# MINIMUM-TIME TRAJECTORY CONTROL OF A TWO-LINK FLEXIBLE ROBOTIC MANIPULATOR\*

D.A. Schoenwald<sup>†</sup>, J.T. Feddema<sup>‡</sup>, G.R. Eisler, and D.J. Segalman Sandia National Laboratories Albuquerque, NM 87185

**Received by OST** 

OCT 2 4 1390

#### Abstract

This paper analyzes the experimental and simulation results of a minimum-time trajectory control scheme for a two-link flexible robot. An off-line optimization routine determines a minimum-time, straight line tip trajectory which stays within the torque constraints of the motors and ends in a quiescent state, i.e., no vibrational transients. An efficient finite-element model is used in the optimization to approximate the flexible arm dynamics. The control strategy described here is used to determine the feedback gains for the position, velocity, and strain gage signals from a quadratic cost criterion based on the finite-element model linearized about the straight line tip trajectory. These feedback signals are added to the modeled torque values obtained from the optimization routine and used to control the robot arm actuators. The results indicate that this combination of model-based and error-driven control strategies achieves a closer tracking of the desired trajectory and a better handling of modeling errors (such as tip payloads) than either strategy alone.

1



DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

<sup>\*</sup>Work supported by the U.S. Department of Energy at Sandia National Laboratories under Contract DE-AC0476DP00789.

<sup>&</sup>lt;sup>†</sup>Graduate Research Associate at The Ohio State University, Columbus, OH 43210. Worked the summer 1990 at Sandia National Laboratories.

<sup>&</sup>lt;sup>‡</sup>Please send all correspondence to J.T. Feddema, Division 1411, Sandia National Laboratories, Albuquerque, NM 87185.

# 1 Introduction

The issue of control of flexible structures is important for many applications involving highspeed, accurate motion of long, slender structural members. As robotic manipulators are applied to weight sensitive applications such as high-speed manipulation, space exploration, and mobile vehicles, many robotic systems will be designed with lighter weight and more flexible materials than present day robots. Future robotic arms will also be expected to accurately position larger payloads, to perform their motion at higher bandwidths, and to exert larger external forces on their surroundings. All of these conditions will increase the structural bending of the links.

This additional flexibility greatly complicates the task of controlling these structures due to the complex dynamics involved. For both rigid and flexible robots, model-based control can greatly improve the speed and transient response of a robotic system. Feedforward and computed torque control have been used to improve trajectory tracking of rigid robots [1]. Open-loop control based on off-line modal analysis [2] and input preshaping [3] have been used to accurately position single-link flexible structures. Additional work has been done on applying singular perturbation approaches [4] to the control of multi-link flexible manipulators.

In these model-based approaches, the key to successful control is in the accuracy of the analytical model. System identification of a flexible link robot is even more difficult than the non-trivial problem of determining the correct model parameters of a rigid robot [5]. Whereas a rigid robot's dynamics are defined by a set of second order ordinary differential equations, a flexible robot's dynamics are defined by a set of fourth order partial differential equations with corresponding boundary conditions [6]. A flexible structure contains state variables not only in the form of joint angle positions and velocities but also in the form of link curvatures and their derivatives. Additional sensors, such as strain gages and accelerometers, have been added to single link arms to control additional flexible states [7][8].

The objective of this research is the minimum-time control of the tip position of a robot arm with two planar flexible degrees-of-freedom along a straight line, subject to motor torque limits and a quiescent end point state. Among the model-based control strategies considered is an open-loop controller with torque trajectories computed off-line by an optimizing program [9][11]. The off-line optimization is used to calculate the motor torques necessary to move the tip of the arm along a minimum-time, straight line trajectory while staying within the physical limits of the motors. A finite-element model of the two-link arm was used to simulate the arm's forward and inverse dynamics [10]. The performance of this method depends heavily on the accuracy of the dynamic model of the structure.

 $\mathbf{2}$ 

To improve the performance of this open loop control, a sensor-based control scheme has been developed and will be described in this paper. This closed loop strategy involves feeding back to the motor the errors in position, velocity, and curvature, i.e., the difference between these actual quantities as measured by on-line sensors and the desired values as dictated by the optimization routine. In previous work [11], the strategy for obtaining the feedback gains was basically through experimentation. Although the results obtained were quite satisfactory, this method is not easily duplicated on other structures and trajectories. These concerns motivated the approach described in this paper in which an optimal control strategy is used to determine the feedback gains.

The procedure to determine these gains starts by linearizing the finite-element model about the desired trajectory. This is done by choosing points along the trajectory and obtaining a linear time-invariant model about each of these points through a Taylor series truncation. The quadratic cost criterion is chosen based on physical knowledge of the robotic system, and the appropriate Riccati equation is computed to obtain the set of feedback gains. In this case, the states represent errors about the nominal trajectory in position, velocity, and curvature. Thus, minimizing the cost criteria will result in optimal tracking of the desired trajectory. The set of feedback gains obtained can be scheduled according to the part of the trajectory the tip is currently traversing. This control strategy is particularly useful for manipulators performing repetitive maneuvers. A similar strategy was implemented for a two-link flexible manipulator traversing a circular trajectory [15]. Only computer simulations were undertaken with no experimental results, and there was no minimum-time objective for the noneuver. However, the results do include feedforward torque values obtained from the inverse dynamics, and the combination of LQR feedback with the feedforward torque values yields much better results than either strategy alone.

