{df^j}{d\lambda} \end{aligned} \tag{2.3}$$

Differences between nominal and actual payload characteristics cause changes in the coefficients $\mathbf{J}_{ij}$ $\mathbf{C}_{ijk}$, and $\mathbf{g}_i$ in equation (1.1). Here we determine the relationship between changes in these coefficients and changes in payload characteristics.

Changes in payload characteristics will result in changes to the pseudo-inertia tensor of the last joint of the robot, i.e., the pseudo-inertia tensor will have the value $I_N + \Delta I_N$ instead of I_N. In order to obtain δM_i, we consider how this affects the inertia matrix. The coefficients in the inertia matrix are given in [3] as

$$\mathbf{J}_{ij}(I_N) = \sum_{p=\max(i,j)}^{N} Tr\left\{ \frac{\partial \mathbf{T}_p}{\partial \mathbf{q}^j} I_p \frac{\partial \mathbf{T}_p^T}{\partial \mathbf{q}^i} \right\} \tag{2.4}$$

where $\mathbf{T}_p$ is the 4×4 homogeneous transformation matrix which transforms vectors given in the coordinate system associated with the p^{th} link of the robot to world or base coordinates, and I_p is the pseudo-inertia of the p^{th} link given in the p^{th} link's coordinate frame.

[2] Such changes are usually determined experimentally.

Introducing an error ΔI_N into the pseudo-inertia of the last joint gives

$$\mathbf{J}_{ij}(I_N + \Delta I_N) = \sum_{p=\max(i,j)}^{N-1} Tr\left\{ \frac{\partial \mathbf{T}_p}{\partial \mathbf{q}^j} I_p \frac{\partial \mathbf{T}_p^T}{\partial \mathbf{q}^i} \right\} + Tr\left\{ \frac{\partial \mathbf{T}_N}{\partial \mathbf{q}^j} (I_N + \Delta I_N) \frac{\partial \mathbf{T}_N^T}{\partial \mathbf{q}^i} \right\}. \quad (2.5)$$

Subtracting (2.4) from (2.5) gives

$$\delta \mathbf{J}_{ij}(I_N, \Delta I_N) = \delta \mathbf{J}_{ij}(\Delta I_N) = Tr\left\{ \frac{\partial \mathbf{T}_N}{\partial \mathbf{q}^j} \Delta I_N \frac{\partial \mathbf{T}_N^T}{\partial \mathbf{q}^i} \right\}. \quad (2.6)$$

Note that the error in the inertia matrix is linear in the pseudo-inertia error, and is *independent* of the nominal payload. To find δM_i, simply plug (2.6) into (2.3), giving

$$\delta M_i = \sum_j \frac{df^j}{d\lambda} Tr\left\{ \frac{\partial \mathbf{T}_N}{\partial \mathbf{q}^j} \Delta I_N \frac{\partial \mathbf{T}_N^T}{\partial \mathbf{q}^i} \right\}. \quad (2.7)$$

Computation of the errors δQ_i follows the same pattern as the computation of δM_i. The errors in the Coriolis terms can be determined in much the same way as the errors in the inertia matrix. From [3] we have

$$\mathbf{C}_{ijk} = \sum_{p=\max(i,j,k)}^{N} Tr\left\{ \frac{\partial^2 \mathbf{T}_p}{\partial \mathbf{q}^j \partial \mathbf{q}^k} I_p \frac{\partial \mathbf{T}_p^T}{\partial \mathbf{q}^i} \right\}. \quad (2.8)$$

The errors in the Coriolis terms due to errors in payload characteristics are therefore given by

$$\delta \mathbf{C}_{ijk} = Tr\left\{ \frac{\partial^2 \mathbf{T}_N}{\partial \mathbf{q}^j \partial \mathbf{q}^k} \Delta I_N \frac{\partial \mathbf{T}_N^T}{\partial \mathbf{q}^i} \right\}. \quad (2.9)$$

The definition of Q_i gives

$$\delta Q_i = \sum_j Tr\left\{ \frac{\partial \mathbf{T}_N}{\partial \mathbf{q}^j} \Delta I_N \frac{\partial \mathbf{T}_N^T}{\partial \mathbf{q}^i} \right\} \frac{d^2 f^j}{d\lambda^2} + \sum_j \sum_k Tr\left\{ \frac{\partial^2 \mathbf{T}_N}{\partial \mathbf{q}^j \partial \mathbf{q}^k} \Delta I_N \frac{\partial \mathbf{T}_N^T}{\partial \mathbf{q}^i} \right\} \frac{df^j}{d\lambda} \frac{df^k}{d\lambda}. \quad (2.10)$$

