Yen, V. and Nagurka, M.L., "A Suboptimal Trajectory Planning Algorithm for Robotic Manipulators," Proceedings of ROBEXS 87, Third Annual Workshop on Robotics and Expert Systems, Pittsburgh, PA, June 4-5, 1987, pp. 129-136.

©ISA, 1987 - Paper #87-0522
0891-4621/87/129-136/$0 + .50PP

A SUBOPTIMAL TRAJECTORY PLANNING ALGORITHM

FOR ROBOTIC MANIPULATORS

V. Yen and M. L. Nagurka
Department of Mechanical Engineering
and The Robotics Institute
Carnegie-Mellon University
Pittsburgh, PA 15213

ABSTRACT

This paper presents a Fourier-based suboptimal control approach for the trajectory planning of robotic manipulators modelled as coupled rigid bodies. The basic idea of the method is to convert a standard optimal control problem of infinite dimensionality (in time) 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 can then be solved via a nonlinear programming numerical algorithm given a performance index which is a continuous function of time. This method can be applied to a large class of problems with different performance indices and requires no model simplification. Manipulator control problems with free final time and states and with inequality constraints can be handled effectively. The method is demonstrated in two examples of trajectory planning for a planar robotic manipulator model.

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 involves the implementation of closed-loop feedback. Here, trajectory refers to the time history of position, velocity, and acceleration for each degree of freedom of a manipulator model. This research focuses on a suboptimal trajectory planning algorithm for unconstrained as well as constrained motion.

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 multiple 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, variational methods are applied 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 in tackling problems such as the optimal control of robotic manipulators, which typically have large numbers of degrees of freedom and strong nonlinearities. In fact, many nonlinear optimal control problems are still computationally infeasible to solve.

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] derived 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" 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 not be valid in many cases.

Townsend, *et al.*, [7] and Schmitt, *et al.*, [8] presented conceptually similar, but alternative, approaches for solving optimal motion problems for 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 series; 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 we guarantee that the suboptimal trajectory converges to the true optimal solution, or if this is not possible, that the suboptimal performance index, at least, converges to the true optimal performance index?
2. **Polynomial function.** Can we specify the (minimum) degree and coefficients of the polynomial function such that convergence is guaranteed?
3. **Boundary Conditions.** Can various types of boundary conditions, such as free, fixed, and coupled terminal conditions, be treated?
4. **Criterion of Optimality.** Can we identify a criterion of optimality that can be used to ensure the quality of the suboptimal solution?
5. **Applicability.** What, if any, limitations exist in applying such suboptimal approaches? For instance, can such approaches be used realistically to solve bang-bang type control problems?

In response to the above questions, this research develops a general purpose Fourier-based suboptimal control algorithm to generate manipulator trajectories. This algorithm approximates the time history of each generalized coordinate by the sum of a fifth order polynomial 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 searching for the optimal parameters of the approximating functions. Due to the nature of the conversion, the computational scheme of this method is based on an inverse dynamic approach and therefore avoids most of the numerical difficulties encountered in optimal control problems. By using standard nonlinear programming techniques, the determination of the optimal motion for high order, nonlinear robotic manipulator models is hence feasible.

Unlike previous schemes, this method does not require model simplification and can be applied to a large class of optimal control problems. Problems with variable terminal time as well as problems with free or constrained terminal states can be handled easily. In addition, a guideline that can be used to confirm the quality of the suboptimal trajectory is suggested.

This paper is organized as follows. In the following section, Section 2, the relationship among the manipulator dynamics, trajectory planning and control system design is discussed. In Section 3, the methodology and the function of various types of trajectory planning algorithms are considered. Section 4 is concerned with the formulation and numerical difficulties of optimal control problems for dynamic systems such as robotic manipulators. The main contribution of this research is presented in Sections 5 and 6. In Section 5, the Fourier suboptimal control approach is developed and in Section 6, 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 7. Conclusions are given in Section 8.