This paper is organized as follows. Section 2 briefly describes the finite-element model and the optimization routine used to obtain the modeled torque, joint angle, joint velocity, and curvature values. Previous work on a feedforward method of control [11] is discussed in Section 3. In Section 4, the methodology by which the feedback gains were obtained is detailed including an explanation of how the cost criterion was chosen. Section 5 describes the results of a computer simulation study used to narrow down the set of gains to attempt on the actual system. These results also brought forth some details to consider in actual implementation such as observation and modeling errors. Finally, Section 6 presents the experimental results obtained on the two-link flexible robot at Sandia National Laboratories.

#### 2 Structural Model and Optimizing Program

The dynamics of flexible arms can be derived via Hamilton's principle. The resulting nonlinear integro-partial differential equations are unwieldy to deal with directly. An assumed modes method in which the flexure variables are expanded as a series of mode shapes can be used to reduce the dynamics to a set of nonlinear ordinary differential equations [12][13]. In this paper, a finite-element model [9] is used to obtain a set of nonlinear ordinary differential equations.

In the finite-element model of the two-link flexible arm, the structure was divided into 11 elements. Figure 1(a) shows the Sandia two-link manipulator, and Figure 1(b) illustrates the geometry of the structure. The first element corresponds to the 1st joint or hub of the manipulator. The second element corresponds to the first bracket. The next three elements represent the 1st link divided into three equal segments with each segment having its own strain gage. The next three elements correspond to the second bracket, second joint, and third bracket. Finally, the last three elements are the 2nd link divided into three equal segments, again with each segment having its own strain gage. The brackets are modeled as very stiff links with stiffness terms EI (Young's modulus times the area moment of inertia) of 10<sup>5</sup>. The motion of the manipulator is assumed to take place entirely within the horizontal plane.

The dynamic equations of the finite-element model can be compactly written in the following form

$$M(\theta)\ddot{\theta} + C(\theta)\dot{\theta^2} + K(\theta)I_1 = \bar{B}\tau$$
<sup>(1)</sup>

where  $\theta$  is a column vector containing the angular positions of the 11 finite elements, and  $\tau = [\tau_1 \ \tau_2]^T$  is the external torque vector where  $\tau_1$  and  $\tau_2$  are the applied torques at the joints. The term  $\dot{\theta}^2$  is an 11x1 vector where each element is the square of  $\dot{\theta}_i$ . The mass matrix is  $M(\theta) = [M_{i,j}\cos(\theta_i - \theta_j)], i = 1, ..., 11, j = 1, ..., 11$ , the centrifugal matrix is  $C(\theta) = [M_{i,j}\sin(\theta_i - \theta_j)], i = 1, ..., 11, j = 1, ..., 11$ , and the stiffness matrix is  $K(\theta) = [-K_{i,j}\sin(\theta_i - \theta_j)], i = 1, ..., 11, j = 1, ..., 11$ . The symbol  $I_1$  refers to a column vector of 11 ones. The matrix  $\overline{B}$  is an 11x2 matrix whose purpose is to transform the 2x1 control vector  $\tau$  into the 11x1 angular position space. It is defined as

The  $M_{i,j}$  and  $K_{i,j}$  are the mass and stiffness elements which are derived from the finiteelement model (see [9]) and are constants for a given structure (i.e., these elements depend only upon physical parameters of the system). Note that the  $\theta_i$  are absolute angles in the inertial coordinate system. Because of this absolute referencing, Eq. (1) does not contain Coriolis terms  $(\dot{\theta}_i \dot{\theta}_j, i \neq j)$ .

This model is efficient from a computational point of view since it only requires one evaluation of the mass and stiffness elements. It also has proven to be quite accurate in modeling the behavior of the actual manipulator. The experimental results in [11] show that the fundamental modal frequencies of the finite-element model closely approximate those of the experimental apparatus (approx. 20 Hz and 8 Hz for the first and second links respectively). In addition, the torque, angular velocity, and joint position profiles obtained by the model closely approximate those of the experimental structure for corresponding maneuvers. These results motivated the idea of combining the physical model with optimization techniques to generate torque histories that would follow a given trajectory in minimum-time with minimal endpoint vibration at the end of the slew. The goal is to track a specified straight-line path from rest-to-rest in minimum-time with no endpoint vibrations at the end of the maneuver while staying within the torque limits.

The optimization problem can be formally stated as follows [9]

minimize:  $J = t_f$ 

with respect to: the finite-element model, the actuator torque limits, and initial conditions subject to:

(3)

where x and y are the cartesian coordinates of the tip of the 2nd link or of the specified straight line trajectory, and  $t_f$  is the time it takes to complete the slew. The first two constraints assure the final tip position is met. The third constraint is an equality tracking constraint that keeps the tip position on the specified line. The next two constraints require the final kinetic and strain energies to be zero (i.e., final position at rest). The last two constraints force the final joint accelerations to be zero. The actuator torque limits were taken into account by limiting the torque values to 80% of the physical limits. This allowed room for additional feedback to be applied to the motors.

The optimizing program itself is a recursive quadratic programming algorithm and has been implemented on a CRAY X-MP at Sandia. It uses the aforementioned finite-element model to obtain the optimal torque profiles and the corresponding angular position and velocity values for all 11 elements at each sampling instant along the trajectory. The minimum time found by the algorithm to perform the slew was 2.058 seconds. These optimal torque, angular position, and angular velocity values are then used in various control schemes described here to perform the desired maneuver on the actual structure.