Now we need to know the error in the gravitational terms. The gravitational forces $\mathbf{g}_i$ are given by [3]

$$\mathbf{g}_i = \sum_{k=i}^{N} -m_k \mathbf{G}^T \frac{\partial \mathbf{T}_k}{\partial \mathbf{q}^i} \bar{\mathbf{r}}_k \quad (2.11)$$

where $\mathbf{G} = [0\ 0\ g\ 0]^T$ is the gravitational force vector, m_k is the mass of the k^{th} link, g is the acceleration due to gravity, and $\bar{\mathbf{r}}_k = [\bar{x}\ \bar{y}\ \bar{z}\ 1]$ is the center of mass of the k^{th} link given in the coordinates of the k^{th} frame. If we define $\mathbf{w}_k \equiv m_k \bar{\mathbf{r}}_k$, then we have

$$\mathbf{g}_i = -\sum_{k=i}^{N} \mathbf{G}^T \frac{\partial \mathbf{T}_k}{\partial \mathbf{q}^i} \mathbf{w}_k \quad (2.12)$$

But $\mathbf{w}_k$ is just the last column of the pseudo-inertia matrix I_k, so that

$$\mathbf{g}_i = -\sum_{k=i}^{N} \mathbf{G}^T \frac{\partial \mathbf{T}_k}{\partial \mathbf{q}^i} I_k \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad (2.13)$$

As before, introducing an error into the pseudo-inertia of the last link gives

$$\delta \mathbf{S}_i = \delta \mathbf{g}_i = -\mathbf{G}^T \frac{\partial \mathbf{T}_N}{\partial \mathbf{q}^i} \Delta I_N \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}. \quad (2.14)$$

We may now calculate the error in $\mathbf{u}_i$ by adding up the individual components, giving

$$\delta \mathbf{u}_i = \delta M_i \dot{\mu} + \delta Q_i \mu^2 + \delta S_i. \quad (2.15)$$

3. CALCULATION OF TORQUE ERROR BOUNDS

If the errors ΔI_N were known exactly, then the torque errors could also be computed exactly. Of course, in practice ΔI_N will not be known exactly. However, if bounds on the norm of ΔI_N can be obtained then we may find bounds on $\delta \mathbf{u}_i$.

To obtain these bounds, note that δM_i, δQ_i, and δS_i are all functions of the pseudo-inertia error ΔI_N; in fact, they are *linear* in ΔI_N, so that $\delta \mathbf{u}_i$ is also linear in ΔI_N. If we write

$$\delta \mathbf{u}_i = Z(\Delta I_N), \quad (3.1)$$

then we wish to maximize or minimize the linear function Z with respect to ΔI_N, subject to $\|\Delta I_N\| \leq E$, where E is the bound on the pseudo-inertia error.

At this point, some observations are in order. First as we noted before, the errors in the $\delta \mathbf{u}_i$ depend linearly upon the pseudo-inertia error ΔI_N. Second, note also that $\delta \mathbf{u}_i$ depends *only* on the kinematics of the robot and on the desired velocity and acceleration, not on the nominal dynamics. These facts are consequences of the fact that both kinetic and potential energy are linear in mass, as can be seen from the derivation of thee Lagrangian form of the dynamic equations. One consequence of the linearity of the linearity of the dynamic equations is that forces caused by the nominal robot dynamics and those caused by errors in the dynamics can be separated, as shown above. The implication of this is that much of the error analysis can proceed without regard to the nominal dynamics of the robot.

The linearity of the $\delta \mathbf{u}_i$ in the pseudo-inertia has some other practical consequences as well. Consider the maximization which must be performed in order to evaluate $\delta \mathbf{u}_i$. This maximization requires that the space of 4×4 symmetric matrices with norm less than E be searched, which in general is a rather formidable problem. However, by choosing a particular class of matrix norms, the problem can be made quite simple in fact, it can be transformed into a linear programming problem.

To see how this transformation can be performed consider the problem of maximizing the function Z in equation (3.1), namely

Problem A: maximize $Z(\mathbf{M}) = \sum_i \sum_j \beta_{ij} \mathbf{M}_{ij}$ (3.2)

subject to $\|\mathbf{M}\| \leq E$ and $\mathbf{M} = \mathbf{M}^T$ (3.3)

Treatment of the minimization problem proceeds analogously to problem A. We will show that problem A transforms into a linear programming problem if the norm used to constrain the matrix $\mathbf{M}$ in (3.3) is chosen properly. This will be accomplished by eliminating some absolute values from the constraints.

$Z(\mathbf{M})$, the function to be maximized, is a linear function of $\mathbf{M}$. It remains to be shown that the constraints can be made linear. Of course if the norm used

in problem A is arbitrary, then in general the constraints will not be linear. However, there is a set of norms, all very easy to calculate, which will yield linear constraints. Consider the class of functions F: $\mathbf{R}^{4\times 4} \rightarrow \mathbf{R}^{+}$ given by $F(\mathbf{M}) = \max_i \sigma_i(\mathbf{M})$, where

$$\sigma_i(\mathbf{M}) = \sum_{j=1}^{4}\sum_{k=1}^{4} \alpha_{ijk} |\mathbf{M}_{jk}|.$$

