OPTIMAL TRAJECTORY OF FLEXIBLE MANIPULATOR WITH MAXIMUM LOAD CARRYING CAPACITY

M. Habibnejad Korayem

Department of Mechanical Engineering
Iran University of Science & Technology
Tehran Iran

Abstract In this paper, a new formulation along with numerical solution for the problem of finding a point-to-point trajectory with maximum load carrying capacities for flexible manipulators is proposed. For rigid manipulators, the major limiting factor in determining the Dynamic Load Carrying Capacity (DLCC) is the joint actuator capacity. The flexibility exhibited by light weight robots or by robots operating at a higher speed dictates the need for an additional constraint to be imposed for situations where precision tracking i. e., the allowable deformation at the end effector is required. An Iterative Linear Programming (ILP) method is used to optimise the load (load mass and load moment of inertia) of elastic robot subject to both constraints. A general computational procedure for the multiple-link case is presented in detail. Simulation is carried out for a two-link planer robot. The results confirm the necessity of the dual constraints.

Key Words Maximum Load, Flexible Manipulator, Trajectory Optimization

چکیده در این مقاله یک فرمولاسیون جدید و حل عددی آن به منظور پیدا کردن مسیر حرکت رباتهای الاستیک پیشنهاد شده است. مهمترین محدودیت رباتهای صلب در تعیین بار دینامیک ماکزیمم، ظرفیت موتورها میباشد در حالیکه اثرات انعطاف پذیری بازوها ناشی از وزن سبک این چنین بازوها و همچنین سرعتهای زیاد حرکت، نیازمندی به یک قید دیگری را پیشنهاد می کند که با توجه به مقدار مجاز انحراف پنجه قابل تعریف میباشد. روش برنامه نویسی برگشت پذیری خطی به منظور پیدا کردن بار (جرم بار و ممان اینرسی بار) رباتهای الاستیک با در نظر گرفتن دو قید بکار گرفته شده و یک برنامه عمومی برای حالت چند لینکی با جزئیات ارائه شد. مشابه سازی حرکت برای یک ربات دولینکی بعمل آمد و نتایج ضرورت اعمال دو قید را بیشتر تائید کرد.

INTRODUCTION

The Dynamic Load Carrying Capacity (DLCC) of a flexible manipulator is often defined as the maximum payload that the manipulator can repeatedly lift in its fully extended configuration [1]. In order to determine DLCC, the inertia effect of the load along a desired trajectory as well as the manipulator dynamics itself must be taken into account. The models in [1-2] have the property that determine the load carrying capacity of a robot manipulator for a given dynamic trajectory. Although there were challenges encountered in determining DLCC, the problem was confined to the solution of maximum load for two given and positions. Given two end-positions of the end effector, the problem is to synthesize a dynamic trajectory for the end effector that would allow carrying the maximum allowable load between the two end-positions. The goal of optimal trajectory is to find the trajectory $\mathbf{X}(t)$ and the input torque $\mathbf{T}(t)$ starting from the initial state $\mathbf{X}(t_i)$ and arriving at the final state $\mathbf{X}(t_f)$ with maximum load carrying capacity. The problem of synthesizing a point-to-point dynamic robot trajectories with maximum load carrying capacity can be formulated as a trajectory optimization problem using the state representation of dynamic equations.

Wang and Ravani [3] have shown that the problem

of synthesizing a point-to-point dynamic robot motion with optimum load carrying capacity can be formulated as a trajectory optimization problem. They applied the method of Iterative Linear Programming (ILP) to solve optimal trajectory problems which have the control forces/torques bounded, and the travel time given. This method is applicable only when the manipulator arm is considered to be rigid. When the manipulator links are subjected to large accelerations, structural flexibilities may be excited. This will violate joint constraint, i. e., we can no longer determine the joint angles from the distance along the path. Hence the torques drive for the rigid manipulator will not drive the flexible manipulator exactly to the desired final state.

By removing the rigid-link assumption, flexible manipulators present some common and some distinctive characteristics compared to rigid systems. Along with the general objective of a minimum-time trajectory subject to torque and/or joint constraints, the elimination of tip vibrations has been observed as a critical problem in flexible manipulators. A first step in trajectory planning based in the inverse dynamic solution was presented by Bayo and Paden [4], where they outlined a comparison between the bang-bang and the Gaussian acceleration profiles. Recently Serna and Bayo [5] proposed a trajectory planner for point-to-point motion, based on the solution of the inverse dynamic problem for flexible manipulators in the frequency domain.

Given two end-positions of the end effector, the trajectory consists of a set of directions from the initial point to the final point. Not knowing which of those points the end effector should pass through on an optimal trajectory, one can determine the optimal path to, say, maximize load, from each of these points to the final point. It is argued that, whatever trajectory is chosen, it must be optimal with respect to the maximum allowable load. Otherwise, the optimal trajectory determined under the actuator constraint alone [3] becomes inadequate. The determined "maximum load" is adequate for the size of actuators but does not show how precisely the robot can reach to the end point under such a load. It is apparent that a flexible manipulator is expected to have the capability to reach the end point in such applications. Therefore, an additional constraint must be imposed when optimal trajectory is to be determined for flexible manipulators.

This paper presents a new method to determine optimal trajectory for flexible manipulators subject to both actuator and end effector deflection constraints. First, the recursive Lagrangian assumed mode method [6] is modified to accommodate the load dynamics, which together with kinematic equations are necessary to determine optimal trajectory. Then, the state space representation of the dynamic equations of motion is presented and the nonlinear state space dynamic equations are linearized. The problem of synthesizing optimum robot trajectories is formulated as a trajectory optimization problem. The trajectories that can carry maximum load are synthesized by numerical solution of the optimization problem. Finally, the ILP method and the computational procedure for computing such optimal trajectory are developed. The procedure allows synthesizing point-to-point robot motions with a specified time and maximum load carrying capacity. The algorithm takes into account the complete dynamic equations and generalized coordinates and actuator constraints. To evaluate the performance of the proposed method, simulation test is carried out.