# 3 Feedforv.ard Control

As stated in the introduction, model-based control strategies can be very effective in improving the speed and transient response of a manipulator, but they are hampered by the accuracy of the model. Likewise, error-driven approaches have the advantage of getting to the desired final state but because the errors start out small, the initial torques will be small resulting in slow response time and significant overshoot. These realities motivated the strategy in [11] referred to here as feedforward control. This approach combines the advantages of model-based and sensor-based control by feeding back the errors in joint position, joint velocity, and curvature as measured by shaft encoders, tachometers, and strain gages, respectively. It also feeds forward the desired torque values as determined by the optimizing program. This strategy is illustrated in Figure 2 and motivates the approach undertaken in this paper.

Results in [11] show significant improvement in tracking the desired trajectory (a straight line of roughly 1.5 meters in length) than either open loop control based entirely on the modeled torque values or closed loop control based entirely on error feedback. When a modeling error consisting of a 0.275 kg tip mass is added, the improvement is even more pronounced. One of the drawbacks to the feedforward approach is that the feedback gains are determined experimentally. This implies a great deal of trial and error experimentation. The strategy in this paper is to choose these gains based on the model by minimizing a quadratic cost criterion. Linear optimal control theory is utilized to obtain these gains from the linearized finite-element model. The advantage of this approach is that it is easily duplicated for different structures and trajectories.

#### 4 LQR Control

In order to apply linear quadratic regulator theory, the dynamics in Eq. (1) must be put into state space form and then linearized about the desired trajectory. The state variables are chosen to be the 11 element positions and the 11 element velocities. That is,

$$z_i = \theta_i, i = 1, \dots, 11, \quad z_{i+11} = \dot{\theta}_i, i = 1, \dots, 11$$
 (4)

where  $z_i$  are the new state variables. Rewriting Eq. (1) we obtain

$$\ddot{\theta} = M(\theta)^{-1} [\tau - C(\theta)\dot{\theta}^2 - K(\theta)I_1]$$
(5)

thus arriving at the following state space description

$$\begin{vmatrix} \dot{z}_{1} \\ \dot{z}_{2} \\ \vdots \\ \dot{z}_{11} \\ \vdots \\ \dot{z}_{12} \\ \vdots \\ \dot{z}_{22} \end{vmatrix} = \begin{bmatrix} z_{12} \\ z_{13} \\ \vdots \\ z_{22} \\ \dots \\ M(z)^{-1}S(z) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \\ \dots & \dots \\ M(z)^{-1}\bar{B} \end{bmatrix} \tau$$
(6)

which can be written in the more compact form

$$\dot{z} = f(z) + g(z)\tau. \tag{7}$$

The matrix M(z) is simply the matrix  $M(\theta)$  with the  $\theta_i$ 's replaced with the appropriately numbered states  $z_i$ . The control vector  $\tau$  is as described earlier, and the matrix S(z) is the matrix  $S(\theta) = -C(\theta)\dot{\theta}^2 - K(\theta)I_1$  with  $\theta$  replaced by the state vector z as follows

$$S(z) = \begin{bmatrix} \sum_{i=1}^{11} [M_{1,i} \sin(z_i - z_1) z_{i+11}^2 + K_{1,i} \sin(z_1 - z_i)] \\ \sum_{i=1}^{11} [M_{2,i} \sin(z_i - z_2) z_{i+11}^2 + K_{2,i} \sin(z_2 - z_i)] \\ \vdots \\ \sum_{i=1}^{11} [M_{11,i} \sin(z_i - z_{11}) z_{i+11}^2 + K_{11,i} \sin(z_{11} - z_i)] \end{bmatrix}.$$
(8)

The state space Eq. (7) is a 22nd order nonlinear ordinary differential equation. This equation must be linearized about the trajectory in order to obtain a linear model useful for applying quadratic regulator theory. To do the linearization, we note that a function f(z) can be expanded as a Taylor series about an operating point  $\bar{z}$  in the following manner

$$f(z) = f(\bar{z}) + \frac{\partial f}{\partial z} |_{z=\bar{z}} \left( z - \bar{z} \right) + O\left( (z - \bar{z})^2 \right).$$
(9)

Since Eq. (7) is affine in the control  $\tau$ , we only need to linearize about the state and not the control. We let x represent the perturbation in the state z abov the operating point  $\bar{z}$ . That is,

$$z = \bar{z} + x \,. \tag{10}$$

Substituting Eqs. (9) and (10) into Eq. (1) and truncating second and higher order terms we obtain the following linearized model

$$\dot{x} = \frac{\partial f}{\partial z} |_{z=\bar{z}} x + g(\bar{z})\tau = Ax + B\tau$$
(11)

where x now represents deviations from the desired trajectory which we would like to drive to zero.

The matrices A and B can be computed as follows

$$A = \frac{\partial f}{\partial z} |_{z=\bar{z}} = \begin{bmatrix} 0 & 0 & \cdots & 0 & \vdots & 1 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 & \vdots & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & \vdots & 0 & 0 & \cdots & 1 \\ \cdots & \cdots \\ H_1 & H_2 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ H_1 & H_2 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ B = g(z) |_{z=\bar{z}} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ M(z)^{-1} |_{z=\bar{z}} \bar{B} \end{bmatrix}$$
(12)

where  $H_i$  is a gradient vector computed as

$$H_{i} = \frac{\partial (M(z)^{-1}S(z))}{\partial z_{i}} |_{z=\bar{z}} = \left[ M(z)^{-1} \frac{\partial S(z)}{\partial z_{i}} - M(z)^{-1} \frac{\partial M(z)}{\partial z_{i}} M(z)^{-1} S(z) \right] |_{z=\bar{z}}$$
(14)

and  $\bar{z}$  has the units of radians and radians/sec. The gradients that are needed to compute the above are included in the appendix.