The matrix 1-norm and ∞-norm, $\max_{i,j} |\mathbf{M}_{ij}|$ and $\sum_{i,j} |\mathbf{M}_{ij}|$ are all functions of this form. It is easily shown that if $\alpha_{ijk} \geq 0$ for all i, j and k, and if for every pair of indices (j,k) there is an i such that $\alpha_{ijk} \neq 0$, then $F(\mathbf{M})$ is a norm. Problem A with this class of norms becomes

$$\text{Problem B: maximize } Z(\mathbf{M}) = \sum_i \sum_j \beta_{ij} \mathbf{M}_{ij} \tag{3.4}$$

$$\text{subject to } \max_i \left\{ \sum_{j=1}^{4}\sum_{k=1}^{4} \alpha_{ijk} |\mathbf{M}_{jk}| \right\} \leq E$$

$$\text{and } \mathbf{M}_{jk} = \mathbf{M}_{kj}.$$

This problem obviously is equivalent to

$$\text{Problem C: maximize } Z(\mathbf{M}) = \sum_i \sum_j \beta_{ij} \mathbf{M}_{ij} \tag{3.5}$$

$$\text{subject to } \sum_{j=1}^{4}\sum_{k=1}^{4} \alpha_{ijk} |\mathbf{M}_{jk}| \leq E$$

$$\text{and } \mathbf{M}_{jk} = \mathbf{M}_{kj}.$$

Problem C may be transformed into a standard linear programming problem by making the substitutions $\mathbf{M}_{ij} = P_{ij} - N_{ij}$ and $|\mathbf{M}_{ij}| = P_{ij} + N_{ij}$, where P_{ij} and N_{ij} are non-negative real numbers, and eliminating the symmetry constraint on $\mathbf{M}$. Thus the problem reduces to

$$\text{maximize } Z(\mathbf{P},\mathbf{N}) = \sum_{j=1}^{4}\sum_{k=j}^{4} \beta'_{jk}(\mathbf{P}_{jk} - \mathbf{N}_{jk})$$

$$\text{subject to } \sum_{j=1}^{4}\sum_{k=j}^{4} \alpha'_{ijk}(\mathbf{P}_{jk} + \mathbf{N}_{jk}) \leq E$$

$$\text{and } P_{ij} \geq 0,\ N_{ij} \geq 0$$

where

$$\alpha'_{ijk} = \begin{cases} \alpha_{ijk} & j = k \\ \alpha_{ijk} + \alpha_{ikj} & j \neq k \end{cases}$$

and

$$\beta'_{ij} = \begin{cases} \beta_{ij} & i = j \\ \beta_{ij} + \beta_{ji} & i \neq j \end{cases}$$

Now that torque error bounds can be obtained from pseudo-inertia errors, these results must be incorporated into the trajectory planning process. Direct use of the results derived above in the trajectory planner described in [4] is not easy, since this trajectory planner requires that we solve (1.5) for μ in terms of λ, μ, and $\mathbf{u}_i$. This solution is required because the trajectory planner must convert torque ranges into μ ranges. When errors are introduced, we have

$$\begin{aligned} \mathbf{u}_i^{\min}(\lambda,\mu) &\leq M_i\dot{\mu} + Q_i\mu^2 + R_i\mu + S_i \\ &\quad + \min_{\|\Delta I_N\| \leq E} \left\{ \delta M_i\dot{\mu} + \delta Q_i\mu^2 + \delta S_i \right\} \\ &\leq M_i\dot{\mu} + Q_i\mu^2 + R_i\mu + S_i \\ &\quad + \max_{\|\Delta I_N\| \leq E} \left\{ \delta M_i\dot{\mu} + \delta Q_i\mu^2 + \delta S_i \right\} \\ &\leq \mathbf{u}_i^{\max}(\lambda,\mu). \end{aligned}$$

For given λ and μ, these inequalities determine a range of values of $\dot{\mu}$. However, the worst-case values of δM_i, δQ_i, and δS_i depend upon $\dot{\mu}$, which in turn depends upon the values of δM_i, δQ_i, and δS_i, so finding the allowable range of values of $\dot{\mu}$ explicitly is difficult. Of course these equations can be solved numerically. For example, to find the maximum allowable value of $\dot{\mu}$, the equation

$$M_i\dot{\mu} + Q_i\mu^2 + R_i\mu + S_i + \max_{\|\Delta I_N\| \leq E} \left\{ \delta M_i\dot{\mu} + \delta Q_i\mu^2 + \delta S_i \right\} = \mathbf{u}_i^{\max}(\lambda,\mu)$$

can be solved by bisection for $\dot{\mu}$.

A simpler solution to the problem is to use the trajectory planner described in [6]. This trajectory planner only needs to have available a test function which determines whether or not a given $(\lambda,\mu,\dot{\mu})$ triple requires excessive torque; in effect, it automatically performs the numerical search for the allowable values of $\dot{\mu}$. But such a function is easily constructed, since for a given $(\lambda,\mu,\dot{\mu})$ we can easily minimize or maximize $\delta\mathbf{u}_i$ in (2.15), and see if $\mathbf{u}_i + \delta\mathbf{u}_i^{\max}$ exceeds $\mathbf{u}_i^{\max}(\lambda,\mu)$ or $\mathbf{u}_i + \delta\mathbf{u}_i^{\min}$ falls below $\mathbf{u}_i^{\min}(\lambda,\mu)$. In particular, the following algorithm checks to see if a particular $(\lambda,\mu,\dot{\mu})$ triple meets all the torque constraints:

for each joint i **do**

begin

compute $\mathbf{u}_i^N = M_i(\lambda)\dot{\mu} + Q_i(\lambda)\mu^2 + R_i(\lambda)\mu + S_i(\lambda)$;

compute $\delta\mathbf{u}_i^{\max} = \max_{\|\Delta I_N\| \leq E} \left\{ \delta M_i(\lambda,\Delta I_N)\dot{\mu} + \delta Q_i(\lambda,\Delta I_N)\mu^2 + \delta S_i(\lambda,\Delta I_N) \right\}$;

compute $\delta\mathbf{u}_i^{\min} = \min_{\|\Delta I_N\| \leq E} \left\{ \delta M_i(\lambda,\Delta I_N)\dot{\mu} + \delta Q_i(\lambda,\Delta I_N)\mu^2 + \delta S_i(\lambda,\Delta I_N) \right\}$;

if $\mathbf{u}_i^N + \delta\mathbf{u}_i^{\min} < \mathbf{u}_i^{\min}(\lambda,\mu)$, then return REJECT;

if $\mathbf{u}_i^N + \delta\mathbf{u}_i^{\max} > \mathbf{u}_i^{\max}(\lambda,\mu)$, then return REJECT;

end;

return ACCEPT

It should be noted that this function is called for *each* (λ,μ) pair; it does not, for example, reject a $(\lambda,\mu,\dot{\mu})$ triple based on an error which is computed for *all* positions or *all* velocities. As a consequence, speed is sacrificed only when absolutely necessary to guarantee that the trajectory will be realizable for all payloads within the allowable range.

4. NUMERICAL EXAMPLE

As an example, we will apply the methods of the previous section to the first three joints of the Bendix PACS robot arm. This arm is cylindrical in configuration and is driven by DC servos. Its dynamics and actuator characteristics, as well as the trajectory planner used, are described in [6].

The coordinate transform $\mathbf{T}_3$ for the PACS arm is

$$\mathbf{T}_3 = \begin{bmatrix} \cos\vartheta & 0 & -\sin\vartheta & -r\sin\vartheta \\ \sin\vartheta & 0 & \cos\vartheta & r\cos\vartheta \\ 0 & -1 & 0 & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.1}$$

The partial derivatives of $\mathbf{T}_3$ are easily computed, and the resulting dynamic coeeficient errors

$$\delta M_\vartheta = (H_{11} + H_{33} + 2rH_{34} + r^2H_{44})\frac{d\vartheta}{d\lambda} + H_{14}\frac{dr}{d\lambda}$$

$$\delta M_z = H_{44}\frac{dz}{d\lambda};\quad \delta M_r = H_{14}\frac{d\vartheta}{d\lambda} + H_{44}\frac{dr}{d\lambda};\quad \delta Q_z = H_{44}\frac{d^2z}{d\lambda^2}$$

$$\delta Q_\vartheta = (H_{11} + H_{33} + 2rH_{34} + r^2H_{44})\frac{d^2\vartheta}{d\lambda^2} + H_{14}\frac{d^2r}{d\lambda^2} + 2(H_{34} + rH_{44})\frac{dr}{d\lambda}\frac{d\vartheta}{d\lambda}$$

$$\delta Q_r = H_{14}\frac{d^2\vartheta}{d\lambda^2} + H_{44}\frac{d^2r}{d\lambda^2} - (H_{34} + rH_{44})\left(\frac{d\vartheta}{d\lambda}\right)^2$$

$$\delta S_z = H_{44}g;\quad \delta S_\vartheta = \delta S_r = 0$$

where g is the acceleration due to gravity and where the matrix $\mathbf{H} \equiv \Delta I_n$.

We will use the norm

$$\|\mathbf{H}\| = \sum_{i=1}^{4}\sum_{j=1}^{4} \alpha_{ij}\,|\mathbf{H}_{ij}|$$

where $\alpha_{ij} > 0$. This makes the problem of finding the error bounds very simple. It is easily seen that if the functional to be maximized is $Z = \sum_i\sum_j \beta_{ij}\mathbf{H}_{ij}$, then the maximum over $\mathbf{H}$ for $\|\mathbf{H}\| \le E$ occurs when all the $\mathbf{H}_{ij}$ are zero except for those $\mathbf{H}_{ij}$ for which $|\beta_{ij}/\alpha_{ij}|$ is a maximum; this number times E is also the maximum value of Z. If we use

$$\alpha_{ij} = \begin{cases} 1 & i = j \\ 1/2 & i \ne j \end{cases}$$

then the resulting bounds on the $|\delta\mathbf{u}_i|$ are

$$|\delta\mathbf{u}_z| \le \left|\frac{dz}{d\lambda}\dot{\mu} + \frac{d^2z}{d\lambda^2}\mu^2 + g\right| \|\Delta I_N\|$$

$$|\delta\mathbf{u}_\vartheta| \le \max\left\{\left|\frac{d\vartheta}{d\lambda}\dot{\mu} + \frac{d^2\vartheta}{d\lambda^2}\mu^2\right|, \left|\left(\frac{d\vartheta}{d\lambda}\right)^2\mu^2\right|, \left|2r\frac{d\vartheta}{d\lambda}\dot{\mu} + 2r\frac{d^2\vartheta}{d\lambda^2}\mu^2 + 2\frac{dr}{d\lambda}\frac{d\vartheta}{d\lambda}\mu^2\right|\right\}\|\Delta I_N\|$$