STATE SPACE REPRESENTATION OF DYNAMICS EQUATIONS AND LINEARIZATION

To determine an optimal trajectory for flexible manipulators, proper modeling of the manipulator and load dynamics is a prerequisite. The method em-

ployed here largely follows that of Book [6], with the addition of dynamic effects of load at the end effector as well as mass at joints to account for actuator and gearing inertia. It is assumed that end effector deflection is primarily caused by link deflection or oscillation at higher speeds, and links are slender beams. By using Lagrange's equations of motion, the dynamic equations of a flexible manipulator are obtained with generalized coordinates q_k and q_{hk}. The resultant system of equations can be organised in a matrix form as

$$\mathbf{J}\ddot{\mathbf{z}} = \mathbf{R} \tag{1}$$

where the elements of **J** and **R** are slightly different from those given in [6]. (**z** is a vector of generalized coordinates $[q_1, q_2, \ldots, q_k, \ldots, q_n, q_{11}, q_{12}, \ldots, q_{1m_1}, q_{21}, \ldots, q_{2m_2}, \ldots, q_{k1}, \ldots, q_{km_k}, \ldots, q_{nm_n}]^T$)

The dynamic model of a robotic arm is formulated in Equation 1 as a second-order system of n_t differential equations where n_t is a total number of generalized coordinates. To facilitate optimization problem, it is helpful to formulate the equations of motion as a first-order system of $2n_t$ equations called state equations. In this section, first the state space formulation of the robot dynamic equation is introduced. Then, the nonlinear state space dynamic equations are linearized.

State Space Representation of Dynamic Equations

When a load is carried by the end-effector, it can be assumed as a variable and modeled as a point mass m_p. Equation 1 can be written as

$$\ddot{\mathbf{z}} = \mathbf{J}^{-1}\mathbf{R} = \mathbf{f}(\mathbf{X}(j), \mathbf{T}(j), m_p) \tag{2}$$

where **T** is an $n \times 1$ vector of the actuator applied torques (forces). By defining the state vector $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2]^t$ where $\mathbf{x}_1 = \mathbf{z}$ and $\mathbf{x}_2 = \dot{\mathbf{z}}$. Equation 2 can be written as follows

$$\dot{\mathbf{X}} = \begin{bmatrix} \mathbf{x}_1 \\ \cdots \\ \mathbf{x}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{x}_2 \\ \cdots\cdots\cdots\cdots \\ \mathbf{f}(\mathbf{X}(j), \mathbf{T}(j), m_p) \end{bmatrix} \tag{3}$$

Equation 3 is the state space representation of the dynamic Equation 1. **X** is a $2n_t \times 1$ vector and $\mathbf{f}(\mathbf{X}(j), \mathbf{T}(j), m_p)$ consists of n_t non-linear functions.

Linearization of Dynamic Equations

The linearized dynamic equations are needed in order to obtain the numerical solution to the nonlinear trajectory optimization problem. Assuming that the effects of the flexibilities are small, the nonlinear state space dynamic equations can be linearized.

The discretized form of the state space dynamic Equation 3 is

$$\frac{\mathbf{X}(j+1) - \mathbf{X}(j)}{\Omega} = \Phi(\mathbf{X}(j), \mathbf{T}(j), m_p) \tag{4}$$

where $\Omega = (t_f - t_i)/m$ and m is the total number of set points used to discretize the end-effector trajectory. Substituting Equation 4 into Equation 3 leads to

$$\Phi(\mathbf{X}(j), \mathbf{T}(j), m_p) = \begin{bmatrix} \mathbf{x}_2 \\ \cdots\cdots\cdots\cdots \\ \mathbf{f}(\mathbf{X}(j), \mathbf{T}(j), m_p) \end{bmatrix} \tag{5}$$

The nonlinear function $\mathbf{f}(\mathbf{X}(j), \mathbf{T}(j), m_p)$ at the $(k+1)^{th}$ trajectory is expanded using the Taylor series about the k^{th} trajectory. After neglecting the higher order (nonlinear) terms, the following equation is obtained as shown in [3]

$$\mathbf{X}(j+1) = [\Psi x]_j \mathbf{X}(j) + [\Psi T]_j \mathbf{T}(j) + (\Psi \mathbf{m})_j m_p + \Psi_j \tag{6}$$

where the matrices $[\Psi x]_j$, $[\Psi T]_j$, $(\Psi \mathbf{m})_j$, and Ψ_j are

given in [3]. $\mathbf{X}(j+1)$ can be written as a linear combination of the load m_p and the torque control $\mathbf{T}(i)$, i= 1, 2, ..., j. Equation 6 then becomes [3]

$$\mathbf{X}(j+1)=\mathbf{X}_{\Omega}(j+1)+\gamma_j m_p+\sum_{i=1}^{j}[\alpha_{ji}]\,\mathbf{T}(i)\quad j=1,2,\ldots,m \tag{7}$$

where

$$\mathbf{X}(1)_{\Omega}=\mathbf{X}_{\Omega}(t_0) \tag{8}$$

$$\mathbf{X}_{\Omega}(j+1)=[\Psi x]_j\,\mathbf{X}_{\Omega}(j)+\Psi_j \tag{9}$$

$$\gamma_1=(\Psi m)_1 \tag{10}$$

$$\gamma_j=[\Psi x]_j\,\gamma_{j-1}+(\Psi m)_j \tag{11}$$

$$[\alpha_{ji}]=[\Psi x]_j\,[\alpha_{j-1,i}] \qquad (\text{for } i<j) \tag{12}$$

$$[\alpha_{ji}]=[\Psi T]_j \qquad (\text{for } i=j) \tag{13}$$

FORMULATION OF THE OPTIMAL TRAJECTORY PROBLEM

The problem of synthesizing dynamic robot trajectories with maximum load carrying capacities can be formulated as a trajectory optimization problem using the state apace formulation of the robot dynamic equations. By considering point-to-point motions with actuator, joint variables, and deflection constraints throughout the trajectory, the complete formulation can be written to maximize

The DLCC m_p

while ensuring that the state space Equation 3 is satisfied, where the individual joint torques are bounded by

$$\mathbf{T}_{min}(\mathbf{X}(t)) \le \mathbf{T}(t) \le \mathbf{T}_{max}(\mathbf{X}(t)) \tag{14}$$

