OPTIMAL DESIGN OF ROBOTIC MANIPULATOR TRAJECTORIES: A Nonlinear Programming Approach

Mark L. Nagurka and **Vincent Yen**

CMU-RI-TR-87-125

Department of Mechanical Engineering and
The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213

April 1987

Partial support for this work was provided by the
National Science Foundation under grant DMC-86-02847.

Table of Contents

List of Figures

Abstract

A nonlinear programming approach for the optimal motion planning of robotic manipulators is presented. In this approach a standard optimal control problem of infinite dimensionality (in time) is converted into an optimization problem of finite dimensionality by approximating the manipulator trajectories by the sum of a polynomial and a set of appropriate eigenfunctions. The optimal control problem is then solved via a nonlinear programming numerical algorithm given a performance index which is a continuous (explicit or implicit) function of time. This method can be used for trajectory planning of high degree-of-freedom manipulators once motion specifications and constraints have been identified. Examples of trajectory planning of a planar robotic manipulator with free and constrained final time and states are presented..

1 Introduction

The problem of controlling a robotic manipulator can be conveniently divided into two closely related subproblems: (i) trajectory planning (also called motion planning), and (ii) trajectory tracking (also called motion control.) For instance, a possible strategy for controlling a manipulator consists of off-line trajectory planning followed by on-line trajectory tracking, the latter usually involving the implementation of closed-loop feedback. Here, the term *trajectory* refers to the time history of displacement, velocity, and acceleration of each degree-of-freedom of a manipulator model. This research focuses on a methodology for suboptimal *(Le.,* near optimal) trajectory planning for unconstrained as well as constrained manipulator motions.

Schemes for trajectory planning generally "interpolate" or "approximate" the desired path by a class of polynomial functions. These schemes then generate a sequence of time-based *control set points* for the control of the manipulator from the initial location to its destination. Quite often, there exists a number of possible trajectories between the two given endpoints. *(In* theory, there exists an infinite number of possible trajectories.) For instance, the manipulator can be moved along a straight-line path that connects the endpoints (straight-line trajectory), or the manipulator can be moved along a smooth, polynomial trajectory that satisfies the position and orientation constraints at both endpoints (joint-inteipolated trajectory). The research reported here exploits this potential of a multiplicity of possible solutions by developing an off-line optimal motion planning algorithm that generates trajectories that minimize a given performance index without violating any constraints. This trajectory generation algorithm can be formulated as an optimal control problem.

In solving optimal control problems, it is typical to apply variational methods to derive the necessary conditions for optimality which can be formulated as two-point boundary-value problems (2PBVPs). Numerical algorithms have been developed to solve some 2PBVPs that are analytically intractable [1,2], Although these algorithms have been applied to solve some optimal control problems, they are inadequate computationally to solve for the optimal control of systems, such as robotic manipulators, that have large numbers of degrees of freedom and strong noniinearities.

In view of these numerical difficulties, various approaches have been suggested for the optimal motion programming of robotic manipulators. For example, by linearizing the manipulator dynamics at the final target point, Kahn and Roth [3] formulated a near-minimum-time control law for open kinematic chains- Vukobratovic and Kircanski [4] used a dynamic programming based method to calculate the optimal velocity profile for a prespecified manipulator path. By neglecting the influence of Coriolis and centrifugal forces, Vukobratovic and Kircanski [5] also applied optimal control theory to solve for the optimal motion of "simplified^{11} robotic models. Kim and Shin [6] presented a suboptimal control approach for manipulators with a weighted minimum time-fuel criterion based on the concept of averaging the dynamics at each sampling interval. Although these approaches have been tested via computer simulations, their success is limited. Each approach is either confined to a problem with a particular type of performance index or it depends upon a simplified dynamic model which may be valid only in a special case.

1 Introduction

The problem of controlling a robotic manipulator can be conveniently divided into two closely related subproblems: (i) trajectory planning (also called motion planning), and (ii) trajectory tracking (also called motion control.) For instance, a possible strategy for controlling a manipulator consists of off-line trajectory planning followed by on-line trajectory tracking, the latter usually involving the implementation of closed-loop feedback. Here, the term *trajectory* refers to the time history of displacement, velocity, and acceleration of each degree-of-freedom of a manipulator model. This research focuses on a methodology for suboptimal *(i.e.*, near optimal) trajectory planning for unconstrained as well as constrained manipulator motions.

Schemes for trajectory planning generally "interpolate" or "approximate" the desired path by a class of polynomial functions. These schemes then generate a sequence of time-based *control set points* for the control of the manipulator from the initial location to its destination. Quite often, there exists a number of possible trajectories between the two given endpoints. (In theory, there exists an infinite number of possible trajectories.) For instance, the manipulator can be moved along a straight-line path that connects the endpoints (straight-line trajectory), or the manipulator can be moved along a smooth, polynomial trajectory that satisfies the position and orientation constraints at both endpoints (joint-interpolated trajectory). The research reported here exploits this potential of a multiplicity of possible solutions by developing an off-line optimal motion planning algorithm that generates trajectories that minimize a-given performance index without violating any constraints. This trajectory generation algorithm can be formulated as an optimal control problem.

In solving optimal control problems, it is typical to apply variational methods to derive the necessary conditions for optimality which can be formulated as two-point boundary-value problems (2PBVPs). Numerical algorithms have been developed to solve some 2PBVPs that are analytically intractable [1,2]. Although these algorithms have been applied to solve some optimal control problems, they are inadequate computationally to solve for the optimal control of systems, such as robotic manipulators, that have large numbers of degrees of freedom and strong nonlinearities.

In view of these numerical difficulties, various approaches have been suggested for the optimal motion programming of robotic manipulators. For example, by linearizing the manipulator dynamics at the final target point, Kahn and Roth [3] formulated a near-minimum-time control law for open kinematic chains. Vukobratovic and Kircanski [4] used a dynamic programming based method to calculate the optimal velocity profile for a prespecified manipulator path. By neglecting the influence of CorioUs and centrifugal forces, Vukobratovic and Kircanski [5] also applied optimal control theory to solve for the optimal motion of "simplified" robotic models. Kim and Shin [6] presented a suboptimal control approach for manipulators with a weighted minimum time-fuel criterion based on the concept of averaging the dynamics at each sampling interval. Although these approaches have been tested via computer simulations, their success is limited. Each approach is either confined to a problem with a particular type of performance index or it depends upon a simplified dynamic model which may be valid only in a special case.

Townsend, *et al.*, [7] and Schmitt, *et* a/., [8] presented conceptually similar, but alternative, approaches for solving for the optimal motion of manipulators. In both approaches each joint angular displacement is approximated by a function. In [7] this function consists of a sum of a polynomial and a half-range cosine scries; in [8] it consists of a sum of a cubic polynomial and a sequence of known functions with unknown weighting coefficients- The optimization problem then involves finding the parameters of the approximating functions that minimize a performance index*

In practice, these approaches are suboptimal Only finite terms of the expansion functions (i.e., the half-range cosine series in [7] and the sequence of known functions in [8]) are included, whereas, in theory, the optimal solution requires infinite terms. Nevertheless, these approaches appear useful in solving several types of optimal motion problems. However, in applying these methods a number of unanswered issues are raised.

1. Convergence. Can it be guaranteed that the suboptimal trajectory converges to the optimal solution, or if this is not possible, that the suboptimal performance index* at least, converges to the optimal performance index?
2. Polynomial function. Can the (minimum) degree and coefficients of die polynomial function be specified such that convergence is guaranteed?
3. Boundary Conditions. Can various types of boundary conditions* such as free, fixed, and coupled final conditions, be treated?
4. Criterion of Optimally. Can a criterion of optimaEty be identified that can be used to ensure the quality of the suboptimal solution?
5. Applicability. What, if any, limitations exist in applying such su'bcptiina! approaches? Fee instance, can such approaches be used to solve bang-bang control problems?

In response to the above questions, this research develops a general purpose Fotuiar-btsed suboptimal control algorithm *to* generate manipulator trajectories* TMs algoritftm approximates die time history of each generalized coordinate by the sum of t fifth cwkr *pAywmM* and a finite term Fourier-type series. Instead of finding the continuous time history of the control variables, the proposed method reduces the optimal control problem to one of scareMiig for the optimal parameters of the approximating functions, the optimal values of any ftee boondaiy conditions, and the optimal final time (if it is not fixed). Due to the nature of the conversion* die computational scheme of this method is based cm in inverse dynamic approach and therefore avoids most of die numerical difficulties eocauntcred in optimal control problems. By using standard nonEneai* programming techniques, the detomiriarioii of Ac optimal motion for high order, aonlfiear robotic manipulator models is hence feasible*

Unlike previous schemes* this method does not require model amplification and cm be applied to a large class of optimal control problems. Problems with variable final time as well as problems with free or constrained final states can be handled directly. In addition* a pMeinc that can be used to *cmExm* the quality of the suboptimal trajectory is suggested

This paper is organized is follows, In Section 2 the methodology tad the function of various types of trajectory planning algorithms are considered- Section 3 is concerned with the formulation and numerical difficulties of optimal control problems, for dynamic systems such as robotic manipulators. The main contributions of this research are presented in Sections 4 and 5. In Section 4, the Fourier suboptimal control approach is developed and in Section 5, some important characteristics of the approach are discussed. Computer simulation results that demonstrate the application and effectiveness of the proposed algorithm are presented in Section 6. Conclusions are given in Section 7.