$$|\delta\mathbf{u}_r| \le \max\left\{\left|\frac{dr}{d\lambda}\dot{\mu} + \frac{d^2r}{d\lambda^2}\mu^2\right|, \left|r\left(\frac{d\vartheta}{d\lambda}\right)^2\mu^2\right|, \left|r^2\frac{d\vartheta}{d\lambda}\dot{\mu} + r^2\frac{d^2\vartheta}{d\lambda^2}\mu^2 + 2r\frac{dr}{d\lambda}\frac{d\vartheta}{d\lambda}\mu^2\right|\right\}\|\Delta I_N\|$$

$$|\delta\mathbf{u}_r| \le \max\left\{\left|\frac{d\vartheta}{d\lambda}\dot{\mu} + \frac{d^2\vartheta}{d\lambda^2}\mu^2\right|, \left|\left(\frac{d\vartheta}{d\lambda}\right)^2\mu^2\right|, \left|\frac{dr}{d\lambda}\dot{\mu} + \frac{d^2r}{d\lambda^2}\mu^2 - r\left(\frac{d\vartheta}{d\lambda}\right)^2\mu^2\right|\right\}\|\Delta I_N\|.$$

The joint torques that can be applied to the PACS arm are limited by saturation of the drive motors which gives a constant torque or force limit for each joint. In addition, there are limits on the voltages which can be applied to the motors; so we need to know how the errors in the joint torques translate into errors in the motor voltages. It will be assumed that the back-EMF constant, winding resistance, and voltage source resistance are known exactly, though this is not necessary. Since for a given speed voltage is a linear function of torque, i.e., $V_i = A_i\mathbf{u}_i + B_i$, the change in voltage will be $\delta V_i = A_i\delta\mathbf{u}_i$. These changes in voltage can then be added to the nominal voltage and tested against the motor voltage limits in much the same way that the torques are checked against the motor torque saturation limits.

The perturbation to the nominal dynamics of the manipulator will be caused by placing a cube with edges of length L and uniform mass density ρ in the gripper of the robot, with its center of mass coincident with the origin of the end effector coordinate system. The pseudo-inertia of this cube is

$$\Delta I_3 = \begin{bmatrix} \frac{1}{12}\rho L^5 & 0 & 0 & 0 \\ 0 & \frac{1}{12}\rho L^5 & 0 & 0 \\ 0 & 0 & \frac{1}{12}\rho L^5 & 0 \\ 0 & 0 & 0 & \rho L^3 \end{bmatrix}$$

The norm of this "error" is $\|\Delta I_3\| = \rho L^3 + \frac{1}{4}\rho L^5$. The maximum torque error for a given range of pseudo-inertia errors occurs when the error bound E is precisely equal to the norm of the actual pseudo-inertia error. Therefore the most stringent test of the results of the previous section is to use a tight error bound i.e., $E = \|\Delta I_3\|$. This has been done for a cube with sides of 5 centimeters and densities of 0, 6, 12, 18, 24, and 30 grams/cc. The path traversed is a straight line from the (Cartesian) point (0.7,0.7,0.1) to (0.4,-0.4,0.4). For comparison, the true optimal solution has been calculated, using the actual dynamics (including the effects of the cube in the gripper). The results are summarized in Tables 1 through 5. Table 1 gives traversal times for the true optimal solution and for the case in which errors are included. The "percent difference" column gives the percentage by which the true optimal traversal time is exceeded. Tables 2 and 3 give minimum and maximum voltages, respectively. The actual and nominal values are both computed for the "nominal" trajectory, i.e. the trajectory which is calculated with errors included. The actual voltages are those required to move the robot with the cube in the gripper, while the nominal values are those which are required without the cube (i.e. with the nominal payload.) The minimum and maximum voltages available are -40 and 40 volts

and it is easily seen that these limits are not exceeded for any joint or for either payload. Tables 4 and 5 give the minimum and maximum torques or forces for each joint. The torque or force limits are given at the head of the column for the appropriate joint; again, the limits are not exceeded.

The phase plane (λ vs. μ) plot and motor voltage vs. time plot for the zero-density case are shown in Figures 1a and 1b. Since the error is zero in this case, the results are exact. For a density of 12 grams/cc., the optimal and nominal (i.e. with errors included) phase plane plots are shown in Figure 2a. Figure 2b gives joint positions vs. time; z and r are in meters, θ in radians. Figures 2c through 2e give nominal and actual motor voltages required to drive the robot along the nominal trajectory for the z, θ and r joints respectively. (The nominal voltages are those which would be required if the actual payload were identical to the nominal payload. The actual torques are the torques required to keep the robot with the perturbed payload on the nominal trajectory.) Figures 3a through 3e show the same plots for a density of 24 grams/cc.