The bounds $\mathbf{T}_{min}(\mathbf{X}(t))$ and $\mathbf{T}_{max}(\mathbf{X}(t))$ are arbitrary known functions of the actuator joint angles and velocities. In addition to constraints on the joint torques, the initial and final states

$$\mathbf{x}_1(t_0)=\mathbf{x}_{1_0} \quad , \quad \mathbf{x}_1(t_f)=\mathbf{x}_{1_f} \tag{15}$$

must be reached, and it is assumed that

$$\mathbf{x}_2(t_0)=\mathbf{x}_2(t_f)=\mathbf{0} \tag{16}$$

During the motion, the joint displacements and link deflection are also usually bounded by

$$\mathbf{x}^-_1 \le \mathbf{x}_1(t) \le \mathbf{x}^+_1 \tag{17}$$

where x^+_1 and x^-_1 are the upper and lower bounds of the generalized coordinates, respectively. The final constraint is that of the upper bound of the payload which is the smaller value of the SLCC's (Static Load Carrying Capacity) calculated at the two endpositions.

The trajectory synthesis problem formulated above is a nonlinear optimization problem. It is written in a form slightly different from a general optimal control problem where the objective function has an integral form. In Equation 3 the objective function consists of a single variable m_p which is not a function of time. This variable is also a single valued quantity for the entire trajectory. It should be pointed out that m_p is implicitly included in the nonlinear state space equation.

The above formulation of the trajectory optimization problem considered the load as a point mass while the load moment of inertia I_p was not presented to avoid complexity. This assumption is also consistent with robot manufaturers specification of SLCC of a manipulator. With minor modification, the formulation can take into account the moment of inertia of the load. The three principal moment of inertia of the load can be treated as independent variables and implicitly included in the robot dynamic equation. Equation 1 then becomes

$$\tilde{z} = f(\mathbf{X}(j), \mathbf{T}(j), m_p, I_{xxp}, I_{yyp}, I_{zzp}) \quad (18)$$

where I_{xxp}, I_{yyp}, I_{zzp} are the three principal moment of inertia of the load. The problem formulation is exactly the same as before with the exception that the objective function becomes

$$\text{Maximize: } m_p + w_x I_{xxp} + w_y I_{yyp} + w_z I_{zzp} \quad (19)$$

where w_x, w_y, w_z are weighting factors. These factors can be chosen in accordance with the importance of each of the moment of inertias.

SOLUTION TO THE OPTIMAL TRAJECTORY PROBLEM

The Iterative Linear Programming (ILP) method is used to solve the trajectory synthesis problem described above. The upper bound of the load m_p is determined from the SLCC at the two end points. Since the manipulator must completely stop at the end points, the maximum DLCC for the trajectory cannot be greater than the SLCC at either one of the two end positions. This means that

$$m_p \le m_p^{(+)} = \min \{m_{si}, m_{sf}\} \quad (20)$$

Where m_{si} and m_{sf} are the SLCC at the initial and final positions, respectively.

By assumming a typical speed-torque characteristics for DC motors, the torques/speeds characteristic function $\mathbf{T}_{max}(\mathbf{x}(j))$ and $\mathbf{T}_{min}(\mathbf{x}(j))$, the joint torque vector constraints can be approximated as follows

$$\mathbf{T}(j) \ge \mathbf{T}_{min}(\mathbf{x}(j)) = -\mathbf{K}_1 - [\mathbf{K}_2]\mathbf{x}_2(j) \quad (21)$$

$$\mathbf{T}(j) \le \mathbf{T}_{max}(\mathbf{x}(j)) = \mathbf{K}_1 - [\mathbf{K}_2]\mathbf{x}_2(j) \quad (22)$$

where $\mathbf{K}_1$ is an $n \times 1$ constant vector and $[\mathbf{K}_2]$ is an $n \times n$ diagonal constant matrix obtained from the equivalent motor constants. Writing in matrix form leads to

$$-\mathbf{Tp} \le -\mathbf{b}_l = \begin{bmatrix} -\mathbf{K}_1 - [\mathbf{K}_2]\mathbf{x}_2(1) \\ -\mathbf{K}_1 - [\mathbf{K}_2]\mathbf{x}_2(2) \\ \cdots \\ -\mathbf{K}_1 - [\mathbf{K}_2]\mathbf{x}_2(m) \end{bmatrix} \quad (23)$$

$$\mathbf{Tp} \le \mathbf{b}_u = \begin{bmatrix} -\mathbf{K}_1 - [\mathbf{K}_2]\mathbf{x}_2(1) \\ -\mathbf{K}_1 - [\mathbf{K}_2]\mathbf{x}_2(2) \\ \cdots \\ -\mathbf{K}_1 - [\mathbf{K}_2]\mathbf{x}_2(m) \end{bmatrix} \quad (24)$$

where $\mathbf{b}_l$ and $\mathbf{b}_u$ are any known smooth functions of the joint positions and velocities, $\mathbf{Tp} = [\mathbf{T}(1), \mathbf{T}(2), \ldots, \mathbf{T}(m)]$ is an nm vector containing the controls from set point 1 to m. It should be noted that while $m_p \ge o$, $\mathbf{Tp}$ is not restricted in sign. The problem can be converted to a "standard" Linear Programming (LP) problem by changing variables. This can be done by letting in Equation 24

$$\mathbf{Tf} = \mathbf{b}_u - \mathbf{Tp} \qquad \mathbf{Tf} \ge o \quad (25)$$

then $\mathbf{Tp} = \mathbf{b}_u - \mathbf{Tf}$

Substituting Equation 25 into Equation 23 leads to

$$\mathbf{Tf} \le \mathbf{b}_u - \mathbf{b}_l \quad (26)$$

by using Equation 7 the joint variable constraints can be written as follows

$$\mathbf{x}_1^- - \mathbf{x}_{1Q}(j+1) \le \gamma_{1j} m_p + \sum_{i=1}^{j} [\alpha_{1ji}]\mathbf{T}(i) \le \mathbf{x}_1^+ - \mathbf{x}_{1Q}(j+1)$$
$$\text{for } j = 1, \ldots, m \quad (27)$$

