ROBOTS 14

Conference Proceedings

November 12-15, 1990
Detroit, Michigan

Sponsored by

Robotics International of the
Society of Manufacturing Engineers
One SME Drive, P.O. Box 930
Dearborn, Michigan 48121

ROBOTS 14

CONFERENCE PROCEEDINGS

Society of Manufacturing Engineers
Dearborn, Michigan

First Printing

Library of Congress Catalog Card Number: 90-61668
International Standard Book Number: 0-87263-390-X
Manufactured in the United States of America

TECHNICAL REPORT

MS90-12

Collision-free Optimal Motion Planning

authors

Jason S. Wong (previously known as Jyhshing Jack Wang)

abstract

Collision-free optimal motion and trajectory planning for robotic manipulators are solved by a method of sequential gradient restoration algorithm. Numerical examples of a two degree-of-freedom (DOF) robotic manipulator are demonstrated to show the excellence of the optimization technique and obstacle avoidance scheme. The obstacle is put on the midway, or even further inward on purpose, of the previous no-obstacle optimal trajectory. For the minimum-time purpose, the trajectory grazes by the obstacle and the minimum-time motion successfully avoids the obstacle. The minimum-time is longer for the obstacle avoidance cases than the one without obstacle. The obstacles avoidance scheme can deal with multiple obstacles in any ellipsoid forms by using artificial potential fields as penalty functions via distance functions. The method is promising in solving collision-free optimal control problems for robotics and can be applied to any DOF robotic manipulators with any performance indices and mobile robots as well. Since this method generates optimum solution based on Pontryagin Extremum Principle, rather than based on assumptions, the results provide a benchmark against which any optimization techniques can be measured.

conference

ROBOTS 14
November 12-15, 1990
Detroit, Michigan

index terms

Robotics

Society of Manufacturing Engineers
1990

Sponsored by Robotics International
of the Society of Manufacturing Engineers

Collision-free Optimal Motion Planning

Jason S. Wong (previously known as Jyhshing Jack Wang)

Collision-free optimal motion and trajectory planning for robotic manipulators are solved by a method of sequential gradient restoration algorithm. Numerical examples of a two degree-of-freedom (DOF) robotic manipulator are demonstrated to show the excellence of the optimization technique and obstacle avoidance scheme. The obstacle is put on the midway, or even further inward on purpose, of the previous no-obstacle optimal trajectory. For the minimum-time purpose, the trajectory grazes by the obstacle and the minimum-time motion successfully avoids the obstacle. The minimum-time is longer for the obstacle avoidance cases than the one without the obstacle. The obstacles avoidance scheme can deal with multiple obstacles in any ellipsoid forms by using artificial potential fields as penalty functions via distance functions. The method is promising in solving collision-free optimal control problems for robotics and can be applied to any DOF robotic manipulators with any performance indices and mobile robots as well. Since this method generates optimum solution based on Pontryagin Extremum Principle, rather than based on assumptions, the results provide a benchmark against which any optimization techniques can be measured.

1. Introduction

The problem of increasing productivity, automated manufacturing, and performing complex tasks in hazardous or remote environments can be solved by robotic systems. Such systems have been applied to a wide variety of industries which includes spray painting, welding, assembling, material handling, highly risky work and remote control jobs. As pointed out by Holcomb and Montemerlo [1] and Lerner [2], remote control robotic systems will be developed in the future space stations. Also as well-known, with the demand of increasing productivity and industrial automation, the problem of controlling the robotic manipulators has received a great deal of interest in the field of automated manufacturing.

1.1 Research Objectives

One of the focal points in robot design lies in the computation and control of the motion of the manipulator. In order to make sure that the manipulator is able to execute a special task most efficiently for human beings, it is important that the manipulator performs from initial states to designated final states in an optimal way under collision avoidance concern. Control on robotics can be separated into two major categories: 1) trajectory planning, 2) trajectory tracking. Various optimal controllers need to be devised in the trajectory tracking problems which are not the subjects in this article. Trajectory planning is not only the determination of the path of the end effector. Trajectory planning generates a specified motion of time history from initial states to final states. Motion planning does not necessarily require optimization techniques but extra excursion of the robot is just not cost-efficient and can cause more potential collision problems. Obviously, the minimum-time trajectory is of particular interest since the productivity in automation is maximized. Various performance goals, for example: distance, energy or time-energy combination, are also applicable. Various concepts for the study of optimal control of robotic manipulators have been studied for this purpose.

1.2 Previous Work

One of the pioneered work is done by Kahn and Roth [3]. The highly nonlinear manipulator dynamical equations of motion are linearized, an approximate bang-bang solution has been developed to the suboptimal feedback control problem.

Gilbert and Johnson [4] have developed a path planning scheme in which the obstacles are avoided via an infinite penalty function generated from distance function. In their study, the nonlinear dynamic equations are approximated by linear subspace functions which are chosen as piecewise polynomial splines. In their examples, distance constraints are violated when spline knot interval sections equal to one; the payload object is made strictly convex by approximating its boundary by arcs of certain curvature; obstacles are assumed to be convex sets; the complex distance finding minimization problem within the optimal control problem is not fully described. In the optimization technique, more than one optima can be drawn at the same case.

Based on Pontryagin extremum principle, the time-optimal motions of various types of robotic manipulators have been investigated by Geering, Guzzella, Hepner and Onder [5] as classified by cylindrical, spherical robots, and a robot with horizontal articulated arm with two links. In the analysis of the time-optimal control problem, the bang-bang control solution satisfies the Pontryagin extremum principle and the study has been made for unconstrainted trajectories. In their examples, two links intercross each other in the planar two-link manipulator.

Due to the difficulty of highly nonlinear robot mathematical model, a near-optimal control algorithm based on Pontryagin extremum principle and Riccati formulation has been presented by Kim, Jamshidi and Shahinpoor [6]. The algorithm reduces the original nonlinear equation set into a linear one by a parameter sensitivity method and P-D controller is used to solve the linearized model.

Ozaki and Mohri [7] has developed the study of collision-free joint trajectories along a given path with some physical constraints such as manipulator dynamics, obstacles avoidance, joint velocities and input torques by formulating artificial potentials into the planning problem for constraints using linear programming algorithm to minimize the error between present and desired trajectory, in which, the nonsmooth time functions were approximated by cubic spline functions.

The technique of dynamic programming has also been a popular solution method to many investigators in the field of robotic manipulators research. Based on dynamic programming, Vukobratovic and Kircanski [8] have determined the energy-optimal velocity distribution of the manipulator end-effector for a prescribed path in the workspace subject to the forces/torques constraints. The given traveling time needs to be discretized in their study.

Singh and Leu [9] have formulated and solved the optimal trajectory planning as an optimal control problem by a path parameterized method of dynamic programming under the constraints of the joint forces/torques and velocities. Bang bang control has been generated for minimum time problems without obstacles avoidance concern.

In order to implement dynamic programming approach, Shin and McKay [10] have studied trajectory planning of robotic manipulators using parametric function and its derivative to reduce dimensions in state space which finds the positions, velocities, accelerations, and torques of the problem by minimizing the cost of the parameter of moving a robotic manipulator along a specified geometric path subject to input torque/force constraints without obstacle avoidance concern. Along a pre-selected geometric path, for quadratic velocity bounds, and piecewise analytic geometric path constraints, the minimum-time control problem has been studied by Shin and McKay [11] with the phase-plane techniques in Cartesian space which has to be converted into joint space by interpolation. Under the assumption that the path is given as parameterized curve, they have also determined a near-minimum time geometric path for the study described above which minimizes approximate lower traversal time bounds using maximum velocity bounds [12]. Their techniques are limited by parameterization.