It was noted above that none of the joint torque or voltage constraints was violated. However, the minimum voltage for the r joint at one point meets the lower voltage limit. This indicates that the trajectory which is generated when payload errors are included is indeed the fastest possible trajectory for the given range of possible payloads; for this particular point, the worst-case payload happens to have the same characteristics as the actual payload. A larger payload would have resulted in violation of a voltage constraint.

Another point to consider is the relationship between the nominal and optimal phase trajectories. It is expected that the nominal phase trajectory will be lower than the optimal trajectory; a nominal trajectory which was higher than the optimal one would lead to a contradiction of the optimality of the optimal trajectory. Also, the difference between the optimal and nominal trajectories increases as the payload error bound increases. This would be expected, since the nominal trajectory must accomodate *all* payloads within a given range; as the range of payloads increases, the worst-case errors also increase, resulting in more restrictive limits on the nominal torques, and hence slower trajectory traversal times.

5. CONCLUSIONS

A method for including payload inertia errors in the manipulator trajectory planning process has been presented. Errors in the payload inertia are characterized by bounds on the norm of the difference between the actual and nominal pseudo-inertias of the payload. Given such a bound, it has been shown that a trajectory can be constructed which meets all torque and force constraints for *all* actual payloads, provided that the norm of the difference of the pseudo-inertias of the actual and nominal payloads differs by less than the given error bound. This technique was applied to the Bendix PACS robot for a number of different payloads, and the resulting trajectories were shown not to violate any joint torque or motor voltage constraints. In the worst case, in which the actual payload mass differs from the nominal mass by approximately one third of the robot's rated maximum load, the traversal time was less than twenty percent over the optimal value.

Density	Time (Seconds)		percent difference
	Nominal	Optimal	
0	1.789	1.789	0
6	1.934	1.844	4.9%
12	2.076	1.898	9.4%
18	2.213	1.950	13.5%
24	2.340	2.002	16.9%
30	2.459	2.054	19.7%

Table 1. Traversal times for nominal and optimal trajectories

REFERENCES

[1] **J. E. Bobrow, S. Dubowsky, and J. S. Gibson, "On the Optimal Control of Robotic Manipulators with Actuator Constraints,"** *Proceedings of the 1983 Automatic Control Conference*, pp. 782-787, June 1983.

[2] **A. J. Koivo and T. H. Guo, "Adaptive Linear Controller for Robotic Manipulators,"** *IEEE Transactions on Automatic Control*, vol. AC-28, no. 2, pp 162-170, February 1983.

[3] **R. P. C. Paul,** *Robot manipulators: Mathematics programming, and control.* Cambridge, Mass. MIT Press, 1981.

[4] **K. G. Shin and N. D. McKay, "Minimum-Time Control of a Robotic Manipulator with Geometric Path Constraints,"** *IEEE Transactions on Automatic Control*, vol. AC-30, no. 6, pp. 531-541, June 1985.

[5] **K. G. Shin and N. D. McKay, "Robot Path Planning Using Dynamic Programming,"** *Proceedings of the 23rd CDC*, pp. 1629-1635, Dec. 1984 (also to appear in *IEEE Trans. on Automatic Control*, vol. AC-31 no. 6, June 1986)..

[6] **K. G. Shin and N. D. McKay, "Minimum Time Trajectory Planning with General Torque Constraints,"** *Proc. 1986 IEEE Conf. on Robotics and Automation.*

Minimum Voltages						
Density	z joint		θ joint		r joint	
	Nominal	Actual	Nominal	Actual	Nominal	Actual
0	29.86	29.86	-39.93	-39.93	-40.00	-40.00
6	30.34	30.91	-38.26	-38.52	-37.94	-40.00
12	30.54	31.67	-33.05	-33.51	-36.12	-39.98
18	30.67	32.39	-24.92	-25.37	-34.47	-39.99
24	30.78	33.07	-19.94	-20.36	-33.01	-40.00
30	30.86	33.73	-16.80	-17.18	-31.70	-39.98

Table 2. Minimum required voltages for nominal and actual payloads.

Maximum Voltages						
Density	z joint		θ joint		r joint	
	Nominal	Actual	Nominal	Actual	Nominal	Actual
0	37.64	37.64	39.86	39.86	40.00	40.00
6	37.40	38.03	37.14	37.73	32.62	32.60
12	36.79	38.04	17.64	18.25	27.01	27.48
18	35.78	37.61	10.61	11.21	23.79	24.94
24	35.12	37.54	7.30	7.89	21.39	23.41
30	34.69	37.69	5.39	5.98	19.51	22.39

Table 3. Maximum required voltages for nominal and actual payloads.

Minimum Torques/forces						
Density	z joint (Newtons)		θ joint (Newton-Meters)		r joint (Newtons)	
	Limit = -629 Nt.		Limit = -170 Nt.-M.		Limit = -15.7 Nt.	
	Nominal	Actual	Nominal	Actual	Nominal	Actual
0	333.52	333.52	-112.29	-112.29	-9.99	-9.99
6	335.91	342.20	-104.78	-105.71	-9.47	-9.99
12	363.02	376.62	-87.70	-89.26	-9.02	-9.99
18	373.03	394.00	-62.91	-64.43	-8.61	-9.99
24	377.84	406.17	-48.15	-49.55	-8.24	-9.99
30	380.11	415.72	-39.14	-40.43	-7.92	-9.98

Table 4. Minimum required torques/forces for nominal and actual payloads.