where γ_{1j}, $\mathbf{x}_{1Q}(j+1)$ are the upper $n \times 1$ vectors of γ_j and $\mathbf{X}_Q(j+1)$ respectively and $[\alpha_{1ji}]$ is the upper $n \times$

n submatrix of $[\alpha_{ij}]$. Equation 27 can be written in the following form by letting $[\Delta_j] = [\alpha_{1j1}, \alpha_{1j2}, \ldots, \alpha_{1jn}, 0, \ldots, 0]$

$$\gamma_{1j} m_p - [\Delta_j]\mathbf{Tf} \le [\mathbf{x}_1^* - \mathbf{x}_{1\Omega}(j+1)] - [\Delta_j]\mathbf{b}_u \quad \text{for } j = 1, \ldots, m \tag{28}$$

$$-\gamma_{1j} m_p + [\Delta_j]\mathbf{Tf} \le [\mathbf{x}_{1\Omega}(j+1) - \mathbf{x}_1] + [\Delta_j]\mathbf{b}_u \quad \text{for } j = 1, \ldots, m \tag{29}$$

where $[\Delta_j]$ is an n×nm matrix. Finally, from Equation 7 the final state $\mathbf{X}(m+1)$ can be determined as follows

$$\mathbf{X}(m+1) = \mathbf{X}_\Omega(m+1) + \gamma_m m_p + \sum_{i=1}^{m} [\alpha_{mi}]\mathbf{T}(i) \tag{30}$$
$$= \mathbf{X}_\Omega(m+1) + [\Gamma]\mathbf{Tm} = \mathbf{X}(t_f)$$

where

$[\Gamma] = [\gamma_m, [\alpha_{m1}], [\alpha_{m2}], \ldots, [\alpha_{mm}]]$ a $2n \times (nm+1)$ matrix

$\mathbf{Tm} = [m_p, \mathbf{T}(1), \mathbf{T}(2), \ldots, \mathbf{T}(m)]$ a$(nm+1)$ vector

Equation 30 can be written as

$$[\Gamma]\mathbf{Tm} = \mathbf{X}(t_f) - \mathbf{X}_\Omega(m+1) \tag{31}$$

It should be noted that $[\Gamma]$ and $\mathbf{X}_\Omega(m+1)$ are computed based on the values of the state and control varibles of the previous iteration. The only unknown in Equation 31 is the vector **Tm**. In order to facilitate the LP solution, Equation 31 can be written with two sets of inequalities as

$$[\Gamma]\,\mathbf{Tm} - \zeta \le \mathbf{X}(t_f) - \mathbf{X}_\Omega(m+1) \tag{32}$$

$$[\Gamma]\,\mathbf{Tm} + \zeta \ge \mathbf{X}(t_f) - \mathbf{X}_\Omega(m+1) \tag{33}$$

where, $\zeta = [\zeta_{pos}, \ldots, \zeta_{vel}, \ldots, \delta_{pos}, \ldots, \delta_{vel}, \ldots]$ is a $2n_t$ vector, where the elements, ζ_{pos}, ζ_{vel}, δ_{pos}, and δ_{vel} represent the final position error, velocity error, deflection position error, and deflection velocity error tolerances, respectively. This modification introduces four more variables (i. e., ζ_{pos}, ζ_{vel}, δ_{pos}, δ_{vel}) and $2n_t$ inequality constraints. Combining all the constraints and expressing the result in matrix form gives

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \gamma_{1j} & -[\Delta_j] & 0 \\ -\gamma_{1j} & [\Delta_j] & 0 \\ \gamma_m & -[\Gamma] & -1 \\ -\gamma_m & [\Gamma] & -1 \end{bmatrix} \begin{bmatrix} m_p \\ \mathbf{Tf} \\ \zeta \end{bmatrix} \le \begin{bmatrix} m_p^{(s)} \\ \mathbf{b}_u - \mathbf{b}_l \\ \mathbf{x}_1^* - \mathbf{x}_{1\Omega}(j+1) - [\Delta_j]\mathbf{b}_u \\ \mathbf{x}_{1\Omega}(j+1) - \mathbf{x}_1^* + [\Delta_j]\mathbf{b}_u \\ \mathbf{X}(t_f) - \mathbf{X}_\Omega(m+1) - [\Gamma]\mathbf{b}_u \\ \mathbf{X}_\Omega(m+1) - \mathbf{X}(t_f) + [\Gamma]\mathbf{b}_u \end{bmatrix} \tag{34}$$

Since the objective function and the constraints are all linear, there is a standard linear programming problem through which the dynamic load carrying capacity (m_p) can be maximized and simultaneously the error ζ at the points of the trajectory can be minimized.

THE COMPUTATIONAL ASPECTS

The computational procedure for the optimal trajectory involves guessing an initial control and state variable trajectory as shown in Figure 1. This can be achieved by subjecting the manipulator to dual constraints,. i. e., actuator capacity and end effector deformation constraints, when the maximum load is determined. By discretizing the next trajectory into m points, the algorithm for the optimum trajectory in terms of an ILP problem is shown in Figure 1. Equation 9 is satisfied when the terminating criteria are specified as follow

$$\max\{\zeta_{pos}, \ldots, \zeta_{vel}, \ldots, \delta_{pos}, \ldots, \delta_{vel}, \ldots\} \le e_1 \tag{35}$$

$$\max\{\text{abs}[\mathbf{X}^{k+1}(j) - \mathbf{X}(j)^k]\; j = 2, \ldots, m\} \le e_2 \tag{36}$$