2 Planning Manipulator Trajectories

In manipulator programming it is typical to view the trajectory planner as a black box. Usually, the inputs are the path specifications, where the path is defined as the space curve along which the manipulator end-effector moves from the initial to final location (position and orientation). The planner also accepts constraint information such as obstacle constraints (whether there are any obstacles present in the path) and dynamic constraints (whether there are any limitations on the generalized forces). The outputs of the trajectory planner are the histories of the joint trajectories and generalized forces.

There are two common approaches for planning manipulator trajectories. One approach requires the user to explicitly specify a *set* of constraints at selected locations, called interpolation points, along the trajectory. The trajectory planner then selects a parameterized trajectory from a class of functions that "interpolates" and satisfies the constraints at the interpolation points. A second approach requires explicit specification of the path that the manipulator traverses by an analytical function, such as a straight-line path in Cartesian coordinates* The trajectory planner then generates a trajectory to approximate the desired path.

In the above two trajectory generation approaches it is desirable to provide simple trajectories that are smooth, accurate, and efficient (in terms of computational requirements and in terms of manipulator performance such as energy consumption.) A fast computation time for generating the sequence of control set points along the desired trajectory is preferred especially for cases of on-line implementation. However, current trajectory planners usually do not account for interaction between the trajectory and controller and for dynamic constraints. As a result, large tracking errors may develop. Another drawback of current trajectory planners is that they lack an objective index to evaluate the trajectory performance.

Recently, the design of trajectory planners has shifted away from a real-time planning objective to an off-line planning phase in order to generate trajectories that can accommodate more constraints and achieve better system performance. For example, Lee [9] proposed an off-line approach in which a trajectory planning problem was formulated as a maximization of the straight-line distance between two consecutive Cartesian set points subject to smoothness and torque constraints.

In essence, this new trend decomposes the control of robotic manipulators into off-line trajectory planning followed by on-line tracking control Running off-line, a sophisticated trajectory planner should be able to (i) generate a trajectory that satisfies path specifications and various types of constraints, and (ii) achieve a trajectory with superior performance (measured by an objective function, *i.e.*, performance index). These goals led to the development of the trajectory planning algorithm presented in this paper.

3 Manipulator Optimal Control

In practice, optimal control approaches have not been implemented widely for programming trajectories of manipulators due to the nonlinear nature and high dimensionality of such systems. As mentioned in the Introduction, the necessary conditions for optimality based on standard optimal control theory lead to a two-point boundary-value problem (2PBVP).

Various numerical techniques have been proposed to solve the 2PBVP. In general, these techniques fall into two categories: gradient-based methods and dynamic programming methods. The utility of the gradient-based methods is limited due to their dependence on gradient-type information which is quite sensitive to numerical errors. The applicability of the dynamic programming methods is hindered by dimensionality problems (*i.e.*, the number of computations as well as storage requirements typically grow much faster than the order of the system.)

In addition to optimal control methods, nonlinear programming methods represent an important class of optimization techniques. The main difference between solving an optimal control problem and a nonlinear programming problem is the dependence on time (a continuous variable). In an optimal control problem one seeks the time history of an optimal trajectory, which in theory consists of an infinite number of points. In a nonlinear programming problem, one searches for a finite number of free variables to optimize a given objective function, where the objective function and constraints are time-independent. In order to bridge the difference between the nonlinear programming and the optimal control methods, two different approaches have been proposed, namely, the Rayleigh-Ritz technique and the method of finite difference. The basic idea of these two methods is to reduce the optimal control problem of infinite dimensionality into a problem of finite dimensionality which can be solved by nonlinear programming methods.

The method of finite difference discretizes the time history of the generalized coordinate into a finite number of piece-wise continuous intervals. The problem is thus changed into a problem of finding the extrema of the objective function with the values of the piece-wise continuous generalized coordinate as free variables. This technique is generally impractical when the degrees-of-freedom or the time interval of interest becomes large since the number of variables increases significantly under such circumstances.

The basis of the Rayleigh-Ritz method is to replace each generalized coordinate by a set of weighted known functions. The unknown weighting coefficients are determined such that the performance index of the original problem can be minimized. The K term approximation can be expressed as

$$\theta(t)=\sum_{k=1}^{K} c_k u_k(t) \tag{1}$$

Here the generalized coordinate variable, θ, consists of a sum of the product of weighting constant, c^ and known approximating function, u_k. If the desired trajectory is specified on one or both of the boundaries, the approximating functions should be constructed in such a way that the given conditions will be satisfied for all values of the weighting constants. If the boundary conditions are natural *(i.e.,* free) boundary conditions, no such special precaution is required. Usually, the number of constants required depends on the complexity of the optimization problem and the shrewdness in selecting the approximating functions. In general, best results are obtained when using approximating functions drawn from a functionally complete set of eigenfunctions.

Two drawbacks of the Rayleigh-Ritz method can be identified First, it is difficult to determine a set of approximating functions that simultaneously satisfies the boundary conditions on both the generalized coordinates and their time derivatives. Second, even if the approximating functions converge to the optimal solution, there is no guarantee that the respective derivatives will converge. The approach presented in the following section generalizes the Rayleigh-Ritz method and corrects for these problems.

4 Fourier-Based Suboptimal Control Algorithm

The dynamic equations of motion of a rigid-body manipulator model are a coupled set of nonlinear ordinary differential equations describing the dynamic behavior of the manipulator. These equations can be derived by a variety of approaches, such as the Newton-Euler, Lagrange-Euler, and generalized D'AIembert formulations. For an n degree-of-freedom manipulator configured as an open kinematic chain the equations can be expressed in the form

$$I(O-M(I(O)e\underline{\ddot{(}}r)-h\underline{V}(I(r),\underline{\dot{9}}(r))+\underline{G}(\underline{0}(r)) \tag{2}$$

where $\underline{8}$ is an n x 1 vector of generalized coordinates associated with the n degrees-of-freedom of the manipulator, $\underline{r}$ is an n x 1 vector of generalized forces applied at the joints, $\underline{A/}$ is an n x n inertial-mass matrix, $\underline{V}$ is an n x 1 vector representing centrifugal and Coriolis effects, G is an n x 1 gravity loading torque vector, t is time, and superscript dot represents time derivative. In general, each element of $\underline{M}$ and $\underline{G}$ is a ^complicated function which depends on $\underline{9}$(r> , while each element of $\underline{V}$ depends on both $\underline{B}(t)$ and $\underline{B}(t)$.

Given the equations of motion of a manipulator model, two types of dynamic problems can be solved In the direct dynamic problem, the generalized force history is specified and the equations of motion can be integrated to obtain the motion trajectories of the manipulator. In the inverse dynamic problem, the desired generalized coordinates and their rates are assumed known a *priori, e.g.,* from a trajectory planning program, and the equations of motion are used to compute the generalized force history.

Numerically, the inverse dynamic approach is much more straightforward than the direct dynamic approach. In the direct dynamic approach, integration of the differential equations of motion is required, while in the inverse dynamic approach the same set of equations is used as a system of algebraic equations. This distinction is important from the perspective of computational efficiency and has implications when considering the accumulated numerical error. Both truncation and roundoff errors significantly influence the convergence of standard optimal control algorithms. Consequently, computational algorithms that are based on a direct dynamic approach generally have serious convergence problems in searching for optimal solutions of high order, nonlinear systems.

The optimal control problem is to find an admissible control, $\underline{T}_{opt}$, that causes the manipulator to follow an admissible trajectory, $\underline{\theta}_{opt}$ and $\underline{\dot{\theta}}_{opt}$, such that the performance index,

$$J(\underline{T}(t)) = f(\underline{\theta}(t_f), \underline{\dot{\theta}}(t_f), t_f) + \int_{t_o}^{t_f} g(\underline{\theta}(t), \underline{\dot{\theta}}(t), \underline{T}(t), t)\,dt \tag{3}$$

is minimized. In equation (3), f and g are general functions of the arguments shown and it is assumed that the initial conditions, $\underline{\theta}(t_o)$ and $\underline{\dot{\theta}}(t_o)$, and the initial time, t_o, are specified. The final time, t_f, and the final states, $\underline{\theta}(t_f)$ and $\underline{\dot{\theta}}(t_f)$, can either be free or fixed.

The problem can be generalized further by adding two types of constraints. The first class of constraints, state variable inequality constraints, can be written as

$$\underline{E}(\underline{\theta}(t), \underline{\dot{\theta}}(t), t) \leq \underline{0} \tag{4}$$

where $\underline{E}$ is an m x 1 $(m < n)$ vector function of the states and possibly time. In the trajectory planning problem, these constraints are usually due to obstacles that must be avoided in the working environment. The second class of constraints, actuator-related inequality constraints, can be expressed as

$$|T_i| \leq \tau_i \ , \qquad i=1,\dots,n \tag{5}$$

where τ_i is the maximum allowable torque at the i-th joint. These constraints reflect the fact that each joint actuator is power limited and subject to saturation.