Bobrow, Dubowsky, and Gibson [13] have studied the problem of minimum-time trajectories along arbitrary pre-planned spatial paths by a special technique in which the actuator torque bounds are assumed to be functions of the robot's current position and velocity. This technique cannot handle the case when the feasible regions in the phase plane are not simply connected. The idea of the time-optimal solution is based on choosing the maximum acceleration/deceleration to make velocity as large as possible at every point without violating constraints. The difficulty is finding multiple switching points for time-optimal problems. Dubowsky, Norris and Shiller [14] have devised a time optimal trajectory planning scheme with obstacle avoidance consideration via a CAD approach in which the minimum distance to obstacles is found from software OPTARM II by a table of various geometric shape. The penalty function for obstacle avoidance needs to have a characteristic of more effective weighting and dramatic steepness. The technique cannot be easily extended to solving optimal trajectory planning for other performance indices and constraints. Based on the same assumption, Rajan [15] has devised a parameterized path method for the minimum-time problem in which the cubic spline paths are parameterized and optimized locally by an iterative scheme. The optimization procedure stops until the minimum-time path comes close enough to the previous path while using

Bobrow's algorithm for inner minimization and varying the path for outer minimization. The algorithm cannot be effectively applied to the planar movements of a manipulator with obstacles in the workspace. The weak points of the algorithm are on the premises that the minimum time path is smooth and a smooth curve is well approximated by splines.

Sahar and Hollerbach [16] have devised a method based on state-space search tree representing all possible solutions, and searching for the best one by using a Symbolics Lisp Machine for time-minimum criterion. The algorithm is a logical approach but not a mathematical approach which is not suitable for routine off-line trajectory planning due to the complexity of computation.

Luh and Lin [17] have devised a kinematical approach which assumes the path consists of a sequence of Cartesian straight line segments and constant limits on Cartesian velocity and acceleration are known a priori without considering the dynamics of the arm.

Weinreb and Bryson [18] have presented the Adjustable Control-Variation Weight (ACW) algorithm for the minimum-time control of a two-link robotic arm through choosing controls subject to the actuator constraints. In their examples, the two links of the planar manipulator intercross each other. Meier and Bryson [19] have developed an algorithm for solutions of time-optimal control problem of a two-link planar manipulator which contains solutions for two-point boundary value problem of constrainted motion between two endpoints.

Zhang and Wang [20] have investigated a collision-free time-optimal control problem of a two-link planar robotic manipulator by applying the method of global linearization transformation in joint space configuration. As a result, the nonlinear equations of motion are transformed into an equivalent linear model and an approximate explicit expression has been obtained for the case of minimum-time control of a two-link planar robotic manipulator with two-dimensional planar geometrical obstacle avoidance. In their example, radius of the circle obstacle is not shown.

Bobrow [21] has continued the study of optimal path planning using minimum-time criterion with obstacles avoidance consideration in which the actuator torque bounds are assumed to be functions of the robot's current position and velocity, where the Cartesian path of the end-effector is represented with uniform cubic B-spline polynomials. The obstacle avoidance is enforced by ensuring the distance between the end-effector and the obstacle which was evaluated by stepping small increments of the path parameter.

Wang [22] has devised the numerically approach of using sequential gradient restoration algorithm to solve Bolza classical optimal control problem on robotics without linearization or parameterization, including the analytical time-optimal solutions of a two-link manipulator and/or actuator constrainted cases, in which the implementation can be extended into obstacle avoidance consideration.

1.3 Overview

We can see that numerous attempts have been made to find collision-free optimal motion of a robotic manipulator without great success. All of the aforementioned investigations are limited in one way or another.

Collision-free optimal control problems for robotic manipulators are difficult due to the two-point boundary-value problem which involves, in addition to the optimality conditions, the kinematical and highly nonlinear dynamical equations of the system, the obstacle constraints, the limits imposed on controls, and the satisfaction of terminal conditions. Generally speaking, analytical solutions for classical optimal control problems with equality and/or inequality constraints are not possible. Therefore, numerical method is resolved. Numerical methods and computer routines are available nowadays ranging from simple integration to TPBVP and optimization at a low price [23].

To solve constrainted optimal control problems, a restoration phase is needed at the end of the gradient phase [24]. The collision-free motion planning problems of robotics can be formulated as a classical optimal control problem and solved by sequential gradient restoration algorithm [25]. Collision can be avoided by continuously controlling the closest point on the arm to the obstacles using virtual potential fields as penalty functions via distance functions [26].

1.4 Present Modeling

As pointed out in recent research, owing to the difficulty of solving TPBVP and highly nonlinear dynamic equations, the classical optimal control problem is mostly approached by approximation (linearization, parameterization, modification) which more or less replaces the original optimal control problem into the assumed one. As in those study where the nonlinear dynamic equations or the two-point boundary value problems are linearized or parameterized, the solutions generated based on those assumptions are not necessarily good approximations to the original ones. The intention of this research is to present a numerical approach for determining the collision-free optimum motion of robotic manipulators, a method to solve classical optimal control problem without any modification, linearization or simplification. Solutions including robot positions, velocities, accelerations and force/torque in both Cartesian space and joint space which satisfies the Pontryagin extremum principle are obtained by solving the manipulator kinematical and dynamical equations with optimality conditions. For given initial and final conditions, under the physical conditions imposed on control in joint space and obstacles constraints, the continuous time-history of the positions, velocities, accelerations, torques/forces and the optimal collision-free motion of a robotic manipulator in minimum time are determined.

Applications of sequential gradient restoration algorithm occur in various branches of science and engineering. With particular regards to aerospace engineering, various problems of coplanar and noncoplanar, orbital and suborbital space flight [27, 28, 29] and atmospheric flight in a windshear [30, 31, 32] have been solved by the sequential gradient restoration algorithm. The same technique has been successfully employed in the thermofluid science [33, 34]. Also, optimal safe landing of a helicopter in autorotation has been studied using sequential gradient restoration techniques [35]. In general, sequential gradient restoration algorithm has proven to be a very promising algorithm in solving engineering optimal control problems [36, 37].

1.5 Advantages over Existing Techniques

To solve collision-free optimal control problems on robotics with constraints, we need a numerical method which has the following advantages:
1) able to solve TPBVP which is essentially the core of the problem [6, 8, 10, 11, 12, 13, 14, 15, 21]: In fact, TPBVP can be solved by shooting method and relaxation method or method of particular solutions. TPBVP is involved in the first-order exact optimality conditions derived from calculus of variation.
2) able to solve highly nonlinear dynamic equations without linearization, parameterization or simplification [3, 4, 6, 7, 10, 11, 12, 17, 20, 21]: Any modification by linearization or simplification directly or indirectly replaces the original problem. The drastic approximation leads to significant error and unsatisfactory, unknown effect to the optima and the obstacle avoidance. For example, a collision-free optimal solution can be declared only when there is not another more optimal solution.
3) able to solve any robotics formulation regardless number of joints or DOF [15, 17, 18, 19, 20]: A technical approach should not be limited by the number of joints or DOF of robotics. Any dynamic systems can be formulated from state functions point of view and solved as control systems regardless number of dimensions.
4) able to avoid the obstacles toward optimization direction without any unnecessary excursion [5, 7, 8, 9, 14, 15, 20, 21]: Collision avoidance should be achieved in a most efficient way, in terms, an optimal way, without requiring extra journey of the robot arm. The weighting effect and clearence between trajectory and obstacle should be specified by only one parameter.
5) able to solve general constraints of robot motion planning: On trajectory planning, we have state constraints, control constraints, or a combination of the above. Obstacle inequality constraints, control inequality constraints can be transformed into equality constraints.
6) able to solve any terminal conditions, any performance indices [3, 13, 14, 15, 18, 21]: In various applications, various performance indices need to be implemented. For example, time, distance, energy or a combination of the above. Point to point task has different initial and final states in applications.
7) has the potential to fully utilize computer power as the computer industries grow in the near future: Several years from now, computers can be many times faster in CPU. We don't reject any ideas which consume more CPU time than we can afford today. On the contrary, we encourage numerical method that fully utilizes the modern scientific computing concepts. Provided we have infinitesimal small stepsize and infinite digits, and we have sufficient CPU time on computers, this calculus of variation approach generates solutions which satisfy exact necessary conditions. Sufficient conditions can also be checked.