$$\text{abs}[m_p^{k+1} - m_p^k] \le e_3 \tag{37}$$

where e_1, e_2, and e_3 are predefined small positive

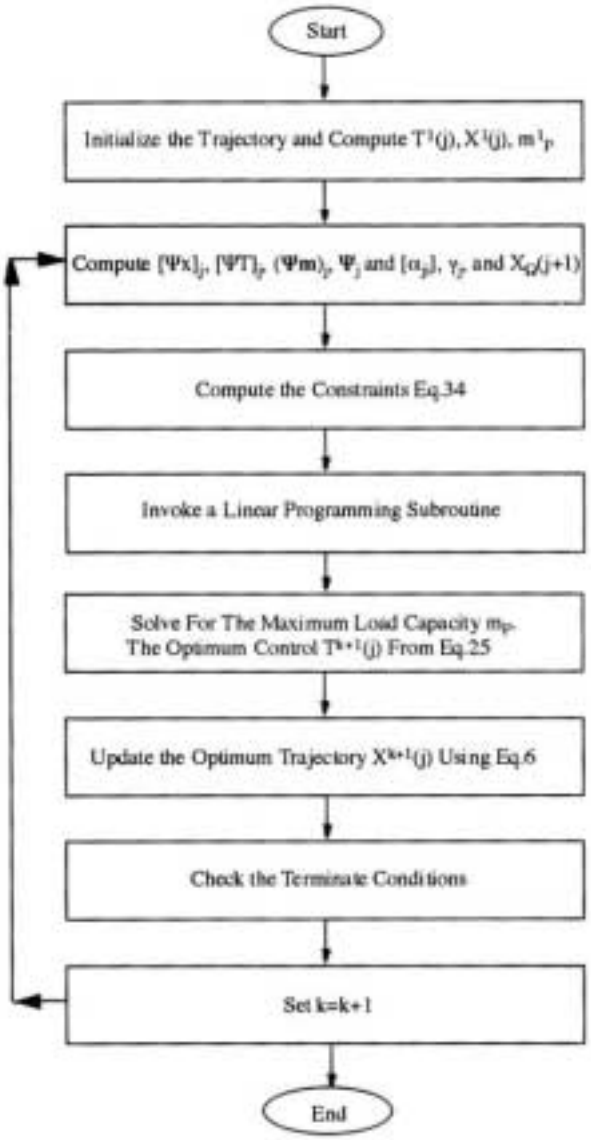

Figure 1. Computational algorithm for two link maniputlators.

constraints. This means that linearization errors are eliminated (or considerably reduced) when the ILP method converges to the optimal solution.

In general, because of the discretization (truncation) error of the difference equation, the continuous state space equation will be stisfied only if the time interval is "sufficiently small." The discretization error can be reduced using the following methods:

1. Use of some higher order difference equation to approximate the continuous differential equation. For example, Equation 9 may be replaced by

$$\mathbf{X}(j+1) - \mathbf{X}(j) = \frac{1}{2}[\Phi(\mathbf{X}(j+1), \mathbf{T}(j+1), m_p) + \Phi(\mathbf{X}(j), \mathbf{T}(j), m_p)]\Omega \qquad (38)$$

2. First solving the problem with a small number of set points and then, using the resulting solution as an initial guess for the problem with a larger number of set points. The additional starting values needed can be obtained by linear or spline interpolation between the old set points. It should also be noted that the round off errors due to the computations may dominate error if the step size (Ω) is too small.

SIMULATION RESULTS AND DISCUSSION

Simulation Conditions

A simulation study was carried out to further investigate the validity and effectiveness of the method and computational aspects presented above. A numerical solution to advance the trajectory synthesis is presented for a two-link flexible manipulator shown in Figure 2. Only link flexibilities are considered while joint compliances are neglected. The bending deflections of links are approximated with clamp free mode shape for each link. Mode shapes are chosen

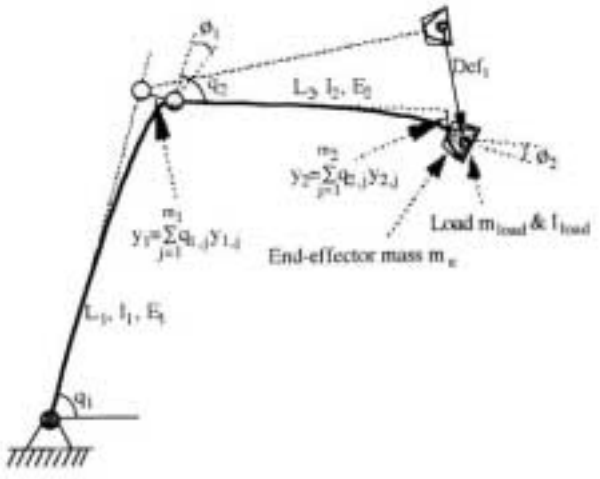

Figure 2. Two- link flexible link manipulator model.

from analytical solution of a Euler-Bernoulli beam eigenfunction analysis. Gravity effect was ignored in this case study in order to isolate the dynamic flexibility effects. The linearized dynamic equations can be rewritten as follows

$$\mathbf{X}(j+1) = [\Psi\, x]_j\, \mathbf{X}(j) + [\Psi\, T]_j\, \mathbf{T}(j) + (\Psi\, m)_j\, m_p + \Psi_j \qquad (39)$$

where the matrices $[\Psi\, T]_j$, $(\Psi\, m)_j$, Ψ_j, and $[\Psi\, x]_j$ are given

$$[\Psi\, x]_j = \begin{bmatrix} 1 & 0 & 0 & 0 & w & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & w & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & w & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & w \\ w\frac{\partial f_1}{\partial q_1} & w\frac{\partial f_1}{\partial q_2} & w\frac{\partial f_1}{\partial \dot{q}_1} & w\frac{\partial f_1}{\partial \dot{q}_2} & w\frac{\partial f_1}{\partial q_{j+1}} & w\frac{\partial f_1}{\partial q_{j+2}} & w\frac{\partial f_1}{\partial \dot{q}_{j+1}} & w\frac{\partial f_1}{\partial \dot{q}_{j+2}} \\ w\frac{\partial f_2}{\partial q_1} & w\frac{\partial f_2}{\partial q_2} & w\frac{\partial f_2}{\partial \dot{q}_1} & w\frac{\partial f_2}{\partial \dot{q}_2} & w\frac{\partial f_2}{\partial q_{j+1}} & w\frac{\partial f_2}{\partial q_{j+2}} & w\frac{\partial f_2}{\partial \dot{q}_{j+1}} & w\frac{\partial f_2}{\partial \dot{q}_{j+2}} \\ w\frac{\partial f_3}{\partial q_1} & w\frac{\partial f_3}{\partial q_2} & w\frac{\partial f_3}{\partial \dot{q}_1} & w\frac{\partial f_3}{\partial \dot{q}_2} & w\frac{\partial f_3}{\partial q_{j+1}} & w\frac{\partial f_3}{\partial q_{j+2}} & w\frac{\partial f_3}{\partial \dot{q}_{j+1}} & w\frac{\partial f_3}{\partial \dot{q}_{j+2}} \\ w\frac{\partial f_4}{\partial q_1} & w\frac{\partial f_4}{\partial q_2} & w\frac{\partial f_4}{\partial \dot{q}_1} & w\frac{\partial f_4}{\partial \dot{q}_2} & w\frac{\partial f_4}{\partial q_{j+1}} & w\frac{\partial f_4}{\partial q_{j+2}} & w\frac{\partial f_4}{\partial \dot{q}_{j+1}} & w\frac{\partial f_4}{\partial \dot{q}_{j+2}} \end{bmatrix}$$