The central concept of the proposed suboptimal algorithm is to convert the optimal control problem into a nonlinear programming problem by approximating each joint angular displacement by the sum of a fifth order polynomial and a finite Fourier-type series. For example, for joint i,

$$\theta_i(t) = P_i(t) + F_{ki}(t), \tag{6}$$

where the auxiliary polynomial, $P_i(t)$, is defined as

$$P_i(t) = p_{i0} + p_{i1}t + p_{i2}t^2 + p_{i3}t^3 + p_{i4}t^4 + p_{i5}t^5 \tag{7}$$

and the K term Fourier-type series is defined as

$$F_{ki}(t) = \sum_{k=1}^{K} a_{ik} \cos\frac{k\pi(t-t_o)}{(t_f-t_o)} + \sum_{k=1}^{K} b_{ik} \sin\frac{k\pi(t-t_o)}{(t_f-t_o)} \tag{8}$$

The velocity and acceleration of the *i*-th joint are obtained by direct differentiation of the above equations. This approach is adopted for the *n* joints, and then the control variables, *i.e.*, the generalized forces, are calculated readily from the equations of motion. Finally, the performance index is computed using a straightforward numerical integration method such as Simpson's composite integral technique.

Assuming that both the initial and final conditions of the state variables (joint displacements and velocities) are given, the coefficients of the fifth order auxiliary polynomial are computed to satisfy the following algebraic equations for each joint *i*:

$$\theta_i(t_o)=P_i(t_o)+F_{ki}(t_o) \tag{9}$$

$$\theta_i(t_f)=P_i(t_f)+F_{ki}(t_f) \tag{10}$$

$$\dot{\theta}_i(t_o)=\dot{P}_i(t_o)+\dot{F}_{ki}(t_o) \tag{11}$$

$$\dot{\theta}_i(t_f)=\dot{P}_i(t_f)+\dot{F}_{ki}(t_f) \tag{12}$$

$$\ddot{\theta}_i(t_o)=\ddot{P}_i(t_o)+\ddot{F}_{ki}(t_o) \tag{13}$$

$$\ddot{\theta}_i(t_f)=\ddot{P}_i(t_f)+\ddot{F}_{ki}(t_f) \tag{14}$$

The search for the optimal trajectory, which in theory consists of an infinite number of points, is thus converted into a nonlinear programming problem with a finite number of free variables. These variables are the Fourier-type coefficients, a_{ik} and b_{ik}, the free boundary conditions of the trajectory, including the initial accelerations, $\ddot{\theta}_i(t_o)$, and final accelerations, $\ddot{\theta}_i(t_f)$, and the final time, if it is not fixed.

The necessity of the fifth order auxiliary polynomial can be justified by the definition of the Fourier series and its property of differentiability. The following theorem can be found in standard engineering mathematics textbooks such as [10].

Theorem of Dirichlet: If $X(t)$ is a bounded periodic function, $X(t) = X(t + 2\alpha)$, which in any one period has at most a finite number of local maxima and minima and a finite number of points of discontinuity, then the Fourier series of $X(t)$ converges to $X(t)$ at all points where $X(t)$ is continuous and converges to the average of right- and left-hand limits of $X(t)$ at each point where $X(t)$ is discontinuous.

The conditions of the Theorem of Dirichlet, usually referred to as the Dirichlet conditions, make it clear that a function need not be continuous in order to possess a valid Fourier expansion. The implication is that it is reasonable to expect that every optimal trajectory can be approximated by a Fourier series since such a trajectory satisfies the Dirichlet conditions.

It is next necessary to show that the suboptimal solution converges to the optimal solution. To do this, the following property is first introduced.

<u>*Property ofDifferentiability*</u>*:* The necessary and sufficient conditions for $\dot{X}(t)$ » $\dot{F}(t)$ in the interval [8,5+2a] where *F(t)* is *X(t)'s* Fourier series and *X(t)* is continuous are (i) *X(t)* is continuous, (ii) *X(t)* is piece-wise diffeientiable in (5, 5+2a), (iii) *X(5)* = *X($+2a)*, and (iv) $\dot{X}$(5j = $\dot{X}$(8+2aJ. This property can be generalized to the second derivative case.

According to this property, it can be concluded that as long as the above four conditions are true, the result of term by tenn differentiation of the Fourier series of period 2a representing *X(t)* in the interval of [5,5+2<x] converges to *X(t)* at each point in [8, 8+2a] at which *X(t)* is continuous. A proof can be found in [11]. Thus, it can be shown that equations (9) through (14) guarantee the feasibility of direct differentiation of the Fourier series from displacement to velocity and from velocity to acceleration as long as they are all continuous. The convergence of the suboptimal trajectory (in terms of displacement, velocity and acceleration) to the optimal solution is thus guaranteed

5 Discussion of Suboptimal Approach

This section discusses some of the detailed characteristics, including the restrictions and strengths* of the proposed approach.

Local Minimum* The method guarantees only that a local minimum solution is achieved. The suboptimal solution may not be unique. Identification of the global optimal solution may require trial-and-error selection of the initial guess.

Numerical Algorithm. The original optimal control problem has been converted to a problem of ordinary extrema which can be solved by a number of well-developed nonlinear programming techniques [12-16]. For example, the Simplex method [16] is adopted in this research.

Accuracy. A closed foim expression of the joint variables is available and, thus, the joint torques can be computed directly from the equations of motion by straightforward algebra. The accuracy of these calculations is limited only by the least significant digit of the computer. As a result, the accuracy of this approach is dominated by the numerical error of the integration algorithm used to the evaluate the performance index. Because the errors of numerical integration ran be estimated and controlled, the problem of convergence which is encountered frequently in implementing standard optimal control approaches is avoided.

Final States and Timt. The final states and final time (if it is not fixed) can be treated as free variables. During the search for the optimal solution, they - together with the coefficients of the Fouriff-type series — are adjusted simultaneously in every iteration to minimize the performance index.

Rtstrictloiis. One of the necessary conditions for the convergence of the proposed approach is the continuity of displacement, velocity and acceleration of the optimal trajectory.

This requirement is violated in bang-bang control problems due to the finite jump(s) of the control variables. Hence, the suboptimal trajectory does not converge to the optimal trajectory at the switch point. However, since bang-bang control has a finite number of switch points, the value of the suboptimal performance index converges to the value of the performance index of the optimal solution.

Although in theory the proposed approach is capable of achieving optimal performance, simulation results show that the speed of convergence of the suboptimal bang-bang control solution is usually very slow. This property is similar to the "Gibbs' phenomenon" ([10], pp. 247-249) which occurs when developing the Fourier series for a square wave function. As a consequence, when high accuracy is desired (say an error in the performance index of less than 5 percent), the number of Fourier-type expansion functions increases dramatically such that the approach may become computationally impractical. In spite of this drawback, the proposed approach can always provide a smooth trajectory except where there is an instantaneous jump of the control input (a situation which is physically impossible).

Criterion for Optimality. A possible means of verifying the quality of the suboptimal control law is to check if it satisfies the necessary conditions for optimality which are derived by variational methods. In practice, this verification can be carried out by substituting the suboptimal solution into an appropriate standard optimal control numerical algorithm and determining if the termination criterion of the selected algorithm is satisfied.

An alternative empirical approach is to append another term of the series to the previous solution and repeat the optimization process. Additional terms can be added, on a term by term basis, until the change in the performance index is sufficiently small. (For unconstrained problems, simulation results show that a two or three term Fourier-type expansion yields satisfactory results.) It should be noted that although it is a reasonable idea to use the previous solution as part of the initial guess of the current optimization process, one cannot fix the preceding terms of the Fourier type series and only treat the newly appended terms as free variables. This is because a Fourier series with finite terms is only optimal in the sense of mean square error. That is, the coefficients determined by the Fourier formulas are the optimal coefficients only in terms of the mean square error between the original function and the finite term Fourier series. The proposed algorithm minimizes the algebraic difference between the true and suboptimal performance indices which is mathematically different from finding a suboptimal trajectory to minimize the mean square error of the true optimal trajectory.

Controller Design. The dynamic equations that describe the manipulator motion are coupled, highly nonlinear, ordinary differential equations. The control system design is complicated by the coupling and nonlinearity (due physically to gravitational torques, reaction torques, and Coriolis and centrifugal torques.) As a result, these effects are often carefully studied in the process of control system design. For instance, if the coupling inertias between joints are small with respect to the effective joint inertias, the manipulator can be treated as independent mechanical systems and the complexity of the control law can be greatly reduced. Another example is a manipulator not moving at high speed, for which the velocity dependent terms are typically neglected, thereby making the implementation of various real-time control laws possible. The simplifications mentioned in these examples are often adopted but limit the operating domain of the manipulator controller.

The interaction among various terms of the dynamic equations is determined not only by the physical characteristics of the manipulator and the load it carries but also by the trajectory. It is possible to search for trajectories that give rise to minimal nonlinear effects and/or minimal dynamic coupling between joint motions so that (in theory) simplified control strategies such as linear control theory and/or decoupled feedback control schemes can be applied. One of the objectives of this research is to explore this approach of selecting special trajectories such that the control system design problem can be simplified.