1.6 Drawbacks

There are certainly some drawbacks: 1) Minimum distance finding problem within the optimal control problem at real-time is difficult. For more complex manipulators and general obstacles, the minimum distance can be found through optical devices and solids modeling techniques. In common sense, a human being has to sense (by eyes) obstacles and potential of collision before he/she can think about avoidance. 2) Due to the consumption of huge amount of CPU time, the method is good for off-line programming but is not yet ready for real-time, on-line applications. This can be remedied by parallel computing techniques.

1.7 Contents

A brief description of the dynamic systems and constraints are given in section 2. Section 3 contains the obstacles avoidance schemes. Section 4 contains the optimal control theory. Section 5 contains the sequential gradient restoration algorithm. In Section 6, numerical examples of a two degree-of-freedom robotic manipulator are demonstrated. The insight of collision-free minimum-time motion are shown in captions and tables. Finally, discussion is in section 7, conclusion and prospective research are presented in section 8. Appendix A illustratess the kinematics of a two-link manipulator example.

2. Dynamic Systems and Constraints

The highly nonlinear dynamic equations and inequality control constraints and/or inequality state constraints are also the main difficulties of optimal control on robotics.

2.1 Dynamics

Under the assumption that the links are uniform rods of mass m_i at the mass center, of moment of inertia I_i, of length l_i, respectively, i is the number of the link. The gravity g is acting parallel to the negative y-axis direction. The dynamical equations can be derived by means of Newton-Euler (Lagrange-Euler) equations [38, 39, 40] or symbolic method [41] and expressed in general as:

$$\tau = \mathbf{M}(\theta)\alpha + \mathbf{C}(\theta,\omega) + \mathbf{G}(\theta) \qquad (1)$$

where τ is the vector of applied torques/forces, $\mathbf{M}(\theta)$ is the inertial matrix terms of the manipulator, $\mathbf{C}(\theta,\omega)$ is the vector of centrifugal and Coriolis terms, $\mathbf{G}(\theta)$ is the vector of gravity terms. For example, a two-link manipulator in Fig. 1. [16]:

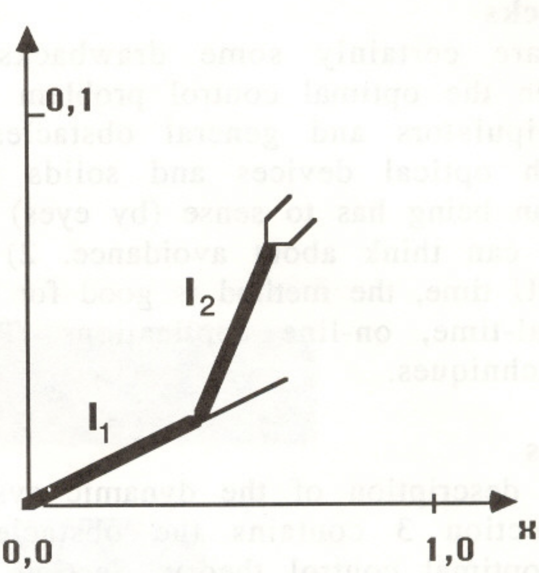

Fig.1. Two-link robotic manipulator

Link 1 of m_1, I_1, l_1

Link 2 of m_2, I_2, l_2

$$M(\theta)=\begin{pmatrix} I_1+I_2+(m_1l_1^2+m_2l_2^2)/4+m_2l_1^2+m_2l_1l_2\cos\theta_2 & I_2+(m_2l_2^2)/4+(m_2l_1l_2\cos\theta_2)/2 \\ I_2+(m_2l_2^2)/4+(m_2l_1l_2\cos\theta_2)/2 & I_2+(m_2l_2^2)/4 \end{pmatrix} \tag{2}$$

$$C(\theta,\omega)=\begin{pmatrix} -m_2l_1l_2\sin\theta_2(\omega_2)^2/2-m_2l_1l_2\sin\theta_2(\omega_1)(\omega_2) \\ m_2l_1l_2\sin\theta_2(\omega_1)^2/2 \end{pmatrix} \tag{3}$$

$$G(\theta)=g\begin{pmatrix} m_2l_2\cos(\theta_1+\theta_2)/2+l_1(m_1/2+m_2)\cos\theta_1 \\ m_2l_2\cos(\theta_1+\theta_2)/2 \end{pmatrix} \tag{4}$$

One can see that these highly nonlinear terms are functions of the joint velocities and angles. θ_i, ω_i, α_i are relative angle, angular velocity, and angular acceleration of link i respectively.

2.2 Control Systems and Inequality Control Constraints

Robotics dynamic system can be formulated in two ways:

2.2.1 Kinematical Formulation

Kinematical formulation is practical in most cases, specially when the model reference dynamic parameters are not known in advance. For example, the payload is never known ahead; or for safety reason that the inertia force caused by acceleration of the robots shall be limited. In kinematical formulation, the control system is as follows:

$$\dot{\theta} = \omega \tag{5}$$

$$\dot{\omega} = \alpha \tag{6}$$

θ, ω, α are vector of state variables. Once the states in joint space of the manipulator are known, we can compute the joint torques which are required to balance the reaction forces/moments acting on the links. The physical inequality constraints imposed on the robot in this study are joint acceleration bounds [20]. With these constraints, we can limit the torques which are related to the joint space configuration. In terms,

$$|\alpha| \leq C, \tag{7}$$

Via the following variable transformation, the joint acceleration can be limited within the bounds

$$\alpha = C\sin(u), \tag{8}$$

C is vector of upper bounds of the absolute acceleration in joint space. u is vector of the new control variable.

2.2.2 Dynamical Formulation

If we know the model reference system in advance, in dynamical formulation, the control system is as follows:

$$\dot{\theta} = \omega \tag{9}$$

$$\dot{\omega} = \mathbf{M}^{-1}(\tau - \mathbf{C}(\theta,\omega) - \mathbf{G}(\theta)) \tag{10}$$

θ, ω, α are state variables. In this formulation, we assume the dynamic parameters in matrices **M**, **C**, **G**, are known. The matrix **M** is always both "symmetric and positive definite" [39], therefore always invertible. The physical inequality constraints imposed on the robot in this formulation are joint torque/force bounds. With these constraints, we can limit the torques in the actuator space configuration. In terms,

$$|\tau| \leq C, \tag{11}$$

Via the following variable transformation, the joint torque can be limited within the bounds

$$\tau_i = C_i\sin(u_i), \tag{12}$$

C is vector of upper bounds of the absolute torque in actuator space. u is vector of the new control variable.