$$[\Psi\, T]_j = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \Omega\frac{\partial f_1}{\partial T_1} & \Omega\frac{\partial f_1}{\partial T_2} \\ \Omega\frac{\partial f_2}{\partial T_1} & \Omega\frac{\partial f_2}{\partial T_2} \\ \Omega\frac{\partial f_3}{\partial T_1} & \Omega\frac{\partial f_3}{\partial T_2} \\ \Omega\frac{\partial f_4}{\partial T_1} & \Omega\frac{\partial f_4}{\partial T_2} \end{bmatrix}; (\Psi\, m)_j = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \frac{\partial f_1}{\partial m_p} \\ \frac{\partial f_2}{\partial m_p} \\ \frac{\partial f_3}{\partial m_p} \\ \frac{\partial f_4}{\partial m_p} \end{bmatrix}; \Psi_j = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \Omega x_1^1 \\ \Omega x_2^1 \\ \Omega x_1^2 \\ \Omega x_2^2 \end{bmatrix}$$

The analytical forms of these matrices are derived for a two-link flexible manipulator by using MATHEMATICA® [7-9]. The numerical values used in the simulation are listed in Appendix I.

Results and Discussions

The algorithm presented here is applicable to robot manipulators of any number of degrees of freedom, as long as the solution of the inverse kinematics can be determined for the entire path. For the purpose of providing effective illustration, the results will be presented for a two link manipulator. The ILP outlined above was applied to a rotary link with a point mass attached to the end. Both links lie on the horizontal plane, and the joint of the first link is located at the origin of the plane coordinate frame. The torque constraints are

$$-\mathbf{K}_1 - [\mathbf{K}_2]\, \mathbf{x}_2(j) \le \mathbf{T}(t) \le \mathbf{K}_1 - [\mathbf{K}_2]\, \mathbf{x}_2(j) \qquad (40)$$

To illustrate the process of determining the bounds on the joint velocities from the known torque/force bounds, the scheme decribed in computational aspects was applied to a two-link manipulator using the above torque constraints. Both links were assumed to have zero velocity initially. In determining the optimal trajectories for maximum allowable load carrying capacity, the end of the second link is constrained initially to move along a given trajectory. Various initial trajectories were simulated and a simple one is prescribed in its parametric form as follows

$$x_4(t) = L + \left(\frac{-t^2}{2} + \frac{t^3}{3}\right), \text{ and } y_4(t) = L + \left(\frac{-t^3}{3} + \frac{t^4}{4}\right) \qquad (41)$$

where L is the link length of both links, a = b = 1, and t ranges from 0 to 1. 05 seconds. Velocities and accelerations are obtained by differentiating the trajectory. The initial conditions for all the generalized coordinates were taken to be zero except that $q_2(0)$ is equal to 90 degrees. The simplification is meant to facilitate the demonstration of the features of the

method. This trajectory was digitised to 30 points. In addition to the torque constraints given above, the following generalized coordinates constraint were imposed.

$$x_i \le x_i(t) \le x_i^+ \qquad (42)$$

In order to initially check the validity of the model, model verification can be performed by comparing the responses of recursive with those of the non-recursive form of the dynamic model [10]. The response of the system was in agreement with them. The open loop system response can be simulated to the following condition $q_1(0) = -1.75$, $q_2(0) = -0.5$, $\tau_1 = \tau_2 = 0$, and all other variables set equal to zero. Clearly, as the flexural rigidity of the links increases, joint variable responses of flexible model should converge to that of rigid model responses. Figure 3 clearly shows that joint variable responses converge to that of the rigid arm case.

Using an algorithm as mentioned above, the optimum trajectory converged after 9 iterations. The results of trajectory optimization are given in Figures 4-7. The optimal joint displacements and joint velocities are plotted in Figure 4. The optimum control trajectory which shows the upper and lower bound of the torque is plotted in Figure 5. The optimal trajectory would involve sudden change of torque at the switching points. Figure 5 shows the bang-bang nature of the control for link 2, and this is reflected in the optimal velocity profile, where the link accelerates in the first half and decelerates in the second half. The simulation results related to optimal trajectory are given in Figure 6. Figure7 gives the LP solution of the maximum load at every iteration. For initial trajectory the m_{load} was forced to be 1.2 Kg while for optimum trajectory it was 2.2 Kg. A load $m_{load} = 2.2$ kg is found to be the maximum load that can be carried in executing the trajectory while not violating either of the constraints.