6 Examples

Example 1. The dynamic system of interest is the two degree-of-freedom (*i.e.*, planar) robotic manipulator shown in Figure 1. If acceleration due to gravity acts in a direction perpendicular to the *x*-*y* plane, the equations of motion are:

$$T_1=H_{11}\ddot{\theta}_1+H_{12}\ddot{\theta}_2-H\dot{\theta}_2^2-2H\dot{\theta}_1\dot{\theta}_2 \qquad (15)$$

$$T_2=H_{12}\ddot{\theta}_1+H_{22}\ddot{\theta}_2+H\dot{\theta}_1^2 \qquad (16)$$

where the coefficients of the joint angular rates are the effective mass moments of inertia given by:

$$H_{11}=M_1d_1^2+I_1+M_2[D_1^2+d_1^2+2D_1d_1\cos\theta_2]+I_2$$

$$H_{22}=M_2d_2^2+I_2$$

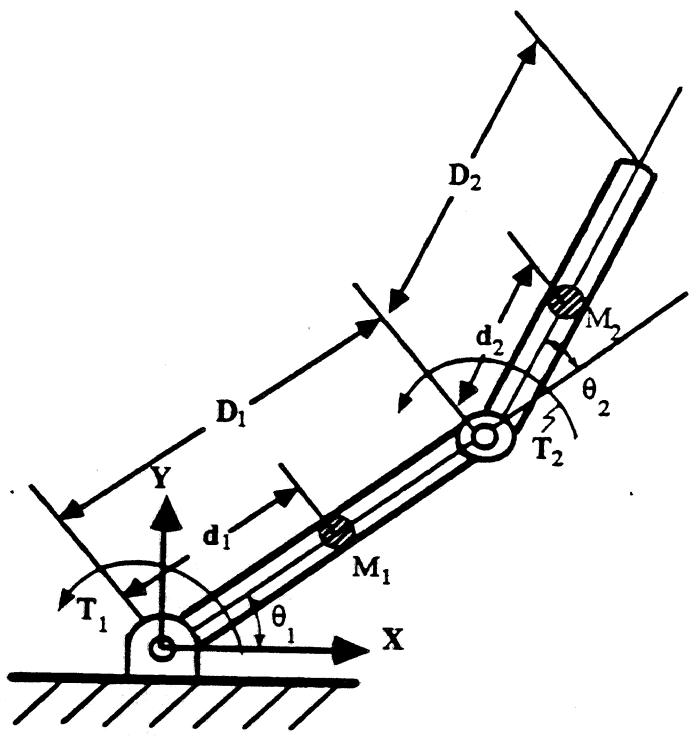
$$H_{12}=M_2D_1d_2\cos\theta_2+M_2d_2^2+I_2$$

$$H=M_2D_1d_2\sin\theta_2$$

Here the mass of link i is M_i, the centroidal mass moment of inertia of link i is I_i, the center of mass of link i is located on the centerline passing through adjacent joints at a distance d_i from joint i, and the length of link i is D_i. In equations (15) and (16), the first two terms on the right represent inertial torques, while the third term represents the effective centrifugal torque. The fourth term in equation (15) represents the effective Coriolis torque.

The path specification is to move the manipulator from initial position $[\theta_1(0), \theta_2(0)] = [0°, 30°]$ to final position $[\theta_1(t_f), \theta_2(t_f)] = [120°, 60°]$ with the initial and final velocity zero. If the final time, t_f, is known, a simple trajectory planning method is to introduce a cubic polynomial for each joint angular displacement. A cubic polynomial has four coefficients, which can be found from the boundary requirements on initial and final displacement and velocity.

Alternatively, if a performance index representing certain performance characteristics of the manipulator is proposed, the optimal trajectory can be obtained by applying the suboptimal method presented in this paper. In applying the method, each joint displacement is approximated by the sum of a fifth order polynomial and a Fourier-type series containing two terms. In this



$M_1 = M_2 = 1\text{kg} \quad I_1 = I_2 = 0.1\text{kg-m}^2$

$D_1 = D_2 = 1\text{m} \quad d_1 = d_2 = 0.5\text{m}$

Figure 1: Two Link Planar Manipulator Model.

example, the performance index is the control effort, represented by the integral of the sum of the joint torques squared:

$$J=\int_0^{t_f}(T_1^2 + T_2^2)\,dt \tag{17}$$

Two different optimization problems are investigated. In the first problem, the final time is fixed (*i.e.*, t_f = 1 second specified) and the suboptimal trajectories with and without state constraints are calculated. Here, the state constraints refer to the requirement that joint displacements must fall within the range defined by the initial and final values. In the second problem, the final time is not fixed but is constrained (*i.e.*, t_f cannot exceed 2 seconds). The suboptimal solution assuming no state constraints is obtained. In both these problems, the Simplex method [16] is used to search for the optimal values of the coefficients of the series and the initial and final accelerations for both joints (with the cubic polynomial trajectory used as the initial guess.) The state constraints are included in the performance index using a penalty function on their violation.

For the first problem, in which the final time is fixed, the time histories of the joint displacements and the performance index integrand are graphed in Figures 2 and 3, respectively, for the cubic polynomial trajectory, the state constrained suboptimal trajectory, and the state unconstrained suboptimal trajectory. These figures show that the cubic polynomial and the state constrained suboptimal solutions remain within the range defined by the two endpoint displacements, whereas the state unconstrained suboptimal solution involves displacements of the joints that deviate outside the boundaries. The state constrained suboptimal trajectory for joint 2 approaches its upperbound much faster than the cubic polynomial trajectory.

Although it appears from Figure 2 that the state unconstrained suboptimal trajectory is excessive, in fact the trajectory is compact. Multiple exposure schematic drawings of the robot executing the cubic polynomial trajectory and the state unconstrained suboptimal trajectory are shown in Figure 4 (at 0.1 second intervals). Furthermore, the control effort of the state unconstrained suboptimal solution remains small during the motion (Figure 3). In contrast, the cubic polynomial and state constrained suboptimal solutions require control efforts that are especially large near the boundaries. Integration of the curves of Figure 3 shows that the value of the performance index decreases from 387.5 N^2-m^2-sec for the cubic polynomial solution to 341.8 N^2-m^2-sec for the state constrained optimal solution and to 122.9 N^2-m^2-sec for the state unconstrained suboptimal solution. In summary, substantially improved performance is possible by relaxing the constraints on the joint displacements, constraints which are usually imposed (implicitly or explicitly) in trajectory generation methods.

For the second problem, in which the final time is not fixed but constrained, the time histories of the joint displacements and the performance index integrand for the suboptimal solution are shown in Figures 5 and 6, respectively, for the cases of fixed and constrained final times. In these figures the curves corresponding to the suboptimal solution with fixed final time

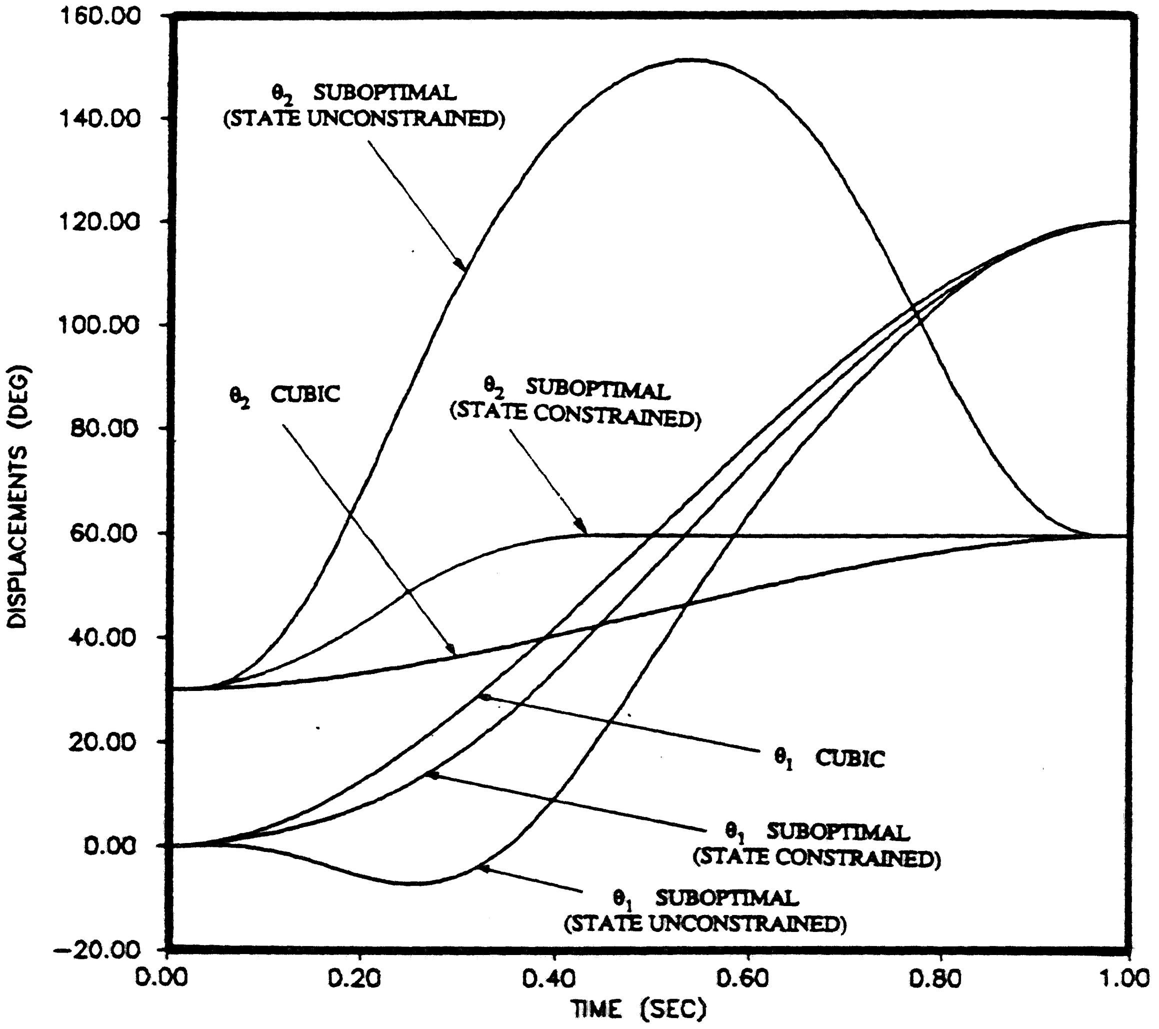


Figure 2: Joint Displacement Histories for Example 1 (Fixed Final Time).