2.3 Equality Constraints

In some cases, the end-effector has to follow a specified path, or the orientation of the arm in the motion is specified and fixed, for example, the robot arm is holding a flash light moving along a specified path, then, the degree of freedom is reduced by the number of constraints. One or more state constraints have to be added in Cartesian space, then converted into joint space. The system is solved with replacement of the algebraic equation into the state variables according to the constraints.

3. Obstacles Avoidance Schemes

By definition, obstacles can be avoidable or unavoidable for a fixed configuration. Configuration has to be fixed in one task to avoid excess excursion and changing kinematics. For examples, in Fig. 2., the obstacle is away from the robot chassis but within the work envelope. That is considered as avoidable. In Fig. 3., obstacles are too close to the robot and there is no space for feasiblly moving the robot arm through the obstacle environment. This is considered as unavoidable.

For simplicity, each obstacle is put into an ellipsoid. It is a little wasteful to put an obstacle which is not necessarily in ellipsoid shape into an ellipse. The advantage is the ellpsoid parameters can be changed to shapen the oral into the figure of the obstacle without wasting too much space. Collision avoidance can be achieved by continuously controlling the closest point on the arm to the obstacles.

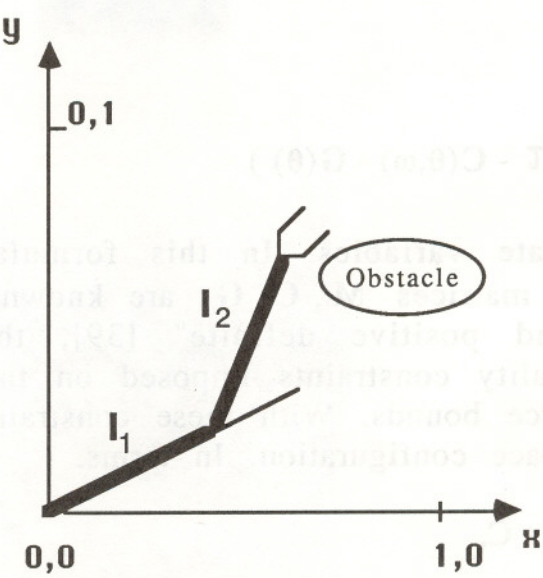

Fig. 2. Avoidable Obstacle

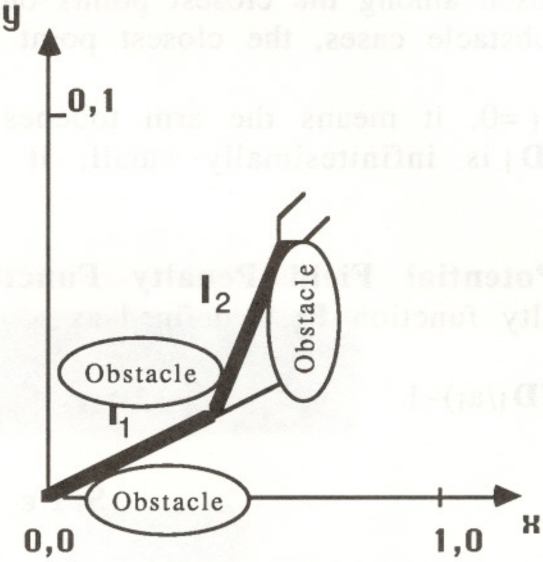

Fig. 3. Unavoidable Obstacles

3.1 States Inequality Constraints

Let $\mathbf{Q_i}$ denotes the i-th obstacle ellipsoid function among m obstacles. The obstacle constraint is:

$$Q_i = a_0(x-x_0)^2 + b_0(x-x_0)(y-y_0) + c_0(y-y_0)^2 + f_0 = 0 \qquad (13)$$

For collision avoidance, it is a must that at all times, for the closest point on the arm,

$$Q_i \geq 0 \qquad (14)$$

3.2 Distance Functions

Distance function $\mathbf{D_i}$ is defined as the function $\mathbf{Q_i}(x,y)$ from the closest point (x, y) on the arm to the i-th obstacle.

The position on each link can be identified by

$$x=x_1+\lambda(x_2-x_1); \qquad y=y_1+\lambda(y_2-y_1) \qquad (15)$$

in terms, x, y are functions of a parameter λ. x_1, y_1, x_2, y_2 are Cartesian coordintes at end points of the links.

Substituting (x, y), $\mathbf{Q_i}(x, y)$ becomes a function of parameter λ. To find the closest point from the arm to the obstacle, we take differentiation and find minimum $\mathbf{Q_i}$ versus λ

$$\frac{dQ_i}{d\lambda} = 0, \qquad (16)$$

then, $\mathbf{D}_i$ is choosen among the closest points on links to the obstacle. In most of the avoidable obstacle cases, the closest point happens to be on the forearm at end-effector.

When $\mathbf{D}_i = 0$, it means the arm touches the i-th obstacle at the closest point. When $\mathbf{D}_i$ is infinitesimally small, it means the arm grazes the i-th obstacle.

3.3 Virtual Potential Field Penalty Function Method (P1)

The penalty function $\mathbf{P}_i$ is defined as

$$\mathbf{S}_i = \exp(\mathbf{D}_i/a_i) - 1 \qquad (17)$$

$$\mathbf{P}_i = \frac{1}{\mathbf{S}_i} \qquad \mathbf{S}_i \geq \varepsilon \qquad (18)$$

$$\mathbf{P}_i = \Gamma \qquad \mathbf{S}_i < \varepsilon \qquad (19)$$

a_i is a small number which denotes the dramatic steepness factor between the trajectory and the i-th obstacle where the penalty becomes active. Γ is a huge number on the edge of the precision boundary that causes computer overflow. ε is a tiny number on the edge of the precision boundary that causes computer underflow. The merit of this infinite penalty function is by choosing a small number a_i, one can define how close the trajectory is allowed to clear the i-th obstacle. By increasing the value a_i, one can supplant the steepness of the penalty function so the trajectory will never get into the obstacles' forbidden area. As **D** increases, **P** sharply decreases, i.e. almost no penalty in farther distance; as **D** decreases, **P** dramatically increases, i.e. a sudden increase of a penalty barrier in the goal function for obstacle avoidance. As soon as **P** dominates the goal function, the problem changes from a minimum-goal one into an obstacle avoidance one. See Fig. 4. as following:

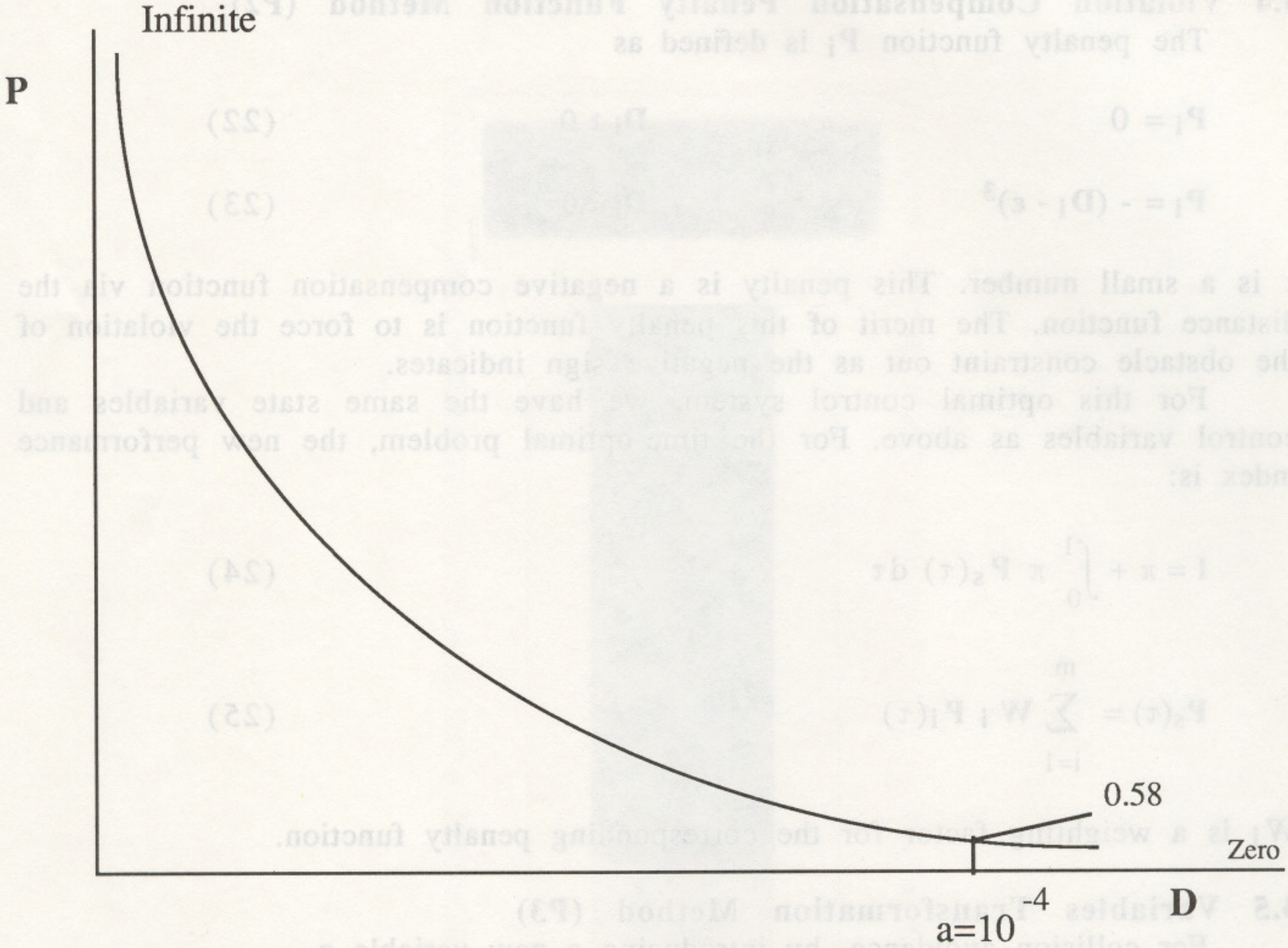

Fig. 4. Penalty function versus distance function diagram

For this optimal control system, we have formulated θ, ω, α as state variables x; u as control variables. For the time-optimal problem, the performance index is:

$$I = \pi + \int_0^1 \pi \, P_s(\tau) \, d\tau \qquad (20)$$

$$P_s(\tau) = \sum_{i=1}^{m} P_i(\tau) \qquad (21)$$

Boundary conditions are initial states x(0), and final states x(1) of specified values. Once the states and the controls are computed, the required reacting torques τ can be solved from Eq. (1).

3.4 Violation Compensation Penalty Function Method (P2)

The penalty function $\mathbf{P_i}$ is defined as

$$\mathbf{P_i} = 0 \qquad \mathbf{D_i} \geq 0 \tag{22}$$

$$\mathbf{P_i} = -(\mathbf{D_i} - \varepsilon)^3 \qquad \mathbf{D_i} < 0 \tag{23}$$

ε is a small number. This penalty is a negative compensation function via the distance function. The merit of this penalty function is to force the violation of the obstacle constraint out as the negative sign indicates.

For this optimal control system, we have the same state variables and control variables as above. For the time-optimal problem, the new performance index is:

$$I = \pi + \int_0^1 \pi \, \mathbf{P_s}(\tau) \, d\tau \tag{24}$$

$$\mathbf{P_s}(\tau) = \sum_{i=1}^{m} \mathbf{W_i} \, \mathbf{P_i}(\tau) \tag{25}$$

$\mathbf{W_i}$ is a weighting factor for the corresponding penalty function.

3.5 Variables Transformation Method (P3)

For collision avoidance, by introducing a new variable z,

$$\mathbf{Q_i} = a_0(x-x_0)^2 + b_0(x-x_0)(y-y_0) + c_0(y-y_0)^2 + f_0 = z^2 \tag{26}$$

$$\dot{z} = [\, a_0(x-x_0)\dot{x} + b_0(x-x_0)\dot{y}/2 + b_0(y-y_0)\dot{x}/2 + c_0(y-y_0)\dot{y} \,]/z \tag{27}$$

where $\dot{x}$, $\dot{y}$, are the time diffrentiation of x, y. We add one or more differential constraints to the control system. For this optimal control system, we have formulated $\theta, \omega, \alpha,$ z as state variables x; u as control variables. For the time-optimal problem, the performance index is: $I = \pi$

3.6 Time Scaling

In the above systems, time has been normalized from $t_{initial}=0$ to $t_{final}=1$ via the following transformation: $t=\pi\tau$, i.e. , $dt=\pi d\tau$. π is a parameter which represents the final time.

3.7 Minimum Distance Problem

For the minimum-distance problem of the end-effector in Cartesian space, the performance index can be replaced by:

$$I = \int_0^1 \pi \left(v_x^2 + v_y^2\right)^{1/2} d\tau + \int_0^1 \pi\, \mathbf{P_s}(\tau)\, d\tau \qquad (28)$$

3.8 Primal Formulation

Optimal control has the characteristic of duality [42, 43]. In this study, the sequential gradient restoration algorithm is employed in conjunction with primal formulation.

4. Optimal Control Theory

The optimal control problem [44] is described in general as follows:

With respect to the vectorial state variable x(t), vectorial control variable u(t) and the vectorial parameter π, the problem of minimizing a functional

$$I = \int_0^1 f(x,u,\pi,t)dt + [h(x,\pi)]_0 + [g(x,\pi)]_1 \qquad (29)$$

subject to differential constraints:

$$\dot{x} - \phi(x,u,\pi,t) = 0, \qquad 0 \le t \le 1, \qquad (30)$$

initial conditions:

$$[\omega(x,\pi)]_0 = 0, \qquad (31)$$

and final conditions:

$$[\psi(x,\pi)]_1 = 0. \qquad (32)$$

where f, h, g, are scalar functions, and ϕ, ω, ψ are vectorial functions of specified dimensions. t is a independent variable. The subscript 0 denotes the initial point, and the subscript 1 denotes the final point.

Optimality Criteria

By introducing the Lagrange multipliers, the problem shown above can be recast as minimizing the augmented functional J

$$J = I + L \tag{33}$$

subject to Eqs. (30-32), where L is the Lagrangian functional

$$L = \int_0^1 \lambda^T(\dot{x} - \phi(x,u,\pi,t))dt + (\sigma^T\omega)_0 + (\mu^T\psi)_1 \tag{34}$$

The symbols $\lambda(t)$, σ, μ denote Lagrange multipliers of appropriate dimensions associated with the constraints. The superscript symbol T denotes the transpose of the matrix.