2 MANIPULATOR DYNAMICS

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 Lagrange-Euler, Newton-Euler, and generalized D'Alembert formulations. For an n degree-of-freedom manipulator configured as an open kinematic chain the equations can be expressed in the form

$$\underline{T}(t)=\underline{M}(\underline{\theta}(t))\ddot{\underline{\theta}}(t)+\underline{V}(\underline{\theta}(t),\dot{\underline{\theta}}(t))+\underline{G}(\underline{\theta}(t)) \qquad (1)$$

where $\underline{\theta}$ is an n x 1 vector of generalized coordinates associated with the n degrees-of-freedom of the manipulator, $\underline{T}$ is an $n \times 1$ vector of generalized forces applied at the joints, $\underline{M}(\underline{\theta})$ is an $n \times n$ inertial-mass matrix, $\underline{V}(\underline{\theta}, \dot{\underline{\theta}})$ is an $n \times 1$ vector representing centrifugal and Coriolis effects, $G(\underline{\theta})$ is an $n \times 1$ 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 $\underline{\theta}(t)$, while each element of $\underline{V}$ depends on both $\underline{\theta}(t)$ and $\dot{\underline{\theta}}(t)$.

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 which are based on a direct dynamic approach generally have serious convergence problems in searching for optimal solutions of high order, nonlinear systems.

Another important aspect that closely relates to manipulator dynamics is the design of the feedback controller. As indicated above, 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 (which is 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, the relative magnitude of the effective inertia at each of the joints and the inertial coupling between joints have practical importance. If the coupling inertias 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 between 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 appears reasonable to search for trajectories that give rise to minimal nonlinear effects and/or minimal dynamic coupling between joint motions so that 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 that can simplify the control system design problem. This concept of marrying the stages of control strategy and trajectory planning in order to increase the effectiveness of the control law is demonstrated by an example in Section 7.

3 PLANNING MANIPULATOR TRAJECTORIES

Typically, in manipulator programming the trajectory planner is viewed as a black box. The inputs to the trajectory planner are usually the path specifications, where the path is defined as the space curve that the manipulator end-effector moves along from the initial to final location (position and orientation). In addition, the planner can accept 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 trajectory and the generalized force history.

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 must traverse 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 of the manipulator is preferred especially for cases of on-line implementation. Because current trajectory planners usually do not account for (i) the interaction between the trajectory and controller, and (ii) the dynamic constraints, large tracking errors may result. 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 which can be evaluated by an objective function (*i.e.*, performance index). These goals led to the development of the trajectory planning algorithm presented in this paper.

4 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 with infinite dimensionality to 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

$$\boldsymbol{\theta}(t)=\sum_{k=1}^{K} c_k u_k(t) \qquad (2)$$

Here the generalized coordinate variable, $\boldsymbol{\theta}$, consists of a sum of the product of known approximating function, u_k, and weighting constant, c_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. Best results are usually obtained when using approximating functions drawn from a functionally complete set of eigenfunctions in the interval of interest.

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.

5 FOURIER-BASED SUBOPTIMAL CONTROL ALGORITHM

Given the dynamic equations of motion of an n degree-of-freedom manipulator model, equation (1), 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 $\dot{\underline{\theta}}_{opt}$, such that the performance index,

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

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 $\dot{\underline{\theta}}(t_o)$, and the initial time, t_o, are specified. The final time, t_f, and the terminal states, $\underline{\theta}(t_f)$ and $\dot{\underline{\theta}}(t_f)$, can either be free or fixed.

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

$$\underline{E}(\underline{\theta}(t),\dot{\underline{\theta}}(t),t) \le \underline{0} \quad (4)$$

where $\underline{E}$ is an $m \times l$ ($m < n$) vector function of the states and possibly time. In the trajectory planning problem, these constraints usually represent obstacle (avoidance) constraints that exist in the working environment. The second class of constraints, actuator-related inequality constraints, can be expressed as

$$|T_i| \le \tau_i \; , \qquad i=1,...,n \quad (5)$$

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 of the joint angular displacements by the sum of a fifth order polynomial and a finite Fourier-type series. For example, for joint i,

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

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

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

and the K term Fourier-type series is defined as

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

The velocity and acceleration of the i-th joint are obtained by direct differentiation of the above equations._ Control variables can be calculated readily from the equations of motion. The performance index can also be computed by straightforward numerical integration methods such as Simpson's composite integral technique.

Assuming both the initial conditions and terminal 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:

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

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

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

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

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

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

Here the initial accelerations, $\ddot{\theta}_i(t_o)$, and final accelerations, $\ddot{\theta}_i(t_f)$, of the joint variables as well as the coefficients of the Fourier series are the variables left to be determined. The search for the optimal trajectory, which in theory consists of an infinite number of points, is thus converted to a nonlinear programming problem with a finite number of free variables. These variables are the Fourier-type coefficients (a_{ik}, b_{ik}) and the free boundary conditions of the trajectory.