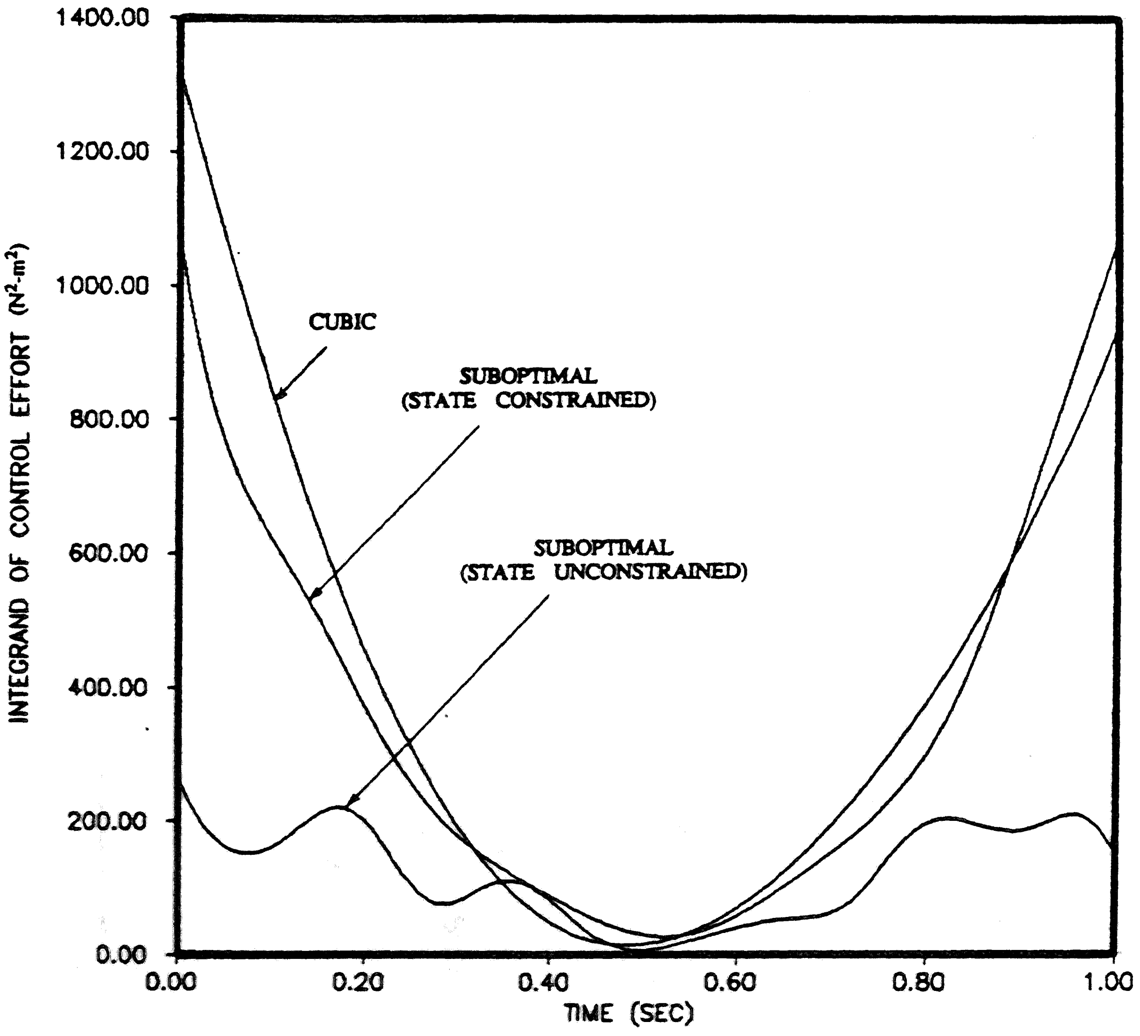
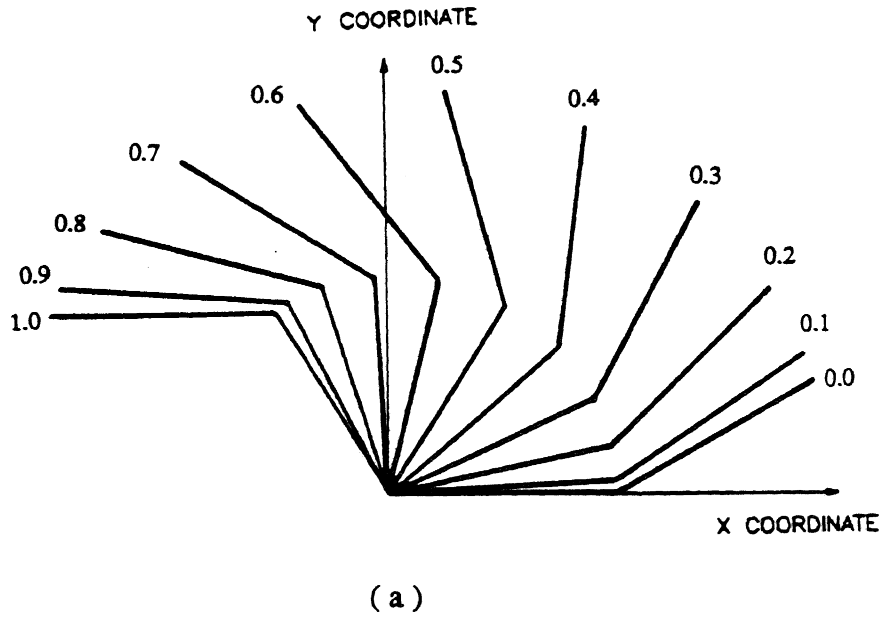


Figure 3: Integrand of Control Effort (Sum of Joint Torques Squared) for Example 1 (Fixed Final Time).



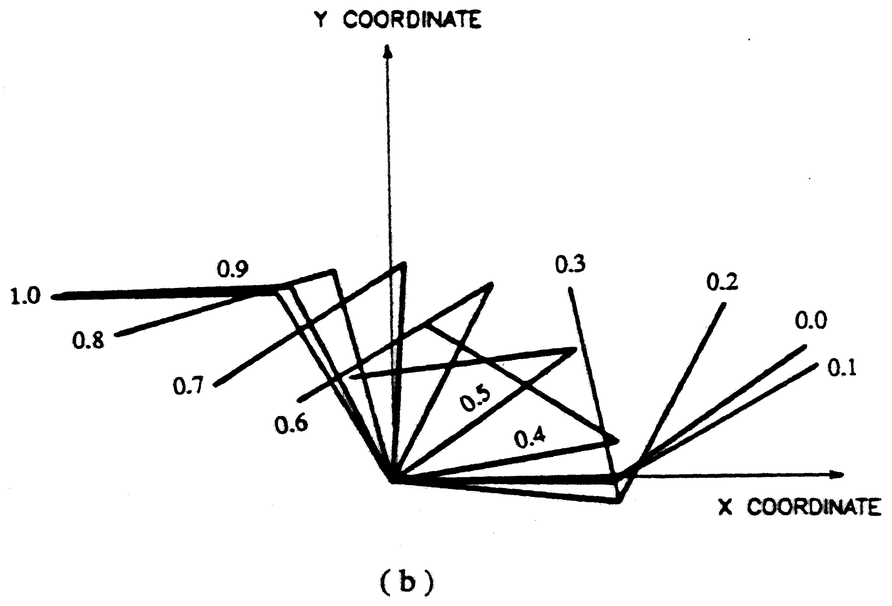


Figure 4: Multiple Exposure Schematic Drawings of Robot Executing (a) Cubic Polynomial Trajectory, and (b) State Unconstrained Suboptimal Trajectory (at 0.1 sec intervals).