Maximum Torques/forces						
Density	z joint (Newtons)		θ joint (Newton-Meters)		r joint (Newtons)	
	Limit = 629 Nt.		Limit = 170 Nt.-M.		Limit = 15.7 Nt.	
	Nominal	Actual	Nominal	Actual	Nominal	Actual
0	421.31	421.31	161.72	161.72	10.03	10.03
6	418.68	426.52	150.39	152.36	8.15	8.18
12	414.50	430.04	78.43	80.48	6.76	6.89
18	406.68	429.55	51.39	53.41	5.96	6.25
24	402.17	432.31	38.18	40.19	5.36	5.87
30	400.42	437.96	30.20	32.20	4.89	5.61

Table 5. Maximum required torques/forces for nominal and actual payloads.

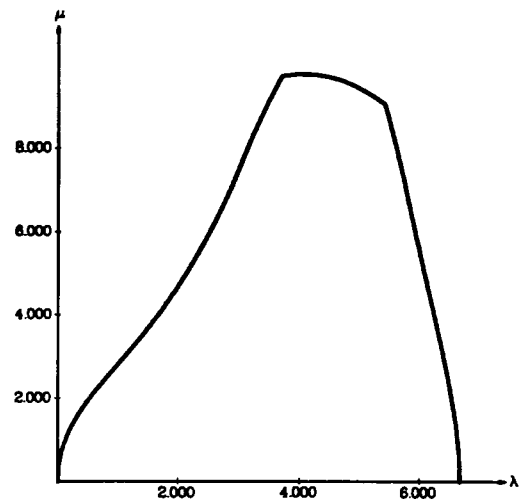

Figure 1a. Phase plane plot for zero errors.

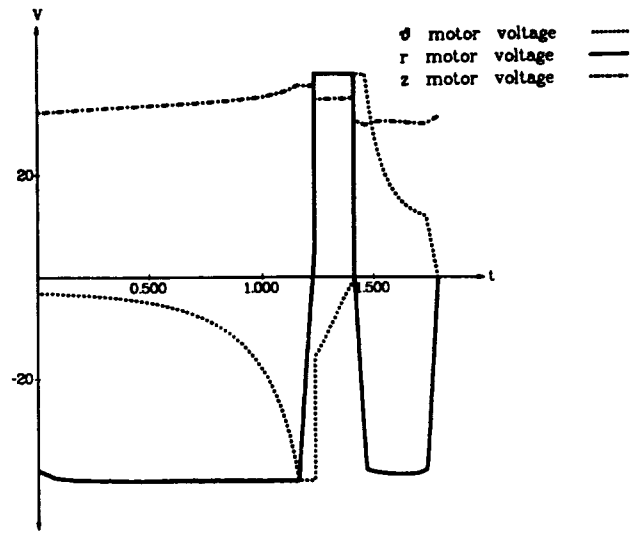

Figure 1b. Voltage vs. time for zero errors.

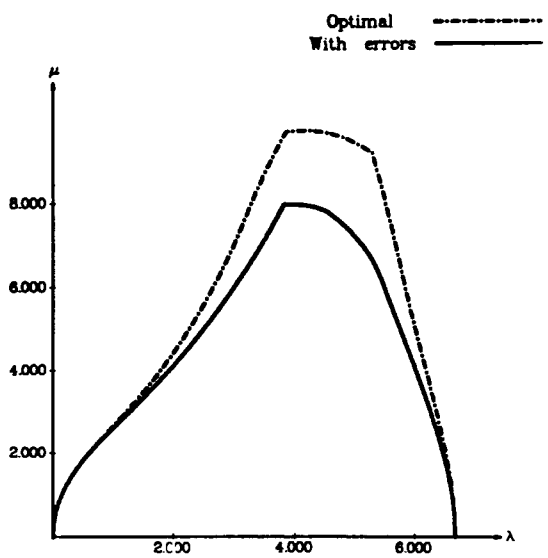

Figure 2a. Phase plane plots for density 12.0 g./cc.

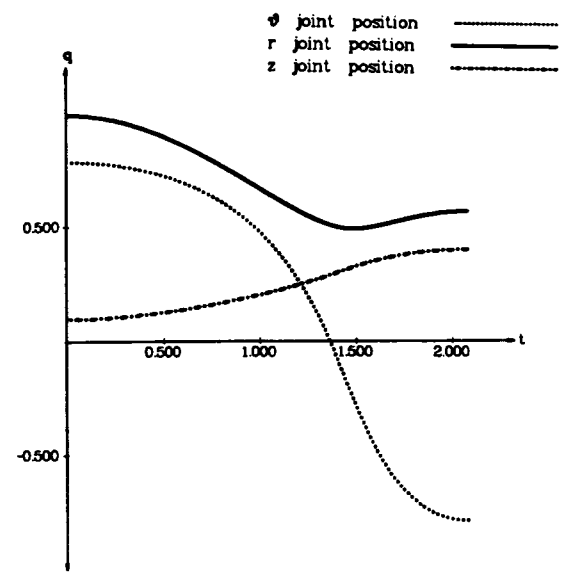

Figure 2b. Joint position vs. time for density 12.0 g./cc.