Since the desired trajectory is a straight line, the operating point  $\bar{z}$  must be chosen as a fixed point along this trajectory. To choose this operating point as the trajectory itself would result in a linear time-varying system that would be difficult to analyze. Thus, it was decided that the linearization would be done about a series of points along the trajectory, e.g., every 200 milliseconds. This implies that there will be a different set of gains for each operating point. The control algorithm then calls for an update of gains each time a new operating point is reached. Furthermore, it will be assumed that at each operating point the linearized system reaches steady state so that the Riccati equation that must be solved is an algebraic one. This will result in constant gains so that again the linearized system will be time-invariant.

The linear quadratic regulator problem is to minimize a quadratic cost criterion

$$J = \frac{1}{2} \int_0^\infty (x^T Q x + u^T R u) dt \tag{15}$$

with respect to the control vector u subject to the system's dynamics

$$\dot{x} = Ax + Bu. \tag{16}$$

The solution to this problem involves solving an Algebraic Riccati Equation (ARE) [14] for a matrix P

$$0 = A^{T}P + PA - PBR^{-1}B^{T}P + Q$$
(17)

which results in the feedback equation

$$u = -R^{-1}B^T P x aga{18}$$

Thus, the gains are equal to  $-R^{-1}B^TP$ , and our linearized system (Eq. (11)) matches the form of Eq. (16) with  $\tau$  as the control vector u.

Once the operating points  $\bar{z}$  have been chosen the only variables in the LQR problem are the choice of Q and R matrices. The Q matrix is a 22x22 matrix which places penalties on the states of the system. It was decided for simplicity that Q would be a diagonal matrix which implies that there is no penalty on cross-state interaction. The larger these diagonal elements are the more important it is that the corresponding state be driven to zero (i.e., the actual state be driven to the trajectory). The R matrix is 2x2 and again it was chosen to be a diagonal matrix. The larger the element is the more penalty is placed on that particular motor. This is useful if one motor is considerably more expensive to run than the other. Once these matrices are decided upon, the schedule of gains can be determined. For computational purposes, the system software package MATLAB was utilized. The complete block diagram of the LQR control strategy is illustrated in Figure 3. In the next two sections, simulation and experimental results are presented which use gains that were calculated from the information in this section.

#### 5 Simulation Results

For the simulations, a program implementing the finite-element model in [9] was utilized to determine which schedule of gains resulting from a wide choice of Q and R matrices would be

best for the experimental setup. The simulations were also undertaken to see if estimating many of the 22 states that are not measurable is feasible or if perhaps some of these states can be ignored altogether. It is also important to determine the accuracy of the model since the feedforward torque values used on the actual system were obtained from the model. A C program run on a SUN 4 workstation did the simulations using a 1 millisecond sampling interval and read in the optimizing values of torque, angular position and velocity, as well as the optimal gain schedule. The results are tabulated in Tables 1-3 according to the different sets of gains obtained by using different Q and R matrices. Also tabulated in Tables 1-3 are results obtained when a modeling error represented by a 0.275 kg tip mass is added as well as observation noise in estimating the states.

Some of the results not tabulated here indicated that trying to estimate the 9 velocity states that are unmeasurable (the two joint velocities are measured by the tachometers) from the position states via finite differences resulted in very poor performance and erratic results. These results also showed that feeding back position states only with no velocity feedback resulted in an undamped system that oscillated wildly. Therefore, it was decided that the best approach for the experimental system was to use the two velocity states measurable from the tachometers and not attempt to feed back the remaining 9 velocity states. The R matrix was chosen so as to put equal weight on both motor torques. After much simulation, a value of 10 was chosen for the two diagonal elements. This choice of R proved to be the most successful. Thus,

$$R = \begin{bmatrix} 10 & 0\\ 0 & 10 \end{bmatrix} . \tag{19}$$

The choice of entries for the Q matrix was more difficult. The first 11 diagonal elements represent penalties on the position states and the remaining 11 elements represent penalties on the velocity states. It was initially decided to penalize the position states more in order to keep the system on the desired trajectory. All 11 position states were assigned a penalty of 10, and the penalty on the velocity states was varied between 0 and 1. Thus we have

$$Q = \begin{bmatrix} 10I_{11\times11} & 0_{11\times11} \\ 0_{11\times11} & \gamma I_{11\times11} \end{bmatrix}$$
(20)

where  $\gamma$  is the parameter to be varied between 0 and 1.