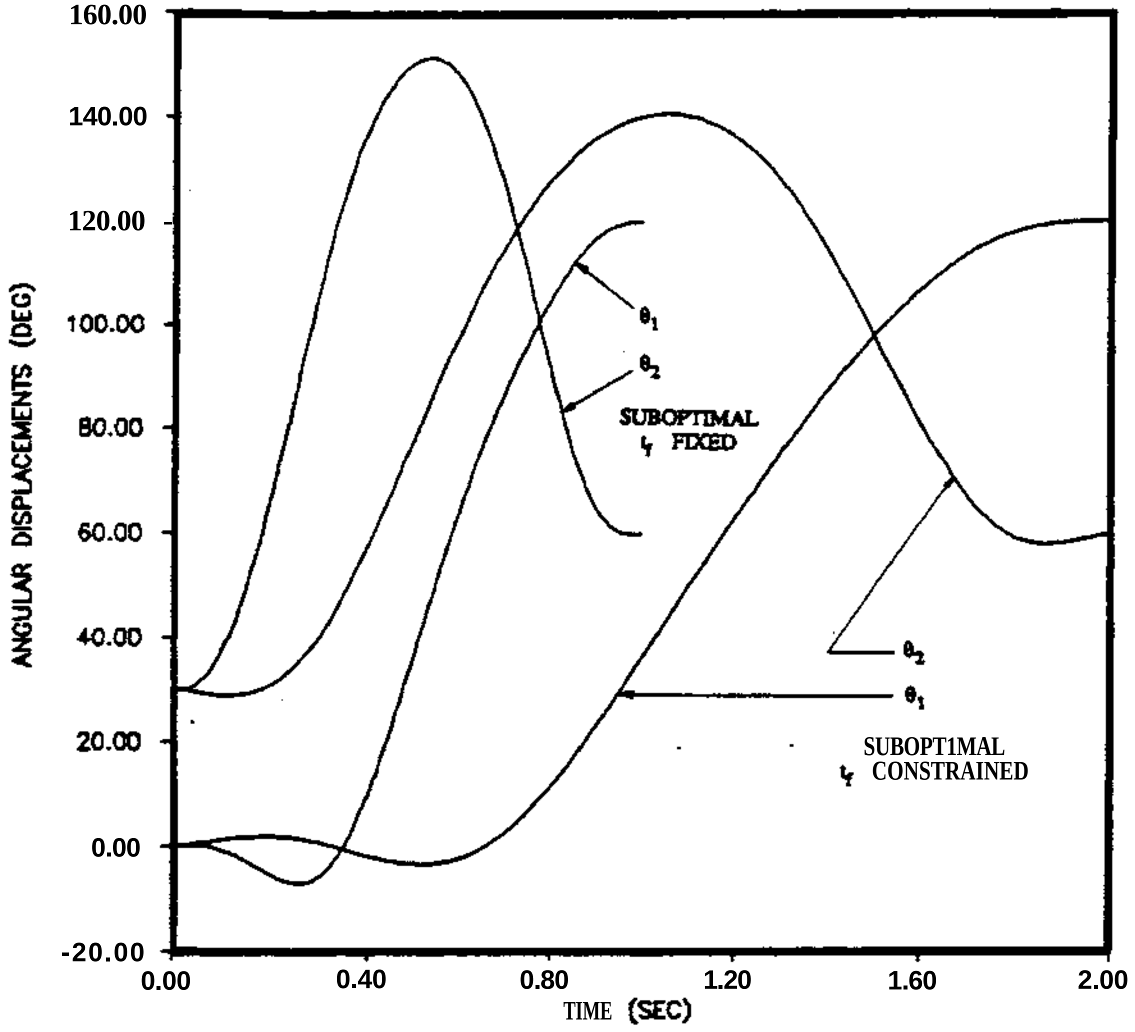


Figure 5: Joint Displacement Histories for Example 1 (Free and Constrained Final Tame).

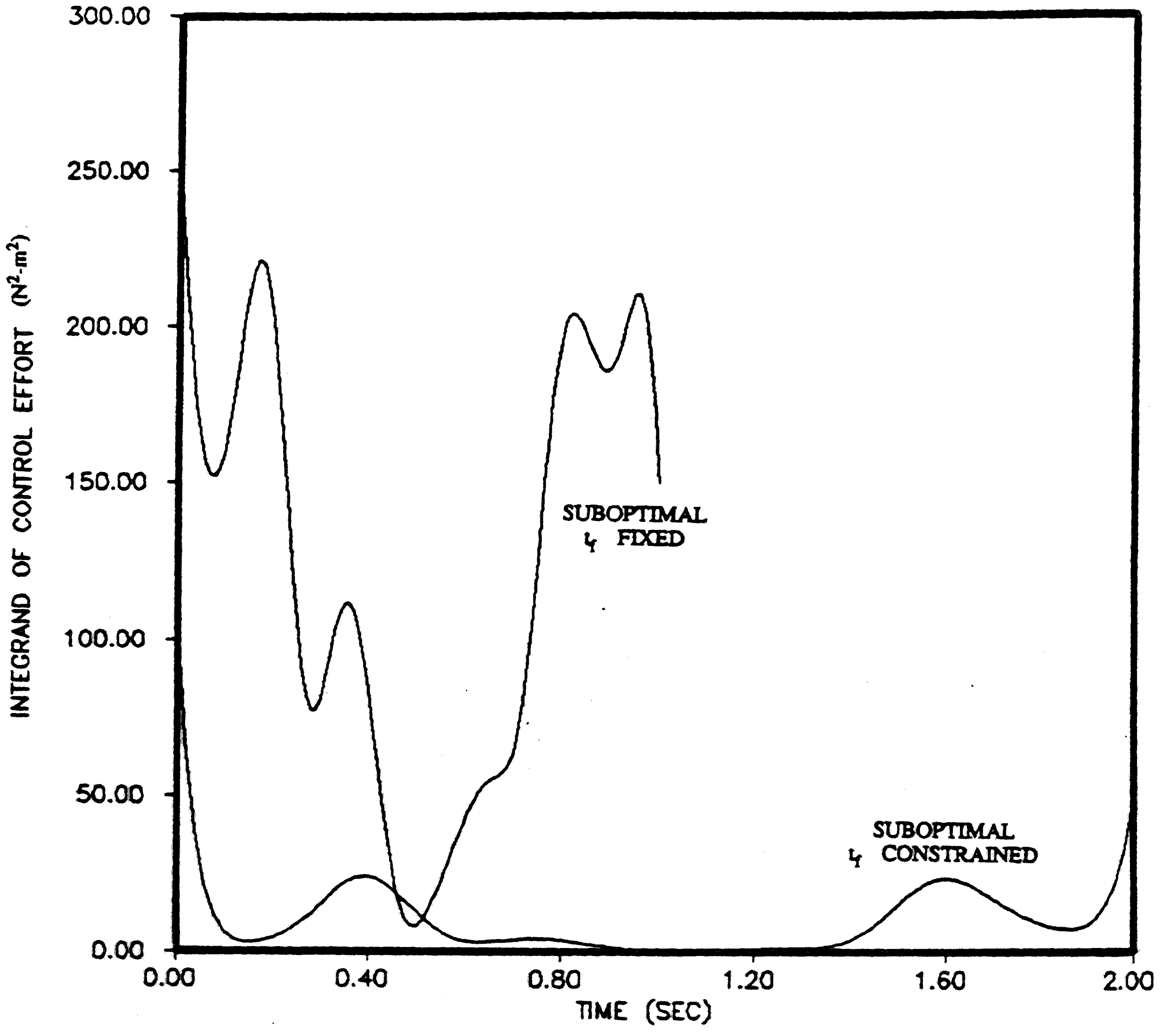


Figure 6: Integrand of Control Effort (Sum of Joint Torques Squared) for Example 1 (Fixed and Constrained Final Time).

are identical to the curves in Figures 2 and 3. The optimal final time coincides with the upperbound of the constraint, i.e., *tj* = 2 second. The displacement trajectories for the constrained time problem exhibit similar shapes to those of the fixed time problem, but are stretched out to fill the full time available- As a result, there is a reduction in the extrema which occur outside the range of the boundary conditions.

From the integration of the curves of Figure 6, the value of the performance index decreases from 122,9 N^2~m^2-sec to 20.7 N^2-m^2-sec when releasing the fixed constraint arid imposing an inequality constraint on the final time. This significant improvement reflects the effect of the final time on system performance.

Example *2.* The simulation results of the previous example suggest that a major improvement in performance in terms of decreased control effort is possible by implementing the (state unconstrained) suboptimal strategy. This example is designed to highlight the advantages of applying the proposed strategy for shaping the trajectory dynamics as desired. In this discussion, the manipulator model and path boundary conditions (with t_f = 1 second fixed) are the same as in the first example.

First consider using the cubic polynomial trajectory as the planned trajectory in a scheme, shown in Figure 7, to generate an actual trajectory. The scheme concatenates the inverse dynamics of a simplified model with the direct dynamics of a Ml dynamic model. (That is, the actual generalized forces, $\underline{T}$*, which are calculated according to the equations of motion of a prespecified simplified model, are used in the integration of the equations of motion of a complete dynamic model to determine the planned trajectory.) The simplified model is a reduced version of the full dynamic model in which one or more effects have been neglected. Clearly, the planned and actual trajectories will be different, especially as more important effects (i.e., nonlinearities* coupling, *etc.)* are neglected In this study, two simplified models are examined One model neglects the Coriolis effect; a second model neglects both Coriolis and centrifugal effects.

Figure 8 compares the planned and actual displacements for the two simplified models. Although in general the trendwise character of the graphs agree, errors in the final boundary conditions exist for both simplified models. Furthermore, the actual displacement of the second joint deviates significantly firom the planned motion* suggesting that the centrifugal component of acceleration is important for the cubic polynomial trajectory.