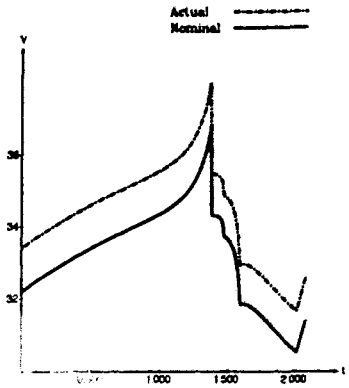

Figure 2c.
Nominal and actual motor voltages for z joint, density 12.0 g./cc.

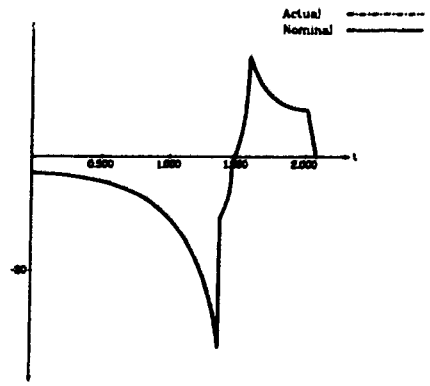

Figure 2d.
Nominal and actual motor voltages for θ joint, density 12.0 g./cc.

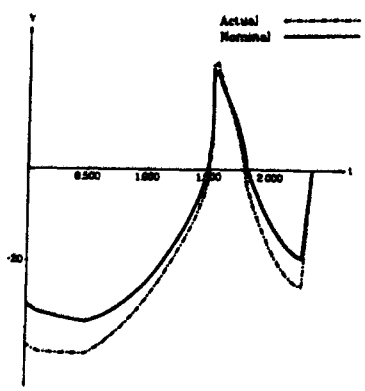

Figure 2e.
Nominal and actual motor voltages for r joint, density 12.0 g./cc.

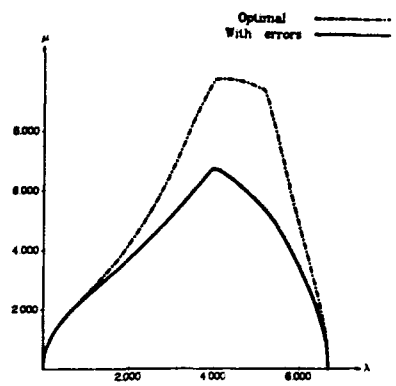

Figure 3a. Phase plane plots for density 24.0 g./cc.

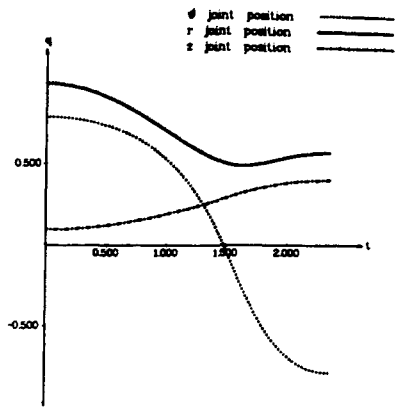

Figure 3b. Joint position vs. time for density 24.0 g./cc.

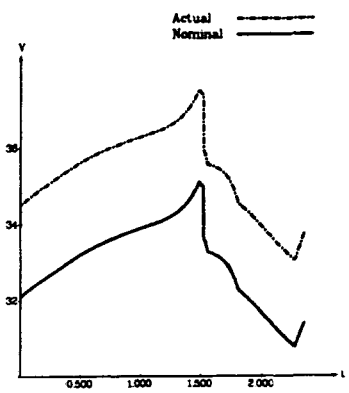

Figure 3c.
Nominal and actual motor voltages for z joint, density 24.0 g./cc.

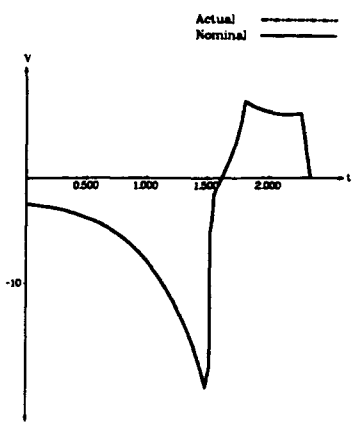

Figure 3d
Nominal and actual motor voltages for θ joint, density 24.0 g./cc.

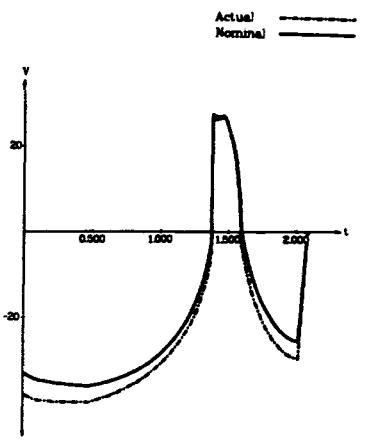

Figure 3e
Nominal and actual motor voltages for r joint, density 24.0 g./cc.