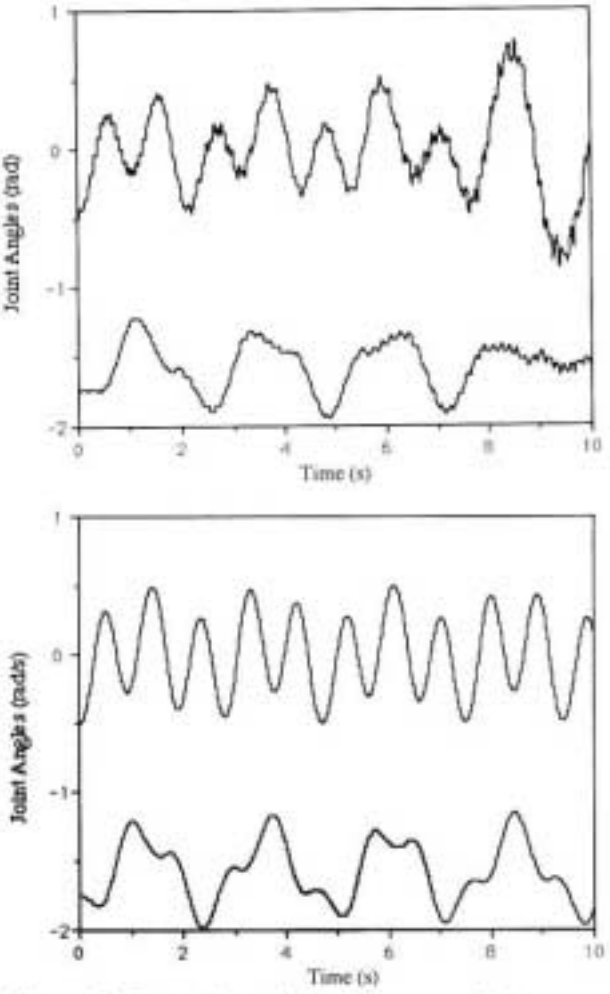

Figure 3. Comparison of joint responses with the non - recursive.

CONCLUSIONS

The main objective of this investigation was to formulate an optimal trajectory and to determine the "maximum load" for flexible manipulators for a given two end points. This was achieved by subjecting the manipulator to dual constraints, that is, actuator capacity and end effector deformation constraints, when the maximum load is determined. In order to be able to control the end effector tracking precision, adding the second constraint was necessary. Whether

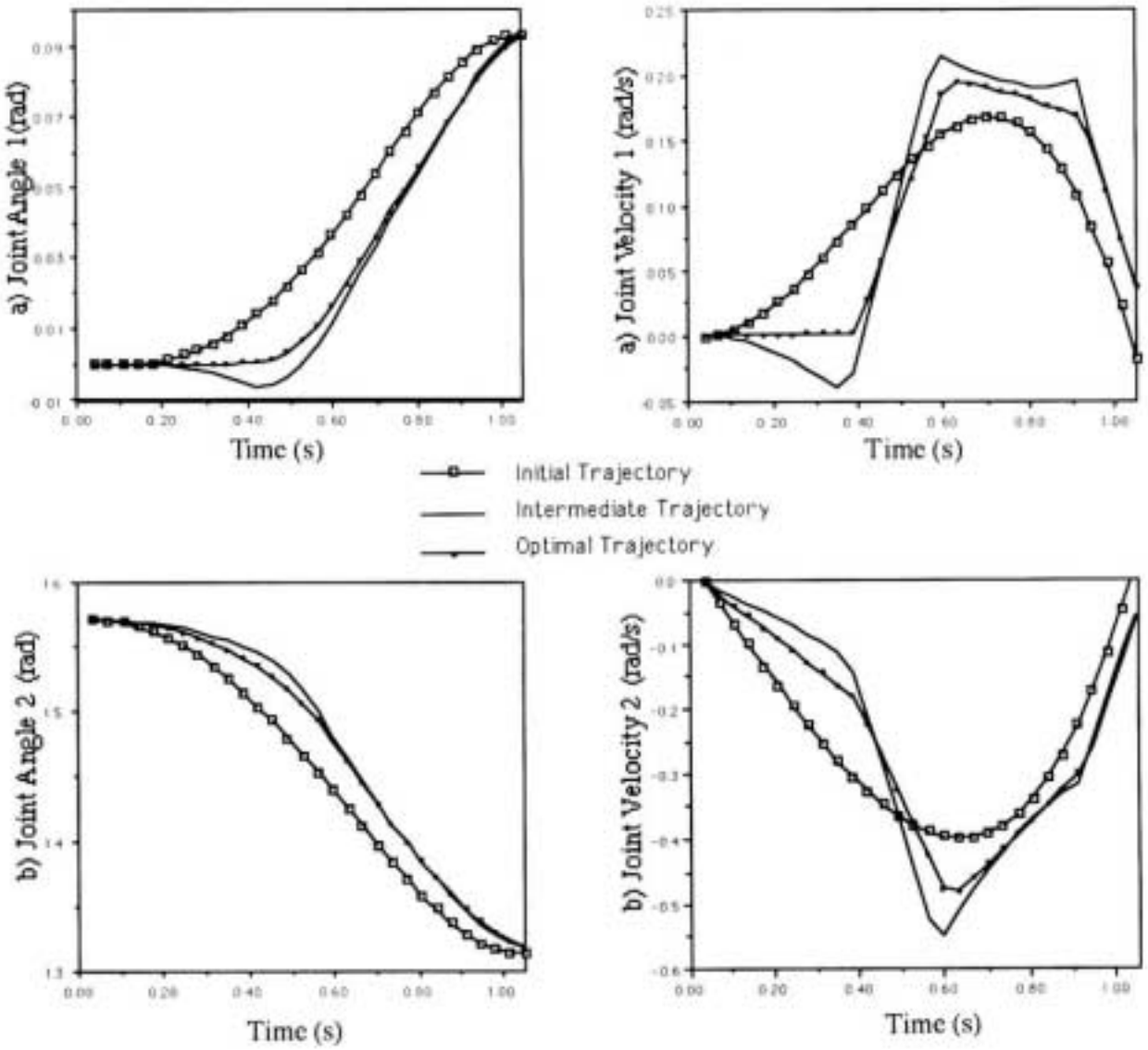

Figure 4. Comparison of the optimal joint positions and velocities with initial positions and velocities.