It is possible to embed the open-loop scheme of Figure 7 in a closed-loop algorithm that attempts to minimize the differences between the planned and actual trajectories. Such a scheme is shown in Figure 9, where the proposal suboptimal framework described in this paper is adopted. The performance index consists of functions of the errors and error rates at the final time and an integral of the errors and *error* rates during the trajectory. The intent of the closed-loop scheme is fo drive the actual trajectory to the planned trajectory (while satisfying the boundary conditions), thereby minimizing the influence of those effects which are neglected in the simplified model

By using such a scheme, it is exported that the simplified model will satisfactorily simulate the real dynamic model in die neighborhood of the proposed "optimal[1]" trajectory. If this is true,

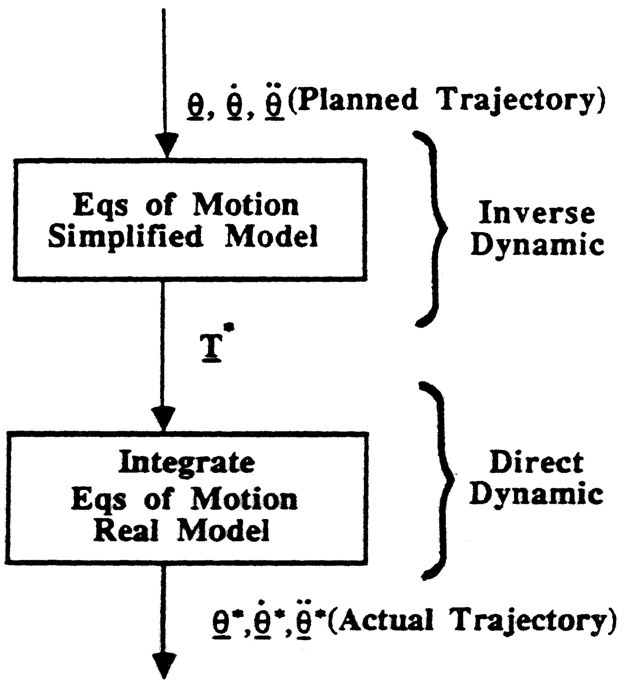

$\underline{\theta}, \dot{\underline{\theta}}, \ddot{\underline{\theta}}$ (Planned Trajectory)
Eqs of Motion Simplified Model
Inverse Dynamic
$\underline{T}^*$
Integrate Eqs of Motion Real Model
Direct Dynamic
$\underline{\theta}^*, \dot{\underline{\theta}}^*, \ddot{\underline{\theta}}^*$ (Actual Trajectory)

Figure 7: Flowchart of Open-Loop Algorithm for Generating Actual Trajectory from Cubic Polynomial Planned Trajectory.

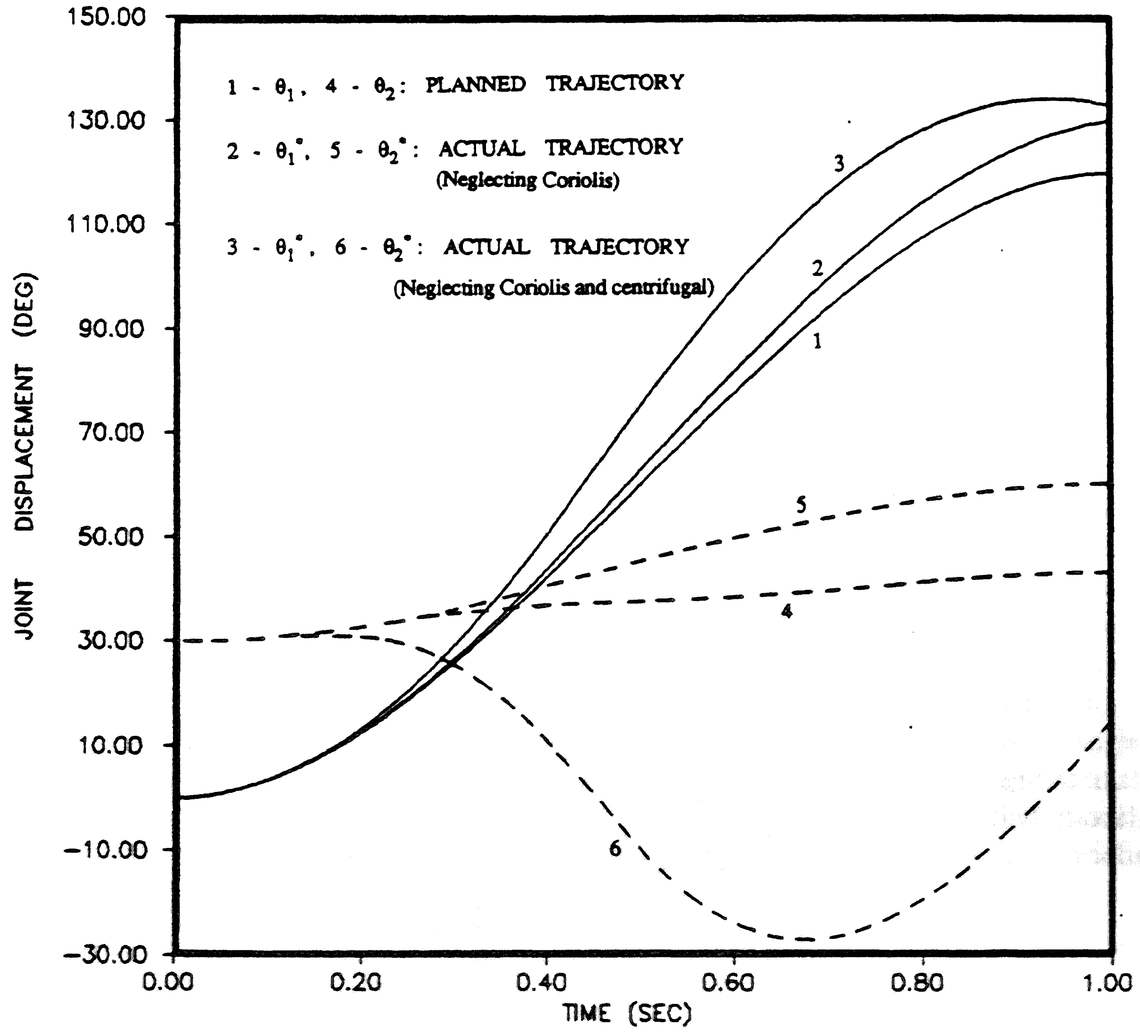


Figure 8: Cubic Polynomial Planned and Actual Displacement Histories for Example 2 (from Figure 7.)

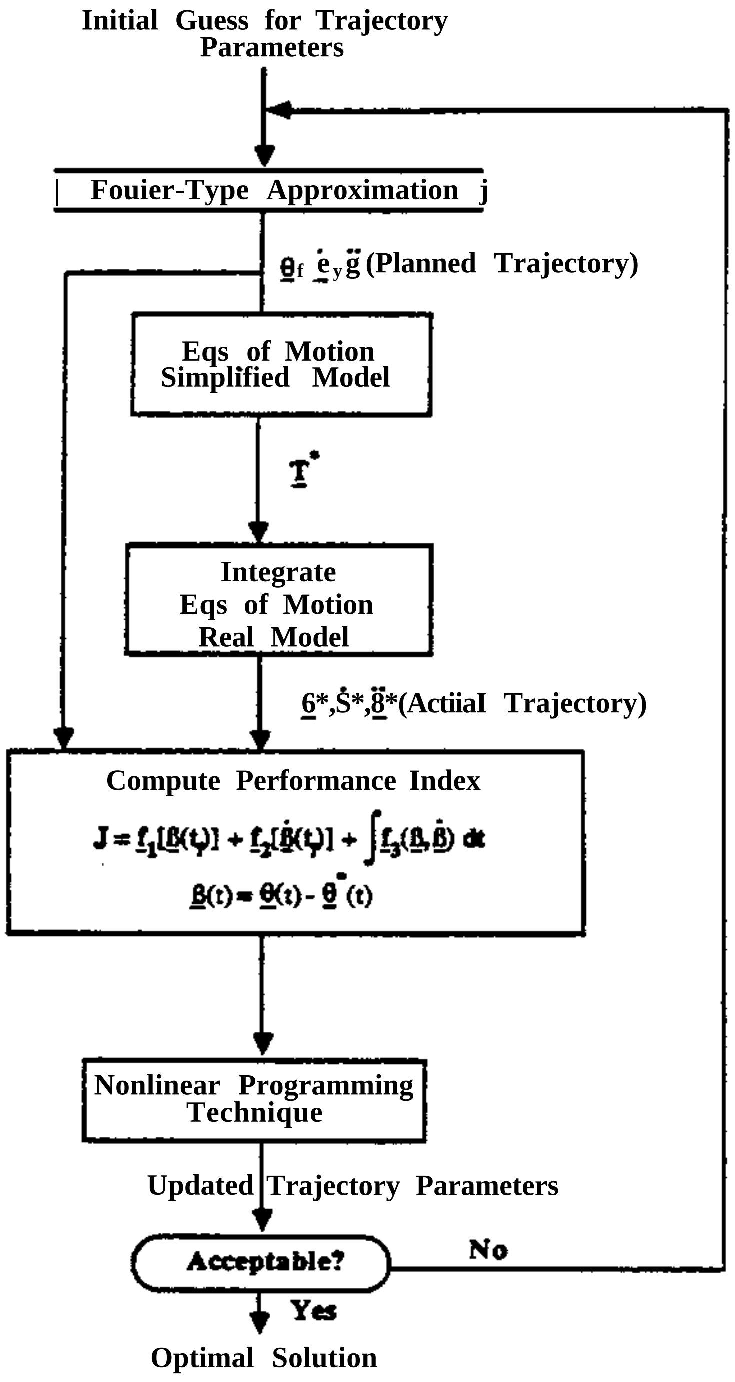

Initial Guess for Trajectory Parameters
Fouier-Type Approximation
θ, θ̇, θ̈ (Planned Trajectory)
Eqs of Motion Simplified Model
T*
Integrate Eqs of Motion Real Model
θ*, θ̇*, θ̈* (Actual Trajectory)
Compute Performance Index
J = f1[β(tf)] + f2[β̇(tf)] + ∫f3(β, β̇) dt
β(t) = θ(t) - θ*(t)
Nonlinear Programming Technique
Updated Trajectory Parameters
Acceptable?
No
Yes
Optimal Solution

Figure 9: Flowchart of Closed-Loop Algorithm for Generating Saboptimal Planned and Actual Trajectories-a simplified feedback controller that accounts only for the dynamics of the simplified model should be able to effectively regulate the dynamic response of the manipulator when moving along the corresponding optimal trajectory. However, the sensitivity of the resulting optimal trajectory to disturbances requires further investigation in order to determine the actual effective operating range of such a simplified controller.