The first-order optimality criteria originated from Pontryagin Extremum Principle for Eqs. (29-34) can be derived from Euler equations in calculus of variation as:

$$\dot{\lambda} - f_x + \phi_x\lambda = 0, \qquad 0 \le t \le 1, \tag{35}$$

$$f_u - \phi_u\lambda = 0, \qquad 0 \le t \le 1, \tag{36}$$

$$\int_0^1 (f_\pi - \phi_\pi\lambda)dt + (h_\pi + \omega_\pi\sigma)_0 + (g_\pi + \psi_\pi\mu)_1 = 0, \tag{37}$$

$$(-\lambda + h_x + \omega_x\sigma)_0 = 0, \tag{38}$$

$$(\lambda + g_x + \psi_x\mu)_1 = 0. \tag{39}$$

In terms, we seek the functions $x(t)$, $u(t)$, π and the multipliers $\lambda(t)$, σ, μ such that the feasibility Eqs. (30-32) and the optimality criteria Eqs. (35-39) are satisfied to certain numerical accuracy.

5. Algorithm

The sequential gradient restoration algorithm, in either the primal formulation or the dual formulation, is an iterative technique which is constructed by a sequence of two-phase suboptimal cycles. Each cycle includes a gradient phase and a restoration phase. In the gradient phase, the value of the augumented functional is decreased in one step, while avoiding excessive constraint violation. In the restoration phase, the value of the constraint error is decreased in one or multiple steps, while avoiding excessive change in the value of the functional. In a complete gradient-restoration cycle, the value of the functional is decreased, while the constraints are satisfied to a pre-selected degree of accuracy. Therefore, a sequence of suboptimal solutions is generated. Each new suboptimal solution is an improvement of the previous one from the point of view for the value of the functional to be minimized. The optimal solution is reached when the optimality error and the constraint error are both satisfied to a certain accuracy. Schematic diagram is shown in Fig.5.

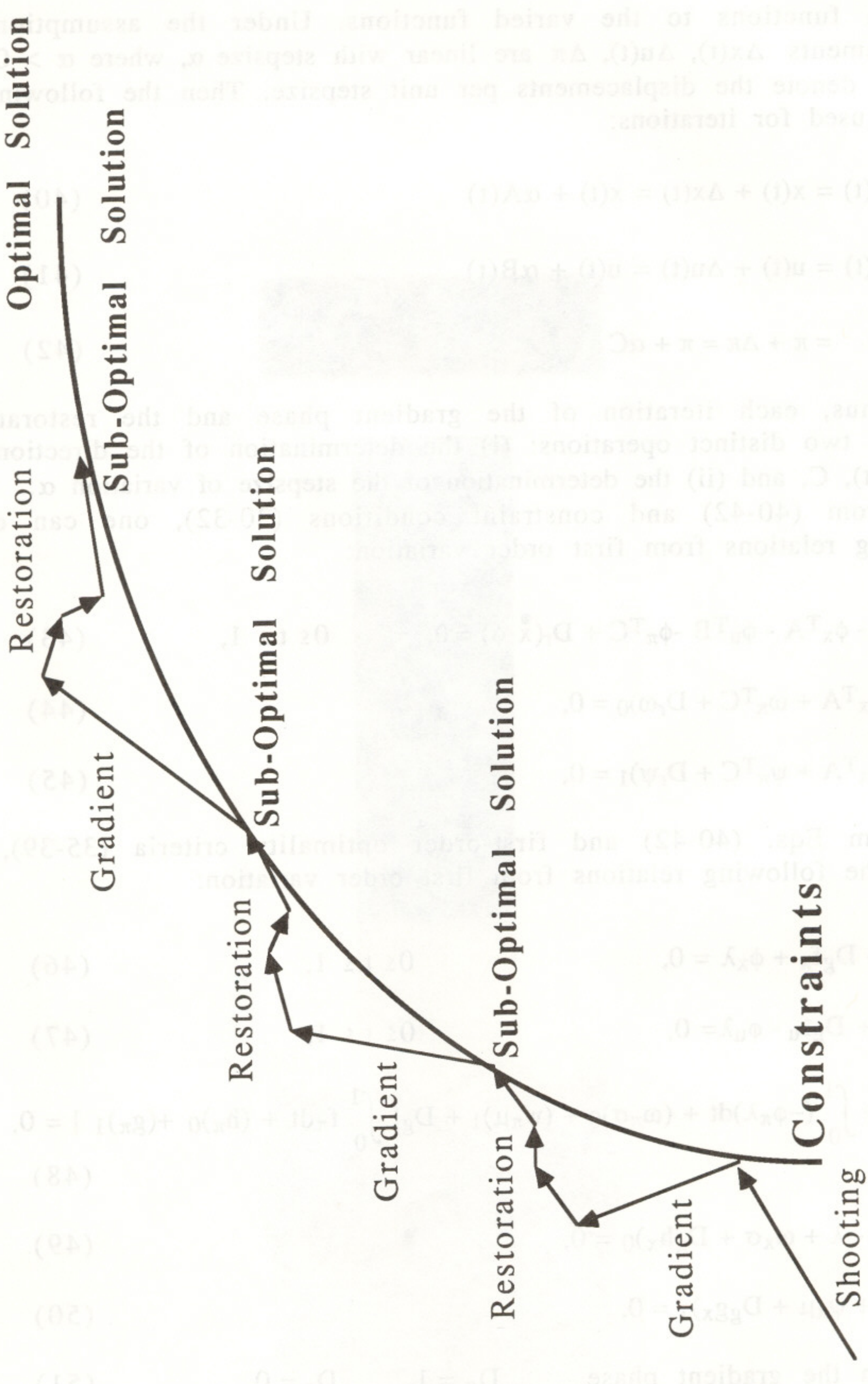

Fig. 5. Sequential Gradient Restoration Algorithm

Let x(t), u(t), π denote the nominal functions; let $\tilde{x}(t)$, $\tilde{u}(t)$, $\tilde{\pi}$, denote the ied functions; let $\Delta x(t)$, $\Delta u(t)$, $\Delta\pi$ denote the displacements leading from the nominal functions to the varied functions. Under the assumption that the displacements $\Delta x(t)$, $\Delta u(t)$, $\Delta\pi$ are linear with stepsize α, where $\alpha > 0$; and $A(t)$, $B(t)$, C denote the displacements per unit stepsize. Then the following relations can be used for iterations:

$$\tilde{x}(t) = x(t) + \Delta x(t) = x(t) + \alpha A(t) \tag{40}$$

$$\tilde{u}(t) = u(t) + \Delta u(t) = u(t) + \alpha B(t) \tag{41}$$

$$\tilde{\pi} \quad = \pi + \Delta\pi = \pi + \alpha C \tag{42}$$

Thus, each iteration of the gradient phase and the restoration phase involves two distinct operations: (i) the determination of the direction functions $A(t)$, $B(t)$, C, and (ii) the determination of the stepsize of variation α.

From (40-42) and constraint conditions (30-32), one can derive the following relations from first order variation:

$$\dot{A} - \phi_x^T A - \phi_u^T B - \phi_\pi^T C + D_r(\dot{x} - \phi) = 0, \qquad 0 \le t \le 1, \tag{43}$$

$$(\omega_x^T A + \omega_\pi^T C + D_r\omega)_0 = 0, \tag{44}$$

$$(\psi_x^T A + \psi_\pi^T C + D_r\psi)_1 = 0, \tag{45}$$

and from Eqs. (40-42) and first-order optimality criteria (35-39), one can derive the following relations from first order variation:

$$\dot{\lambda} - D_g f_x + \phi_x\lambda = 0, \qquad 0 \le t \le 1, \tag{46}$$

$$B + D_g f_u - \phi_u\lambda = 0, \qquad 0 \le t \le 1, \tag{47}$$

$$C + \int_0^1 (-\phi_\pi\lambda)dt + (\omega_\pi\sigma)_0 + (\psi_\pi\mu)_1 + D_g\left[\int_0^1 f_\pi dt + (h_\pi)_0 + (g_\pi)_1\right] = 0, \tag{48}$$

$$(A - \lambda + \omega_x\sigma + D_g h_x)_0 = 0, \tag{49}$$

$$(\lambda + \psi_x\mu + D_g g_x)_1 = 0, \tag{50}$$

where, in the gradient phase, $\qquad D_g = 1, \qquad D_r = 0, \qquad$ (51)

in the restoration phase, $\qquad D_g = 0, \qquad D_r = 1. \qquad$ (52)

The above linear two-point boundary-value problem [LTP-BVP] can be solved for the direction functions A(t), B(t), C, by the method of particular solutions [45, 46].