In the tabulated data, six quantities of interested are listed. The peak torque for each joint is tabulated to see that it stays within the physical limits of the structure which are 7.0 and 2.25 N-m for the first and second joints respectively. The peak position and velocity errors among all 11 elements over the entire simulation run were listed in degrees and degrees/sec. Finally, the rms (root-mean-square) errors in position and velocity were tabulated in degrees and degrees/sec. The simulation was run for 2058 sampling instants or 2.058 seconds, the same length of time the optimization was carried out to determine the modeled torque values. Measurement noise was included in the simulations to model the effect of sensor noise. The shaft encoder noise was modeled as zero mean uniformly random noise with variations of  $\pm 0.1$  degrees and the tachometer noise was also modeled as zero mean uniformly random noise with variations of  $\pm 1.0$  degrees/sec. The estimated error in curvature as measured from the strain gages was modeled as zero mean uniformly random noise with variations of  $\pm 0.05$  degrees. The computer results of the simulations are compiled in Tables 1-3. The plot of tip position superimposed over the modeled trajectory for  $\gamma = 0.1$  is shown in Figure 4. Also on the same graph is the corresponding experimental tip position for the same gains ( $\gamma = 0.1$ ). The discrepancy between the two plots, particularly the overshoot at the end, is mostly due to modeling errors especially the underestimation of the magnitude of sensor noise in the simulations. A row of dashes in the tables implies the output became numerically unbounded.

The results for  $\gamma = 1$  show that in the presence of sensor noise both with and without the tip mass the simulations become numerically unstable. This is primarily due to the fact that for  $\gamma = 1$  both position and velocity gains are considerably larger (on the order of 10 to 100) than for smaller  $\gamma$ 's, resulting in large oscillations when sensor noise is introduced. In addition, the lack of joint dampening in the model and the one millisecond integration stepsize of the simulation are also contributing factors. In the absence of measurement noise, the results are quite good with or without the modeling error. The results for  $\gamma = 0.1$  show a much greater tolerance to measurement noise. Here, the system remains stable even with the tip mass. This is partly due to the reduction of gain magnitude (roughly 1 to 10 for this case). In particular, the velocity gains are smaller which reduces the effect of the tachometer noise which was modeled as being much larger than the shaft position or strain gage noise.

The final choice of  $\gamma = 0$  produced results very similar to that of  $\gamma = 0.1$ , though the gains were much smaller in magnitude (less than 1). In this case there was no penalty placed on the velocity states. This resulted in very small velocity gains, but the position gains were also small. This may explain why the data with the tip mass are not as accurate as they are for  $\gamma = 0.1$ . The small gains do not provide enough of an error signal to compensate for the modeling error. In all three cases, the peak values of joint torque remained within the physical limits of the motors. It can also be observed from the tables that the addition of the tip mass (which is a modeling error not accounted for in the finite-element model) substantially degraded performance in tracking the desired joint velocities. However, the positions remained just a few degrees from their desired values. On the other hand, with the exception of the  $\gamma = 1$  case, the addition of sensor noise only marginally decreased performance.

	no tip mass		0.275 kg tip mass	
	no noise	with noise	no noisc	with noise
peak torque (1st joint), N-m	6.044	*********	6.253	
peak torque (2nd joint), N-m	1.336		0.859	
peak position error, deg	0.228		8.576	
peak velocity error, deg/sec	17.888	' ·	145.494	
rms position error, deg	0.061		1.856	
rms velocity error, deg/sec	2.147		20.469	

Table 1: Simulation results for  $\gamma = 1$ .

	no tip mass		0.275 kg tip mass	
	no noise	with noise	no noise	with noise
peak torque (1st joint), N-m	6.024	6.051	6.232	6.107
peak torque (2nd joint), N-m	1.222	1.24	0.878	0.86
peak position error, deg	0.175	1.413	9.259	9.59
peak velocity error, deg/sec	10.182	48.121	152.786	157.536
rms position error, deg	0.038	0.333	1.943	2.664
rms velocity error, deg/sec	1.404	5.103	21.322	20.803

Table 2: Simulation results for  $\gamma = 0.1$ .

	no tip mass		0.275 kg tip mass	
	no noise	with noise	no noise	with noise
peak torque (1st joint), N-m	6.012	6.032	6.231	6.092
peak torque (2nd joint), N-m	1.23	1.222	0.99	0.972
peak position error, deg	0.207	0.689	9.647	11.59
peak velocity error, deg/sec	16.67	15.643	184.636	186.772
rms position error, deg	0.041	0.171	1.962	3.05
rms velocity error, deg/sec	3.208	2.676	24.711	22.058

Table 3: Simulation results for  $\gamma = 0$ .

١

In the above simulations, the gains were switched every 200 milliseconds for a total of 11 sets of gains. Other simulations were run in which the gains were switched every 20 milliseconds and still more simulations were attempted with just one set of gains for the entire 2.058 seconds. The results of these variations in scheduling showed no noticeable difference in performance regardless of how many times the gains were switched, i.e., the gains remained fairly constant for different operating points along the trajectory. Thus, it was determined for the sake of simplicity that one set of gains corresponding to  $\gamma = 0.1$  will probably be best for the actual structure because it was effective in handling both sensor noise and modeling error.

### 6 Experimental Results

The complete robotic system includes the two-link flexible arm structure, two motor amplifiers, two DC servo  $I_{112} \rightarrow I_{21}$ , two motor encoders, two motor tachometers, three strain gages attached to each link, an accelerometer at the tip of each link (see Figure 1), and a VMEbus computer system. The motor amplifiers and DC servo motors were chosen so that the control inputs to the system are the motor torques, i.e., the amplifiers essentially act as current regulators for the motors. The peak motor torques are 7 N-m (996 oz-in) for the motor at the base and 2.25 N-m (318 oz-in) for the motor at the elbow. Given the weights and moments of inertia of the links and the motor hubs, this translates to angular accelerations of approximately 152 deg/sec<sup>2</sup> for joint 1 and 2918 deg/sec<sup>2</sup> for joint 2.