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, which are 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 true optimal solution. To do this, the following property is first introduced.

Property of Differentiability: The necessary and sufficient conditions for $\dot{X}(t) = \dot{F}(t)$ in the interval $[\delta,\delta+2\alpha]$ where $F(t)$ is $X(t)$'s Fourier series and $X(t)$ is continuous are (i) $X(t)$ is continuous, (ii) $X(t)$ is piecewise differentiable in $(\delta, \delta+2\alpha)$, (iii) $X(\delta) = X(\delta+2\alpha)$, and (iv) $\dot{X}(\delta) = \dot{X}(\delta+2\alpha)$. This property can be generalized to the second derivative case.

According to this property, we can conclude that as long as the above four conditions are true, the result of term by term differentiation of the Fourier series of period 2α representing $X'(t)$ in the interval of $[\delta, \delta + 2\alpha]$ converges to $X'(t)$ at each point in $[\delta, \delta + 2\alpha]$ at which $X(t)$ is continuous. A proof can be found in [11].

From this property, we find 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 (including displacement, velocity and acceleration) to the true optimal solution is thus guaranteed.

6 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.

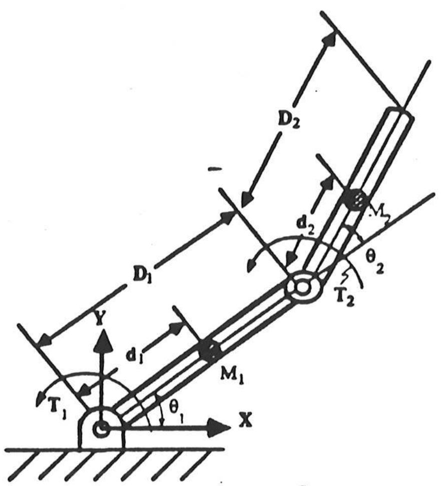
Numerical algorithm. The original optimal control problem has been converted to a problem of ordinary extrema which can be solved by a number of well-developed nonlinear programming techniques [12-15]. For example, the simplex method [16] is adopted in this research.

Accuracy. A closed form 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 can be estimated and controlled, the problem of convergence which is encountered frequently in implementing standard optimal control approaches is avoided.

Terminal states and time. The terminal states and terminal time can be treated as free variables. During the search for the optimal solution, they -- together with other variables, such as the coefficients of the Fourier series -- are adjusted simultaneously in every iteration to minimize the performance index.

Restrictions. One of the necessary conditions for the convergence of the proposed approach is the continuity of displacement, velocity and acceleration of the true 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 will only converge to the average of the neighboring points at the point of switch. However, since bang-bang control has finite switch points, the value of the suboptimal performance index will still converge to the value of the performance index of the true optimal solution.

Although in theory the proposed approach is capable of achieving true 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%), 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 in cases where there is an instantaneous shift of the control input (a situation which is physically impossible).



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

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

Figure 1. Two Link Planar Manipulator Model.

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 good idea to use the previous solution as part of the initial guesses 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 to the true optimal trajectory.

7 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:

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

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

where

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

$$H_{22} = M_2 d_2^2 + I_2$$

$$H_{12} = M_2 D_1 d_2 \cos\theta_2 + M_2 d_2^2 + I_2$$

$$H = M_2 D_1 d_2 \sin\theta_2$$

The path specification is to move the manipulator from initial position $[\theta_1(0), \theta_2(0)] = [0^\circ, 30^\circ]$ to final position $[\theta_1(1), \theta_2(1)] = [120^\circ, 60^\circ]$ in 1 second with the initial and final velocity zero.

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, a performance index representing certain performance characteristics of the manipulator can be proposed. The optimal trajectory can then be obtained by applying the suboptimal method presented in this paper. In this example, the performance index is the control effort, represented by

$$\sum_{k=1}^{2} T_k^2$$

In the suboptimal approach, each angular displacement is approximated by the sum of a fifth order polynomial and a two term Fourier-type series. 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 time histories of the angular displacements and the performance index (using a cubic polynomial trajectory and the suboptimal trajectory) are shown in Figures 2 and 3, respectively. Figure 2 shows that the cubic polynomial trajectory connects the two end points by smooth interpolation, whereas the suboptimal solution involves displacements of the joints that deviate_outside the boundaries. Although the suboptimal trajectory appears excessive and inefficient, the performance index remains small during the motion. This contrasts with the cubic polynomial trajectory for which the performance index is especially large near the boundaries. Integration of the curves of Figure 3 show that as a result of the optimization the value of the performance index decreases from 387.5 to 122.9 N^2-m^2-sec, indicating a significant reduction in manipulator control effort.