Stepsize

Eqs. (40-42) define one-parameter functions of the stepsize α. For this parameter, the functionals I, J, P become functions of α as following:

$$\tilde{I} = \tilde{I}(\alpha) \qquad \tilde{J} = \tilde{J}(\alpha) \qquad \tilde{P} = \tilde{P}(\alpha) \tag{53}$$

Then, bisection technique is used for the one-dimension search to find the stepsize, starting from reference stepsize α_g in gradient phase, until (i)

$$\tilde{J}(\alpha) < \tilde{J}(0), \qquad \tilde{P}(\alpha) < P_*, \tag{54}$$

P_* is a preselected number, not necessarily small; and starting from reference stepsize α_r in restoration phase, until (ii)

$$\tilde{P}(\alpha) < \tilde{P}(0) \tag{55}$$

In a complete, successful gradient-restoration cycle, the following condition must be satisfied or the cycle is restarted with reduced stepsize.

$$I_i < I_{i-1} \tag{56}$$

where I_i denotes the value of the functional (29) after current cycle. I_{i-1} denotes the value of the functional (29) after the previous cycle.

Updating suboptimal solution schemes

Once the direction function A(t), B(t), C, and stepsize α are solved, the states, the controls, and the parameters are updated according to Eqs. (40-42).

Summary of Algorithm

Let P be the square norm of the error associated with the feasibility Eqs. (30-32), and Q be the square norm of the error associated with the optimality criteria Eqs. (35-39), then

$$P = \int_0^1 N(\dot{x}-\phi)dt + N(\omega)_0 + N(\psi)_1 \tag{57}$$

$$Q = \int_0^1 N(\dot{\lambda} - f_x + \phi_x\lambda)dt + \int_0^1 N(f_u - \phi_u\lambda)dt$$

$$+ N[\int_0^1 (f_\pi - \phi_\pi \lambda)dt + (h_\pi + \omega_\pi \sigma)_0 + (g_\pi + \psi_\pi \mu)_1]$$

$$+ N(-\lambda + h_x + \omega_x \sigma)_0 + N(\lambda + g_x + \psi_x \mu)_1, \qquad (58)$$

where, N denotes quadratic norm operation.

Thus, numerical convergence for optimal solution can be declared when

$$P \le \varepsilon_1, \qquad (59)$$

$$Q \le \varepsilon_2, \qquad (60)$$

ε_1, ε_2 are preselected, small, positive numbers.

The algorithm is started from providing nominal functions of u(t), and π. The nominal functions can be provided arbitrarily, but good nominals help convergence. The nominal controls are provided with a standard shooting method of Modified Quasilinearization Algorithm, followed by solving the nominal states based on nominal controls, to some accuracy of terminal conditions.

Then, the restoration phase is started. Eqs. (43-45) are solved with (52) and search of stepsize in restoration phase. The one or more iteration restoration phase is completed only until Eq. (59) is satisfied. Then, the gradient phase is started. Eqs. (46-50) are solved with (51) and search of stepsize in gradient phase until Eq. (60) is satisfied for only one iteration. The restoration phase is started again. Thus, a sequence of suboptimal solutions is generated. Each new solution is an improvement of the previous one from the point of view for the value of the functional to be minimized. The optimal solution is reached when Ineqs. (59-60) are both satisfied.

6. Numerical Examples

Numerical examples for time-optimal control with obstacles avoidance schemes of a two-link robotic manipulator are shown in this section. The numerical and analytical solutions of time-optimal control without obstacles can be refered to [22]. The following physical parameters are taken from Asada [46], Sahar and Hollerbach [16] and Zhang and Wang [20]. The obstacle is put on the midway, or even further inward, of the previous no-obstacle optimal trajectory on purpose. The algorithm can be applied to any degree-of-freedom robots with arbitrarily given physical parameters and boundary conditions.

In joint space,

initial position	$(\theta_1, \theta_2)_i = (0.25, 0.35)$	rad.
final position	$(\theta_1, \theta_2)_f = (0.8208, 1.4208)$	rad.
initial velocity	$(\omega_1, \omega_2)_i = (0.0, 0.0)$	rad/sec.
final velocity	$(\omega_1, \omega_2)_f = (0.0, 0.0)$	rad/sec.
acceleration bounds	$(C_1, C_2) = (0.5, 1.0)$	rad/(sec)^2.
gravity constant	$g = 9.8$	m/(sec)^2.

Link 1

mass m_1 = 50 kg, length l_1 = 0.5 m, moment of inertia I_1 = 5.0 kg/(m)^2.

Link 2

mass m_2 = 30 kg, length l_2 = 0.5 m, moment of inertia I_2 = 3.0 kg/(m)^2.

The ellipse obstacle is represented by the following equation:

$Q = a_0(x-x_0)^2 + b_0(x-x_0)(y-y_0) + c_0(y-y_0)^2 + f_0 = 0$; where, $f_0 = -(r_0)^2$

The following symbols are used in the tables:

P1: Virtual Potential Field Penalty Function Method
P2: Violation Compensation Penalty Function Method
P3: Variables Transformation Method
E : Ellipse Obstacle. C : Circle Obstacle, when $b_0 = 0$

Table 1. Comparison of Obstacles Avoidance Schemes

	(P1-C)	(P2-C)	(P3-C)
x_0 (m)	0.5	0.5	0.5
y_0 (m)	0.76	0.76	0.76
r_0 (m)	0.1	0.1	0.1
a_0	1.0	1.0	1.0
b_0	0.0	0.0	0.0
c_0	1.0	1.0	1.0
a (m)	10^{-4}	***	***
ε (m)	***	3.0×10^{-4}	***
W	***	0.5×10^{10}	***
minimum time (sec)	2.914	3.071	4.337

*** denotes there is no such value for the scheme.
2.137 sec is the minimum time without obstacle avoidance.

Table 1. contains the insight of comparison between two penalty function methods and the variables transformation method for circle obstacle (b_0=0). Circle obstacle is centered at (0.5, 0.76) of radius 0.1. Virtual potential field penalty function has only one parameter and generates graze-by trajectory. Violation compensation penalty function is difficult to implemente as a clear collision avoidance scheme owing to the infinite combination of two parameters. Variables transformation method avoids obstacle successfully but is too constrainted when the robot is away from the obstacles to generate true local optima. (Case P1-C in Table 1. is same as P1-C2 in Table 2. and shown in Fig. 13.)

Table 2. Comparison of Different Location of Ellipse and Circle Obstacles.

P1	(E1)	(E2)	(C1)	(C2)
x_0 (m)	0.51	0.50	0.51	0.50
y_0 (m)	0.62	0.76	0.62	0.76
r_0 (m)	0.2	0.2	0.1	0.1
a_0	2.0	2.0	1.0	1.0
b_0	2.0	2.0	0.0	0.0
c_0	2.0	2.0	1.0	1.0
a (m)	10^{-4}	10^{-4}	10^{-4}	10^{-4}
minimum time (sec)	4.332	3.330	3.931	2.914

2.137 sec is the minimum time without obstacle avoidance.