The links of the robot arm are made of aluminum. The flexible part of link 1 is 0.46m by 15.24cm by 4.76mm, and the flexible part of link 2 is 0.49m by 7.62cm by 1.59mm. The EI's of the links are 98.6 and 1.75 N-m<sup>2</sup> for the first and second links respectively. Assuming a cantilever beam, the theoretical first modal frequencies of link 1 and link 2 are approximately 18.1 Hz and and 5.4 Hz, respectively. These are fairly close to the experimentally measured natural frequencies of about 20 Hz and 8 Hz. These frequencies vary only slightly with the position of the links.

A VME-based 68030 computer, encoder counter board, and I/O board are used to control the system. The controller's sampling time is 1 millisecond. Two twelve bit encoder counters measure the angles of the motors. Eight twelve bit A/D channels are used to measure the tachometer of each joint, the six strain gages, and the two accelerometers. To reduce noise in the system, the strain gage signals are first filtered by 100 Hz low-pass filters before being sampled, and the accelerometer signals are filtered by 10 Hz low-pass filters. These filters are present only in the experimental system and were not included in the simulations. Two twelve bit D/A channels are used to output the desired torques to the motor amplifiers. Experimental results have already been obtained utilizing the feedforward approach [11]. These results showed that open loop control using the modeled torque values only was not adequate due to inherent modeling error as well as external modeling error such as a payload. Error driven control only (such as PD control) also is not adequate due to large overshoots if gains are small or instability if gains are large. But, the feedforward approach which combines the two strategies yielded accurate tracking even with a payload.

The experimental results for the LQR strategy of Section 4 are obtained using the hardware described above. A program written in C implements the LQR approach on the actual system. Since the sensors available on the structure cannot possibly provide accurate measurements for all 22 system states, an estimation scheme is necessary. As in the simulations, only the two velocity states directly provided by the tachometers are utilized with the other 9 velocity states ignored. The 11 measured position states are constructed via the shaft encoder and strain gage signals in the following manner

$$\theta_1 = q_1 \tag{21}$$

$$\theta_2 = \theta_1 \tag{22}$$

$$\theta_3 = \theta_2 + \epsilon_{1,1} \tag{23}$$

$$\theta_4 = \theta_3 + \epsilon_{1,2} \tag{24}$$

$$\theta_5 = \theta_4 + \epsilon_{1,3} \tag{25}$$

$$\theta_6 = \theta_5 \tag{26}$$

$$\theta_7 = \theta_6 + q_2 \tag{27}$$

$$\theta_8 = \theta_7 \tag{28}$$

$$\theta_9 = \theta_8 + \epsilon_{2,1} \tag{29}$$

(00)

$$\theta_{10} = \theta_9 + \epsilon_{2,2} \tag{30}$$

$$\theta_{11} = \theta_{10} + \epsilon_{2,3} \tag{31}$$

where the  $\epsilon_{i,j}$  are the curvatures as measured by the *j*th strain gage on the *i*th link, and the  $q_i$  are the *i*th shaft encoder positions. The strain gage measurements are just numbers representing D/A counts. These values are converted into angular measurements by multiplying them by a curvature conversion factor. This factor is obtained by comparing simulated curvatures (in units of radians/meter) to the experimental strain values for a pulse response maneuver. The measurement is then multiplied by the length of the element (one-third of the link length) to get the value in radians.

The actual error states are constructed by subtracting the desired values of these 11 states (determined by the optimizing program and the finite-element model as described in

Section 2) from the above measured values. Each joint torque assumes full state feedback (all 22 states except for the 9 ignored velocity states). In the experiments, gain switching was attempted with no noticeable difference in results, therefore, only one set of gains was employed for the actual maneuver.

In addition to the choices of Q and R matrices presented in the simulations, system performance for other values was studied. These resulted in large overshoots at the end of the tip trajectory, motivating further refinements in choosing Q. The R matrix remained unchanged, but it was decided that more penalty should be placed on the two velocity states to result in less overshoot (due to increased damping). In particular, the penalty on the first joint velocity was increased substantially since most of the observed overshoot was in the first link. The resulting Q matrix was

 $Q = \begin{bmatrix} 10I_{11\times11} & \vdots & 0_{11\times11} \\ & \ddots & \vdots & \ddots & \ddots & \ddots \\ & \vdots & 80 & & \\ 0_{11\times11} & \vdots & 0.8I_{5\times5} & \\ & \vdots & & 4 & \\ & \vdots & & 0.8I_{4\times4} \end{bmatrix}$ 

(32)

The plots resulting from these gains are shown in Figures 5 and 6. Figures 5(a),(b)show the tip position without and with the 0.275 kg tip weight, respectively. The results are an improvement over the previous gains since there is less overshoot and hence faster settling time. Also shown on these graphs are the tip position plots for the feedforward method. It can be seen that the feedforward method results in less overshoot and better tracking of the trajectory. But, the gains for the feedforward method were hand-tuned (trial and error experimentation) to get good performance. An important point to be made here is that the LQR method could be used to obtain a good first choice of gains then some hand-tuning could be implemented to further refine the gains. This would be much easier than the feedforward method to implement because the LQR procedure provides a good starting point. In fact, this was done with very little refining needed to get just as good a performance as the feedforward approach. In both methods, it takes a little more than the optimal time of 2.058 seconds to reach the quiescent point. This is because the errordriven control component must assume the role of driving the system to the quiescent point due to inherent modeling errors. Both methods require an additional 2 seconds to reach a vibrationless state.