Example 2. The previous example represents a typical, standard optimal control problem since it concentrates only on the performance of the open loop system. However, with an appropriately selected performance index and a simplified dynamic model, a scheme to improve the closed loop system performance can be proposed. The flowchart of this scheme is given in Figure 4.

The actual (on-line) generalized forces, $\underline{T}$, are calculated according to a prespecified simplified model. The performance index is chosen to represent the difference between the planned trajectory, $\underline{\theta}$, and the actual trajectory (*i.e.*, system response), $\underline{\theta}^*$. The influence of the dynamic terms, which were neglected in the simplified model, can thus be minimized when the manipulator moves along the optimal trajectory. It is expected that the simplified model can satisfactorily simulate the real dynamic model in the neighborhood of the proposed trajectory. Thus, 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 the simplified controller.

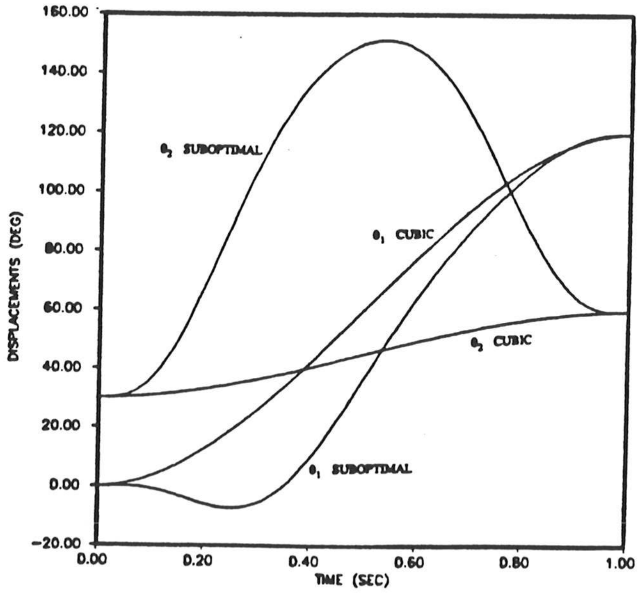


Figure 2. Joint Displacement Histories for Example 1.

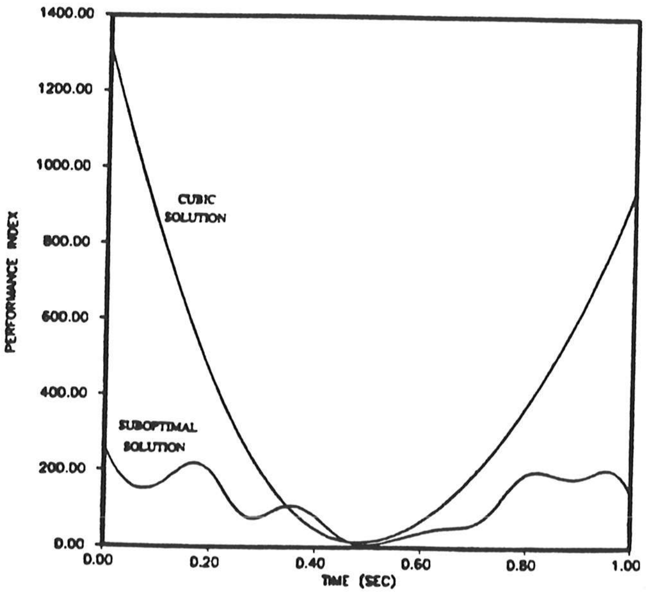


Figure 3. Control Effort History for Example 1.