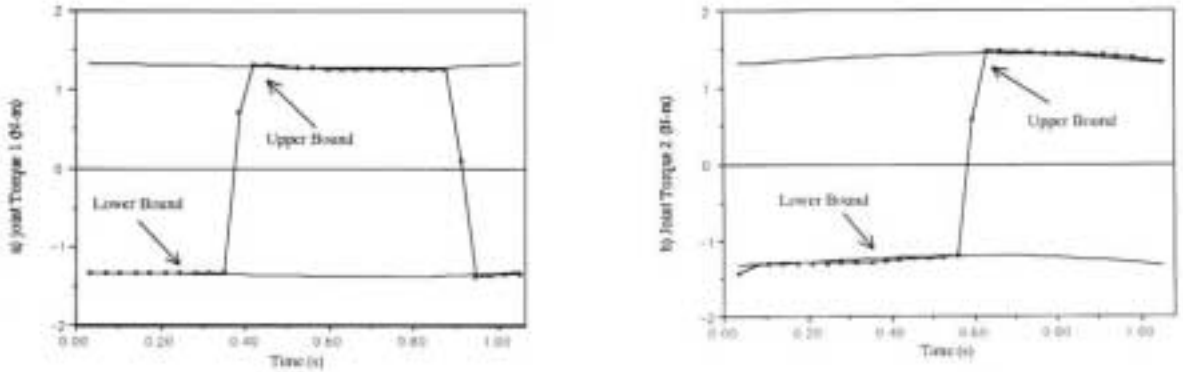

Figure 5. Optimal torque against torque bounds for maximum allowable load.

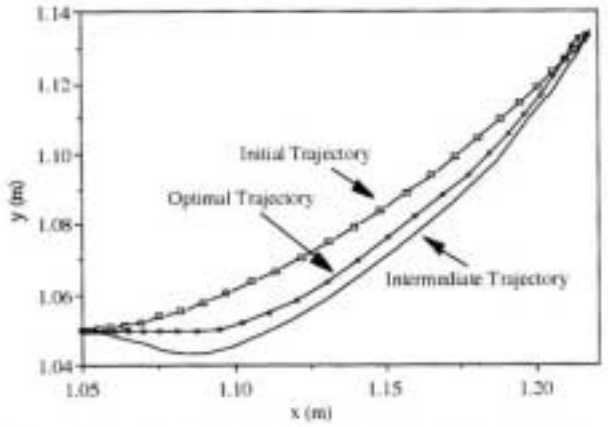

Figure 6. Optimal Cartesian space trajectory for maximum allowable load.

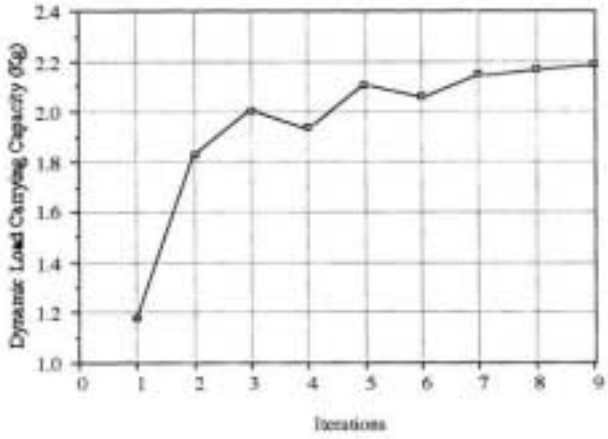

Figure 7. Optimal maximum allowable load computed at each iteration.

the first or the second constraint is more strict depends on the required tracking accuracy. A load $m_{load} = 2.2$ kg was found to be the maximum load that can be carried in executing the trajectory while not violating either of the constraints. The work also shows that in dealing with flexible manipulator dynamics and in determining their optimal trajectories in particular it has greatly benefited from using a symbolic derivation language.

REFERENCES

1. Y. Yao, M. H. Korayem and A. Basu, "Maximum Allowable Load of Flexible Manipulator for a Given Dynamic Trajectory," *International Journal of Robotics and Computer - Integrated Manufacturing*, Vol. 10, No. 4, (1993), pp. 245-251.
2. M. H. Korayem, Y. Yao and A. Basu, "Dynamic Load Carrying Capacity for a Multi - Link Flexible Arm," *Proc. Int. conf. Control and Robotics Automation*, August 4-7, (1992), pp. 79-83.
3. L. T. Wang and B. Ravani, "Dynamic Load Carrying Capacity of Mechanical Manipulators-Part II: Computational Procedure and Applications," *Transactions of ASME, Journal of Dynamic System, Measurement and Control*, Vol. 110, March (1988), pp. 53-61.
4. E. Bayo and B. Paden, "On Trajectory Generation for Flexible Robots," *Journal of Robotic System*, (1987), pp. 229-235.
5. M. A. Serna and E. Bayo, "Experiments on the Payload-Adaptation of a Flexible One-Link Manipulator with Unknown Payload," *Proc. IEEE Int. Conf. Robotics and Automation*, (1990), pp. 910-913.
6. W. J. Book, "Recursive Lagrangian Dynamics of Flexible Manipulator Arm", *International Journal of Robotics Research*, Vol. 3, No. 3, (1984), pp. 87-101.
7. S. Wolfram, **MATHEMATICA**, Addison- Wesley, (1991).
8. R. E. Maeder, **Programming in Mathematica**, Addison-Wesley, (1990).
9. M. H. Korayem, Y. Yao, and A. Basu, "Application of Symbolic Manipulation to Inverse Dynamics and Kinematics of Elastic Robots," *International Journal of Advanced Manufacturing Technology*, Vol. 12, No. 3, (1994).
10. S. Cetinkunt and W. J. Book, "Symbolic Modeling and Dynamic Simulation of Robotic Manipulators with Compliant Links and Joints," *Robotics & Computer Integrated Manufacturing*, Vol. 5, No. 4, (1989), pp. 301-310.

APPENDIX I

Numerical Values for simulation.

Parameter	Value	Unit
Young's Modulus	$E_1 = E_2 = 2.06 \times 10^{11}$	N/m^2
Area Moment of Inertia	$I_1 = I_2 = 9.9 \times 10^{-12}$	m^4
Link Length	$L_1 = L_2 = 1.05$	m
Link Linear Mass Density	$\mu_1 = \mu_2 = 4.05 \times 10^{-1}$	kg/m
Actuator Constants	$k_1 = 0.63$ & $k_2 = 0.18$	N-m & N-m/rad respectively
Stall Torque	$\tau_s = 0.63$	N-m
No Load Speed	$w_0 = 3.5$	rad/sec
Initial Joint Angles	$q_1(0) = 0$ & $q_2(0) = 90$	degree
Mass of End Effector	$m_e = 0.1$	kg