Figures 6(a),(b) illustrate the torque profiles of each joint for the LQR method with the modeled (open loop) torque profiles superimposed. The torque profile of the LQR control

shows how the errors in states perturb the modeled torque trajectory. These plots are virtually identical to the torque profiles for the feedforward method [11], which indicates that both methods have similar performance. The chattering in the torque profile is primarily due to the velocity feedback which is noisier than the position feedback.

### 7 Conclusions

This paper presents a methodology for controlling a two-link flexible robot arm along a specified path in minimum time. An optimization program plans the motion based on a finiteelement model, motion constraints, and motor torque limits. To account for modeling errors, an error-driven component, consisting of the errors between the actual state parameters and the desired states according to the model, is added to the computed torque. An LQR design strategy is implemented to determine the optimal feedback gains for this error-driven component. Specific states and joint controls were penalized by increasing or decreasing the corresponding entries in the Q and R matrices of the quadratic cost criterion. Based on the experimental results, a promising strategy is to obtain an initial set of gains via the LQR method that is optimal for the given model and then improve on these gains by hand-tuning them on the physical structure. This combination strategy will converge to a desired performance level quicker than either method by itself.

This method of control would be best applied to manipulators performing repetitive motion such as assembly robots or high bandwidth micro-manipulators (e.g., disk drive positioning arms). While the optimization is time consuming, it would be cost effective if the motion was to be repeated hundreds of times. Future work will look at improvements to the LQR strategy such as considering feedback of integrated position states to reduce noise effects as well as formulating the problem in discrete time. In addition, an LQG design whereby a Kalman filter is used to estimate the states could alleviate the problem of full state feedback necessary in the LQR method.

# Acknowledgements

We would like to express our sincere gratitude to two other people for their involvement in this research project. First, we would like to thank Alan Morimoto for designing and building Sandia's two-link flexible robotic arm. Second, we would like to thank Rush Robinett for his continual enlightenment on the subject of modeling and control of flexible structures.

# References

- [1] An, C.H., Atkeson, C.G., Griffiths, J.D., and Hollerback, J.M., "Experimental Evaluation of Feedforward and Computed Torque Control," *IEEE Transactions on Robotics* and Automation, Vol. 5, No. 3, pp. 3/8-372, June 1989.
- [2] Petterson, B.J., and Robinett, R.D., "Model Based Damping of Coupled Horizontal and Vertical Oscillations in a Flexible Rod," AIAA Guidance, Navigation, and Controls Conference, Boston, MA, August 14-16, 1989.
- [3] Singer, N., "Residual Vibration Reduction in Computer Controlled Machines," Technical Report AI-TR 1030, MIT Artificial Intelligence Laboratory, Cambridge, MA, January 1989.
- [4] Siciliano, B. and Book, W., "A Singular Perturbation Approach to Control of Lightweight Flexible Manipulators," The International Journal of Robotics Research, Vol. 7, No. 4, pp. 79-90, August 1988.
- [5] Khosla, P.R., and Kanade, T., "Parameter Identification of Robot Dynamics," Proceedings of 24th Conference on Decision and Control, pp. 1754-1760, Ft. Lauderdale, FL, December 1985.
- [6] Tarn, T.J., Bejczy, A.K., and Ding, X., "On the Modelling of Flexible Robot Arms," Department of Systems Science and Mathematics, Washington University, St. Louis, MO, Robotics Laboratory Report SSM-RL-88-11, September 1988.
- [7] Kotnik, P., Yurkovich, S., and Özgüner, Ü., "Acceleration Feedback for Control of a Flexible Manipulator Arm," Journal of Robotic Systems, Vol. 5, No. 3, pp. 181-196, June 1988.
- [8] Wells, R.L., Schueller, J.K., and Tlusty, J., "Feedforward and Feedback Control of a Flexible Robotic Arm," *IEEE Control Systems Magazine*, Vol. 10, No. 1, pp. 9-15, January 1990.
- [9] Eisler, G.R., Segalman, D.J., and Robinett, R.D., "Approximate Minimum-Time Trajectories for Two-link Flexible Manipulators," *Proceedings of the 1990 American Control Conference*, San Diego, CA, May 23-25, 1990.
- [10] Segalman, D.J., "A Mathematical Formulation for the Rapid Simulation of a Flexible Multilink Manipulator," SAND89-2308, Sandia National Laboratories, Albuquerque, NM.

- [11] Feddema, J.T., Eisler, G.R., and Segalman, D.J., "Integration of Model-Based and Sensor-Based Control of a Two-link Flexible Robot Arm," *Proceedings of the 1990 IEEE Conference on Systems Engineering*, Pittsburgh, PA, August 9-11, 1990.
- [12] Khorrami, F., "Analysis of Multi-link Flexible Manipulators via Asymptotic Expansions," Proceedings of the 28th IEEE Conference on Decision and Control, pp. 2089-2094, Tampa, FL, December 1989.
- [13] Schoenwald, D.A., Özgüner, Ü., and Chan, H., "An Analysis of Distributed Vibration Control of Flexible Manipulators using Integral Manifolds," *Proceedings of the 28th IEEE Conference on Decision and Control*, pp. 2095-2100, Tampa, FL, December 1989.
- [14] Anderson, B.D.O., and Moore, J.B., *Optimal Filtering*, Englewood Cliffs, NJ: Prentice-Hall, 1979.
- [15] Ward, L.S., "Dynamics and Control of a Nonlinear Flexible Manipulator," M.S. thesis, Massachusetts Institute of Technology, Cambridge, MA, September 1986.