In this example, the performance index is assumed to be:

$$J = 100[\theta_1(1)-\theta_1^*(1)]^2 + 100[\theta_2(1)-\theta_2^*(1)]^2 + \int_0^1 [(\theta_1-\theta_1^*)^2 + (\theta_2-\theta_2^*)^2]dt \qquad (18)$$

Again, two simulation cases are studied. In one case, the actual torques are generated based on a simplified model that neglects the Coriolis effect. In a second case, the simplified model neglects both Coriolis and centrifugal effects. In both cases, the cubic polynomial solution is taken as the initial guess of the planned trajectory.

Figure 10 displays the resulting suboptimal trajectories, *i.e.*, the planned and actual joint displacement histories for the two cases. The figure shows that the tracking error for the first simulation case (in which only Coriolis is neglected) is so small that the curves of the planned and actual trajectories coincide. Note that the minimization of the Coriolis term is achieved by making the velocity of each joint approximately equal to zero during part of the trajectory. For the second simulation case (in which both Coriolis and centrifugal effects are neglected), the tracking error is observable in the figure, *i.e.*, a small difference between the planned and actual trajectories exists. The implication is that the trajectory for which the manipulator dynamics appears to have no contributions of Coriolis and centrifugal effects cannot be reached fully. However, the actual trajectory comes very close to matching the planned trajectory for which these effects are absent. Finally, it should be mentioned that the boundary conditions of the trajectories for both cases are satisfied, since the performance index has a strong penalty on their violation.

In summary, this example demonstrates that the manipulator dynamics can be influenced strongly by the trajectory. By adopting a "smart" trajectory, it is suggested that the effectiveness of simplified controller designs may be increased.

7 Conclusion

A basic problem in robotics is planning motions to solve some prespecified task, and then controlling the manipulator as it executes the commands necessary to achieve those actions.

This paper presents a general-purpose suboptimal trajectory generation algorithm for robotic manipulators. The proposed approach is a Fourier-based method that converts an optimal control problem into a nonlinear programming problem. The algorithm is especially

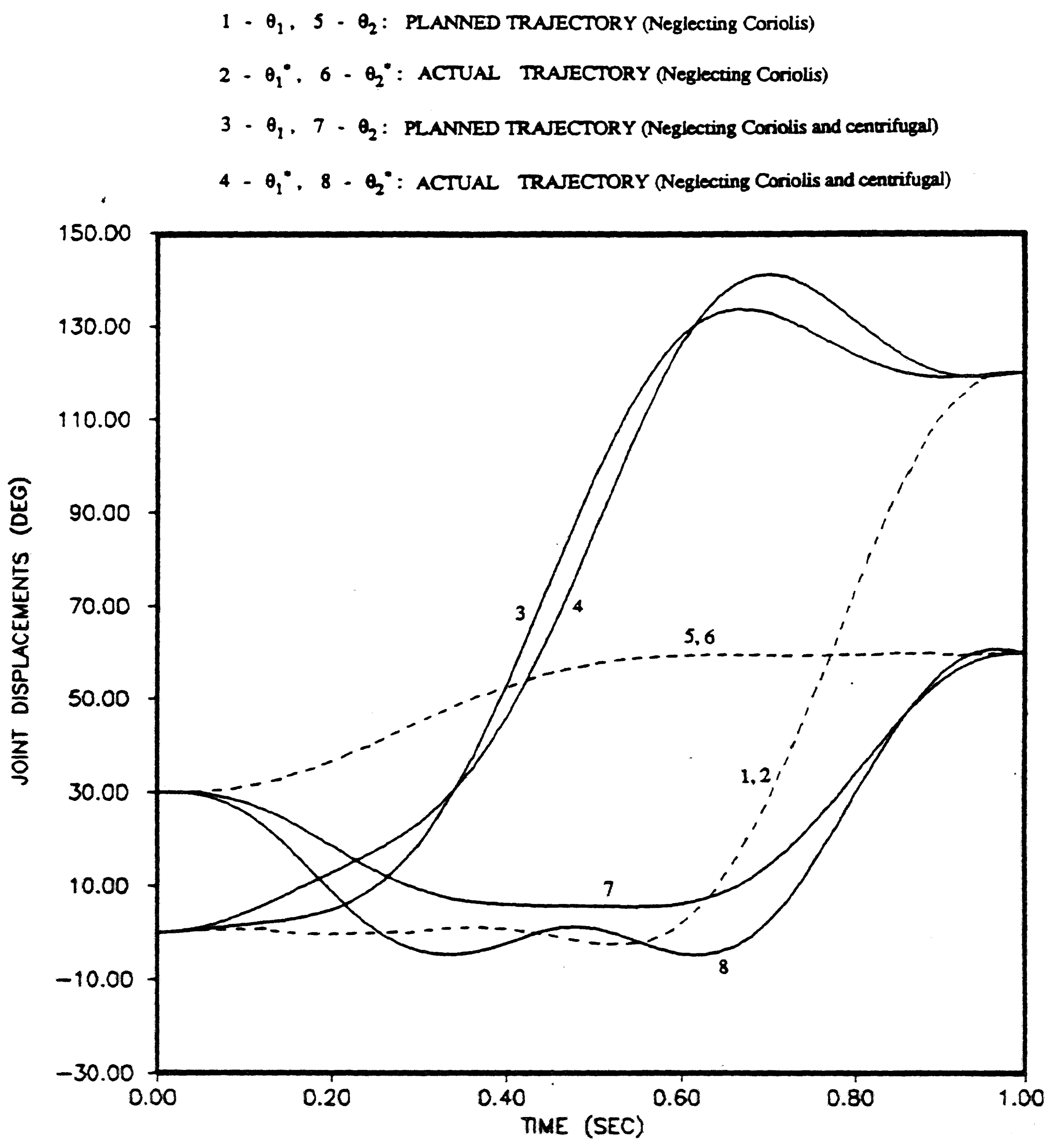


Figure 10: Suboptimal Planned and Actual Displacement Histories for Example 2 (from Figure 9.)

effective in finding optimal manipulator motions for a variety of performance indices while sidestepping many of the numerical difficulties typically encountered when applying optimal control theory directly to find such trajectories. A novel feature of this work is the feasibility of integrated trajectory planning and controller design.

8 References

1. **Kirk, D.E.**, *Optimal Control Theory: An Introduction*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1970.
2. **Sage, A.P. and White, C.C.**, *Optimum Systems Control*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1977.
3. **Kahn, M.E. and Roth, B.**, "The Near-Minimum-Time Control of Open-Loop Articulated Kinematic Chains," *ASME Journal of Dynamic Systems, Measurement, and Control*, Vol. 93, 1971, pp. 164-172.
4. **Vukobratovic, M. and Kircanski, M.**, "One Method for Simplified Manipulator Model Construction and its Application in Quasi-optimal Trajectory Synthesis," *Mechanism and Machine Theory*, Vol. 17, 1982, pp. 369-378.
5. **Vukobratovic, M. and Kircanski, M.**, "A Method for Optimal Synthesis of Manipulator Robot Trajectories," *ASME Journal of Dynamic Systems, Measurement, and Control*, Vol. 104, 1982, pp.188-193.
6. **Kim, B.K. and Shin, K.G.**, "Suboptimal Control of Industrial Manipulators with a Weighted Minimum Time-Fuel Criterion," *IEEE Transactions on Automatic Control*, Vol. AC-30, No. 1, Jan 1985, pp. 1-10.
7. **Townsend, M. and Seireg, A.**, "Optimal Trajectories and Controls for Systems of Coupled Rigid Bodies," *ASME Journal of Engineering for Industry*, May 1972, pp.472-482.
8. **Schmitt, D., Soni, A.H., Srinivasan, V., and Naganathan, G.**, "Optimal Motion Programming of Robot Manipulators," *ASME Journal of Mechanisms, Transmissions, and Automation in Design*, Vol. 107, 1985, pp.239-244.
9. **Lee, B.H.**, "An Approach to Motion Planning and Motion Control of Two Robots in a Common Workspace," Ph.D. Dissertation, Computer Information and Control Engineering Program, University of Michigan, Ann Arbor, MI, 1985.
10. **Wylie, C.R.**, *Advanced Engineering Mathematics*, 4th ed., McGraw-Hill, Inc., New York, NY, 1975.
11. **Hildebrand, F.B.**, *Advanced Calculus for Applications*, 2nd ed., Prentice-Hall, Inc., Englewood Cliffs, NJ, 1976, pp. 223-225.
12. **Avriel, M.**, *Nonlinear Programming: Analysis and Methods*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1976.
13. **Beveridge, G.S.G. and Schechter, R.S.**, *Optimization: Theory and Practice*, McGraw-Hill, Inc., New York, NY, 1970.
14. **Fox, R.L.**, *Optimization Methods for Engineering Design*, Addison-Wesley Publishing Company, Inc., Reading, MA, 1971.
15. **Reklaitis, G.V., Ravindran, A., and Ragsdell, K.M.**, *Engineering Optimization: Methods and Applications*, John Wiley & Sons, Inc., New York, NY, 1983.
16. **Nelder, J.A. and Mead, R.**, "A Simplex Method for Function Minimization," *Computer Journal*, Vol. 7, 1965, pp. 308-313.