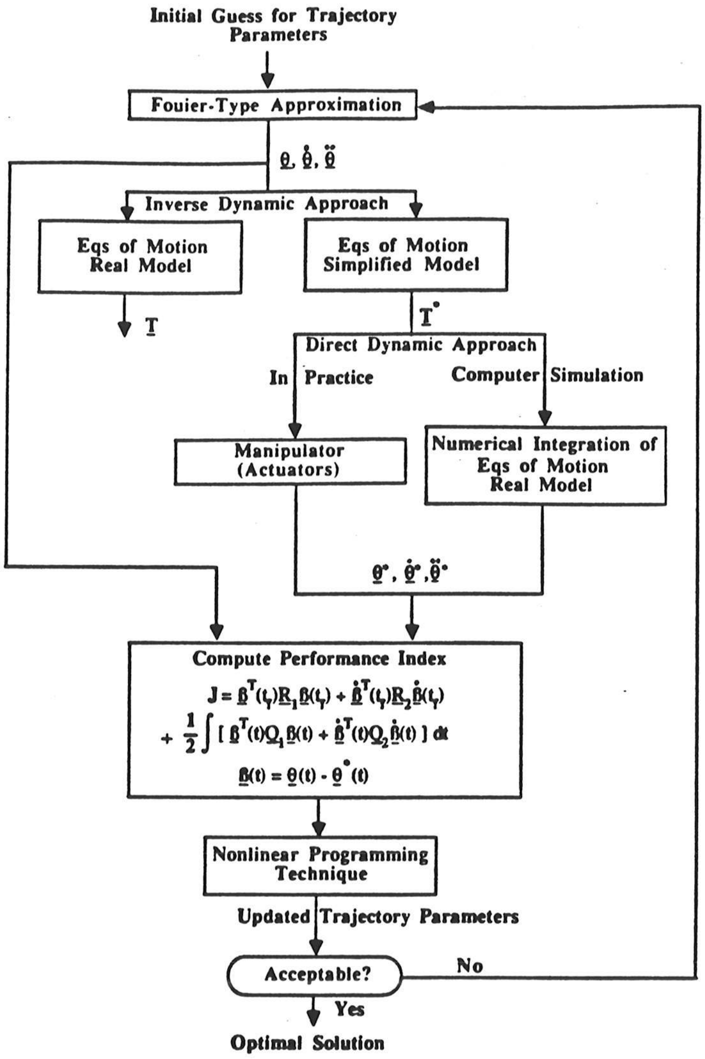

Initial Guess for Trajectory Parameters
Fouier-Type Approximation
$\underline{\theta}, \dot{\underline{\theta}}, \ddot{\underline{\theta}}$
Inverse Dynamic Approach
Eqs of Motion Real Model
Eqs of Motion Simplified Model
$\underline{T}$
$\underline{T}^*$
Direct Dynamic Approach
In Practice
Computer Simulation
Manipulator (Actuators)
Numerical Integration of Eqs of Motion Real Model
$\underline{\theta}^*, \dot{\underline{\theta}}^*, \ddot{\underline{\theta}}^*$
Compute Performance Index
$J = \underline{\beta}^T(t_f)\underline{R}_1\underline{\beta}(t_f) + \dot{\underline{\beta}}^T(t_f)\underline{R}_2\dot{\underline{\beta}}(t_f)$
$+ \frac{1}{2}\int [\,\underline{\beta}^T(t)\underline{Q}_1\underline{\beta}(t) + \dot{\underline{\beta}}^T(t)\underline{Q}_2\dot{\underline{\beta}}(t)\,]\,dt$
$\underline{\beta}(t) = \underline{\theta}(t) - \underline{\theta}^*(t)$
Nonlinear Programming Technique
Updated Trajectory Parameters
Acceptable?
No
Yes
Optimal Solution

Figure 4. Flowchart of an Integrated Algorithm for Trajectory and Controller Design.

In this example, the manipulator model and path specification are the same as in the first example. The performance index is assumed to be:

$$J = 100[\theta_1(1) - \theta_1^*(1)]^2 + 100[\theta_2(1) - \theta_2^*(1)]^2 + \int_0^1 [(\theta_1 - \theta_1^*)^2 + (\theta_2 - \theta_2^*)^2]\,dt \tag{17}$$

Two simulation cases were studied. In the first case, the actual torques were generated based on a simplified model (*i.e.*, equations of motion) that neglected the Coriolis effect. In the second case, the simplified model neglected both Coriolis and centrifugal effects.

Figures 5 and 6 display the joint displacement histories from the cubic polynomial and the suboptimal algorithm, respectively. The planned displacements and the actual displacements for the two simulation cases are plotted.