# Appendix

The gradients that are needed to compute the linearized state space model of the flexible manipulator are as follows.

$$\frac{\partial M(z)}{\partial z_1} = \begin{bmatrix} 0 & -M_{1,2}\sin(z_1 - z_2) & \cdots & -M_{1,11}\sin(z_1 - z_{11}) \\ -M_{2,1}\sin(z_1 - z_2) & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ -M_{11,1}\sin(z_1 - z_{11}) & 0 & \cdots & 0 \end{bmatrix}$$
(33)

$$\frac{\partial M(z)}{\partial z_2} = \begin{bmatrix} 0 & M_{1,2} \sin(z_1 - z_2) & \cdots & 0 \\ M_{2,1} \sin(z_1 - z_2) & 0 & \cdots & -M_{2,11} \sin(z_2 - z_{11}) \\ \vdots & \vdots & \ddots & \vdots \\ 0 & -M_{11,2} \sin(z_2 - z_{11}) & \cdots & 0 \end{bmatrix}$$
(34)

$$\frac{\partial M(z)}{\partial z_{11}} = \begin{bmatrix} 0 & 0 & \cdots & M_{1,11} \sin(z_1 - z_{11}) \\ 0 & 0 & \cdots & M_{2,11} \sin(z_2 - z_{11}) \\ \vdots & \vdots & \ddots & \vdots \\ M_{11,1} \sin(z_1 - z_{11}) & M_{11,2} \sin(z_2 - z_{11}) & \cdots & 0 \end{bmatrix}$$
(35)

$$\frac{\partial M(z)}{\partial z_i} = [0]_{11 \times 11} , i = 12, \dots, 22$$
 (36)

$$\frac{\partial S(z)}{\partial z_1} = \begin{bmatrix} -\sum_{i=2}^{11} M_{1,i} \cos(z_1 - z_i) z_{i+11}^2 + \sum_{i=2}^{11} K_{1,i} \cos(z_1 - z_i) \\ M_{2,1} \cos(z_1 - z_2) z_{12}^2 - K_{2,1} \cos(z_1 - z_2) \\ \vdots \end{bmatrix}$$
(37)

$$\frac{\partial S(z)}{\partial z_2} = \begin{bmatrix} M_{11,1}\cos(z_1 - z_{11})z_{12}^2 - K_{11,1}\cos(z_1 - z_{11}) \\ M_{1,2}\cos(z_1 - z_2)z_{13}^2 - K_{1,2}\cos(z_1 - z_2) \\ -\sum_{i=1,i\neq 2}^{11} M_{2,i}\cos(z_2 - z_i)z_{i+11}^2 + \sum_{i=1,i\neq 2}^{11} K_{2,i}\cos(z_2 - z_i) \\ \vdots \\ M_{11,2}\cos(z_2 - z_{11})z_{13}^2 - K_{11,2}\cos(z_2 - z_{11}) \end{bmatrix}$$
(38)

÷

$$\frac{\partial S(z)}{\partial z_{11}} = \begin{bmatrix} M_{1,11} \cos(z_1 - z_{11}) z_{22}^2 - K_{1,11} \cos(z_1 - z_{11}) \\ M_{2,11} \cos(z_2 - z_{11}) z_{22}^2 - K_{2,11} \cos(z_2 - z_{11}) \\ \vdots \\ -\sum_{i=1}^{10} M_{11,i} \cos(z_{11} - z_i) z_{i+11}^2 + \sum_{i=1}^{10} K_{11,i} \cos(z_{11} - z_i) \end{bmatrix}$$
(39)

$$\frac{\partial S(z)}{\partial z_{12}} = \begin{bmatrix} 2M_{2,1}\sin(z_1 - z_2)z_{12} \\ \vdots \\ 2M_{11,1}\sin(z_1 - z_{11})z_{12} \end{bmatrix}$$
(40)  
$$\frac{\partial S(z)}{\partial z_{13}} = \begin{bmatrix} 2M_{1,2}\sin(z_2 - z_1)z_{13} \\ 0 \\ 2M_{3,2}\sin(z_2 - z_3)z_{13} \\ \vdots \\ 2M_{11,2}\sin(z_2 - z_{11})z_{13} \end{bmatrix}$$
(41)

$$\frac{\partial S(z)}{\partial z_{22}} = \begin{bmatrix} -2M_{1,1} \sin(z_1 - z_{11})z_{22} \\ -2M_{2,11} \sin(z_2 - z_{11})z_{22} \\ \vdots \\ -2M_{10,11} \sin(z_{10} - z_{11}) \\ 0 \end{bmatrix}$$
(42)

with the last 22 gradients being 11x1 vectors.

19

:



Figure 1(a). Sandia two-link, flexible robot arm.















Figure 4. Simulated and experimental tip trajectories for  $\gamma$ =0.1 and no payload.



Figure 5(a). Tip position without payload.



Figure 5(b). Tip position with 0.275 kg payload.



Figure 6(a). Joint 1 torque resulting from LQR control.



Figure 6(b). Joint 2 torque resulting from LQR control.

#### DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.