Figure 5 shows that significant tracking error occurred for the cubic polynomial trajectory when the actual torques were calculated based on simplified models. (Here, the tracking error is defined as the difference between the actual and planned trajectories.) For both joints the tracking error was largest for the second simulation case in which both the Coriolis and centrifugal terms were neglected. For both simplified models, the actual trajectories failed to satisfy the final boundary conditions and deviated significantly, especially for joint 2.

Figure 6 shows that the tracking error for the suboptimal trajectory was reduced significantly in comparison to the error associated with the cubic polynomial trajectory. In fact, the tracking error for the first simulation case (in which only Coriolis was neglected) was so small that the curves in the figure coincide. (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 were neglected), the tracking error is observable in the figure; however, it remains small relative to the tracking error of Figure 5.

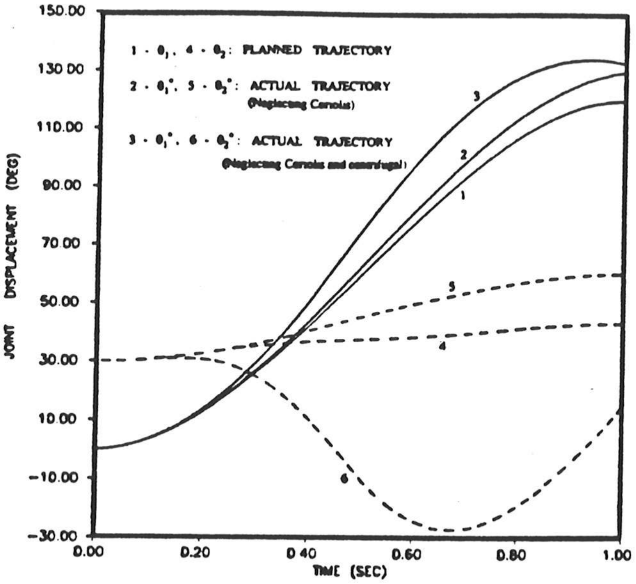


Figure 5. Joint Displacement Histories from Cubic Polynomial for Example 2.

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 simplifed controller designs may be increased.

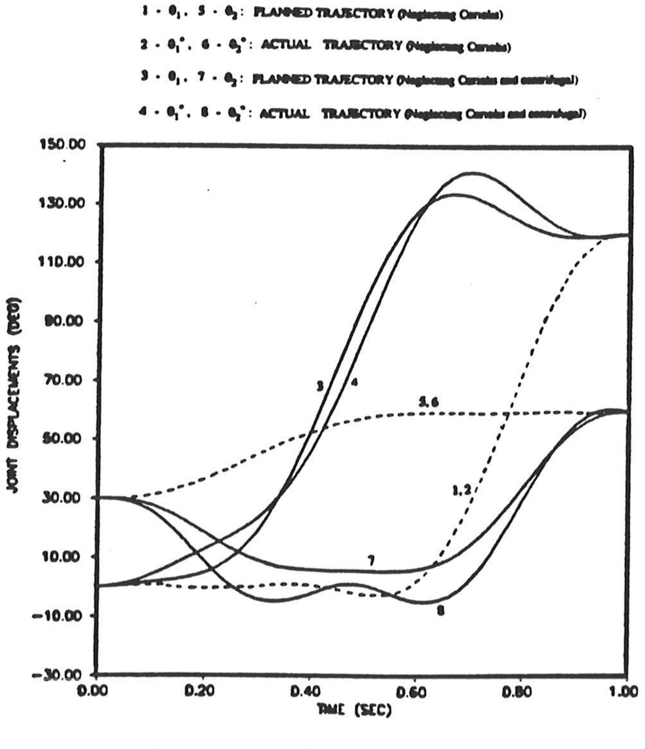


Figure 6. Joint Displacement Histories from Suboptimal Algorithm for Example 2.

8 CONCLUSION

This paper presents a general-purpose suboptimal trajectory generation algorithm for robotic manipulators. The proposed approach is a Fourier-based method which converts an optimal control problem into a nonlinear programming problem. The algorithm is especially effective in finding optimal manipulator motions for a variety of performance indices while sidestepping many numerical difficulties typically encountered when directly applying optimal control theory to find such trajectories. A novel feature of this work is that integrated trajectory planning and controller design is realizable by the proposed methodology.

9 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 Sciereg, 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.