In Table 2., the circle and ellipse obstacle avoidance are listed for side-by-side comparison. As we see, Ellipse (r_0=0.2) is larger in size than Circle (r_0=0.1). Long axis is 45 degree clockwise oriented. Both E1, C1 have the same center location at (0.51, 0.62) and is more close to the robot than E2, C2 which have the same center location at (0.5, 0.76). Owing to the extra journey the obstacle avoidance causes, the bigger the obstacle or the closer the obstacle, the longer the minimum-time is, in trajectory.

For the sake of brief, the results corresponds to the first ellipse (E1) in Table 2. are shown in Figs. 6-11.
Fig. 6 contains the optimal trajectory in minimum time
Fig. 7 contains the joint angle profile in minimum time.
Fig. 8 contains the joint velocity profile in minimum time.
Fig. 9 contains the joint acceleration profile in minimum time.
Fig. 10 contains the torque profile in minimum time.
Fig. 11 contains the distance function profile in minimum time.
Fig. 12 contains the minimum-time trajectory which corresponds to Table 2, P1-E2.
Fig. 13 contains the minimum-time trajectory which corresponds to Table 2, P1-C2.

Table 3. Comparison of Different Radius of Circle Obstacles.

P1	(C1)	(C2)	(C3)
x_0 (m)	1.0	1.0	1.0
y_0 (m)	1.0	1.0	1.0
r_0 (m)	0.6	0.5	0.4
a_0	1.0	1.0	1.0
b_0	0.0	0.0	0.0
c_0	1.0	1.0	1.0
a (m)	10^{-4}	0.5×10^{-4}	10^{-4}
minimum time (sec)	3.829	2.696	2.137

(C3) obstacle is outside the previous no-obstacle optimal trajectory.
2.137 sec is the minimum time without obstacle avoidance.

In Table 3., the circle is moved to center location (1, 1). Radius of the circle is varied as (C1, C2, C3)=(0.6, 0.5, 0.4). In C3 case, the obstacle is outside the work envelop of the robot, so the minimum-time for C3 is the same as the one without obstacle. The minimum-time is longer as the obstacle is bigger in C1, C2 cases.

Fig. 14 contains the optimal trajectory in minimum time which corresponds to Table 3, P1-C3. Owing to the influence of the existing penalty function, there are two curves overlapped on the trajectory. One curve is for previous optimal trajectory without obstacle avoidance scheme, another one is optimal trajectory with obstacles avoidance scheme and obstacle is outside the work envelop.

6.1 Severe Obstacle Avoidance

In joint space,

initial position $(\theta_1, \theta_2)_i$ = (0.349, 0.628) rad.
final position $(\theta_1, \theta_2)_f$ = (0.497, 0.855) rad.
initial velocity $(\omega_1, \omega_2)_i$ = (0.0, 0.0) rad/sec.
final velocity $(\omega_1, \omega_2)_f$ = (0.0, 0.0) rad/sec.

The following case study shows: the arm started near the edge of one side of the obstacle and ended near the edge of another side of the obstacle.

Table 4. Severe Obstacle Avoidance.

	(P1-C)	(P2-C)
x_0 (m)	0.667	0.667
y_0 (m)	0.667	0.667
r_0 (m)	0.1	0.1
a_0	1.0	1.0
b_0	0.0	0.0
c_0	1.0	1.0
a (m)	10^{-3}	***
ε (m)	***	0.2×10^{-1}
W	***	0.5×10^{8}
minimum time (sec)	2.800	2.911

*** denotes there is no such value for the scheme.
1.088 sec is the minimum time without obstacle avoidance.

As we see, minimum-time control is not necessarily related to minimum-distance of the end-effector. The collision avoidance scheme has excellence to move around and avoid severe obstacle. Fig. 15 contains the optimal trajectory in minimum time which corresponds to Table 4, P1-C.

7. Discussion

The insights of the merit of the optimal obstacles avoidance are shown above in Fig. 6-15. All the obstacle avoidance trajectories have the following characteristics: 1) grazing by the obstacle. 2) trying to achieve previous no-obstacle trajectory at near bang-bang control for minimum-time. (At least one joint bang-bang control is the solution for minimum-time without obstacles avoidance) 3) achieving previous no-obstacle optimal trajectory with collision avoidance scheme in the cases of no-obstacles. 4) being able to move around and avoid the severe obstacle.

Virtual potential penalty function method does not cause obstacles constraints violation or over-constrained situation, is the one and only best method. Violation compensation method is difficult to implement owing to the two weighting factors which causes a little obstacle constraint violation from time to time. Variables transformation method is over-constrainted when the obstacles are away from potential collision. This discussion matches the one in [4] even though the implementation of penalty function is different. This approach also illustrates the experimental results for optimization with inequality and/or equality constraints.

As we can see from the comparison tables, the minimum-time of collision-free optimal trajectory is relatively depended on the size and location of the obstacles. The jerk control can be overcome by achieving a near optimal motion in which the trajectory is farther away from the obstacle and the minimum-time is longer.

More intensive research need to be done on minimum distance finding through optical devises or solid modeling. Since the robot manipulators are usually constructed by connected links, in most of the cases, we can say obstacles avoidance for fixed configuration is equivalent to the end-effector obstacle avoidance even though this statement is not true in general. We have to solve end-effector obstacles avoidance before we solve other type of problems because the object is usually on the grip.

The numerical experiments have been done on IBM AS9000 mainfraim and VAX 8800, are also attempted to be done on Macintosh. The CPU time for current research varies from 20 minutes (IBM) to one hour (VAX). The accuracy also varies from machine to machine without very much difference. As the computer industries are growing, the CPU time or accuracy is not a problem for future scientific computation.

The numerical results are constructed by 100 cycles and 300 iterations whichever reached first. The convergence is fast at early stages, it slows down after the sub-optimal solutions come close to the optimal solution. To save computation, one can set up lower limits for cycle, iteration and CPU time, so near-optimal solutions will be generated based on Pontryagin Extremum Principle.

8. Conclusion

In this paper, collision-free optimal motion and trajectory planning for robotic manipulators are solved by a method of sequential gradient restoration algorithm. Numerical examples of a two degree-of-freedom robotic manipulator are demonstrated. The obstacle is put on the midway, or even further inward, of the previous no-obstacle optimal trajectory on purpose. For trying to achieve previous no-obstacle trajectory, the trajectory tangentially grazes by the obstacle and the minimum-time motion successfully avoids the obstacle. The minimum-time is longer for the obstacle avoidance cases than the one without obstacle. All the numerical experiments indicate the obstacles avoidance scheme has the same characteristics which allows the trajectory gets as close to the optimal as possible but barely graze by the obstacle. The weighting and effective point of the penalty can be defined by one parameter which justify the closeness between the trajectory and the obstacle. The trajectory will try to achieve optimization under the obstacles barrier. This is the most outstanding characteristic than other schemes to achieve collision avoidance and also find the optimal motion without extra excursion.

The obstacles avoidance schemes can deal with multiple obstacles in ellipsoid forms by continuously controlling the closest point from the arm to the obstacle using virtual potential fields as penalty functions via distance functions. The algorithm is very promising in solving collision-free optimal control problems for any degree-of-freedom robotic manipulators with any performance indices and mobile robots as well. The minimum-time motion is at least one joint bang-bang control or near bang-bang control with obstacles avoidance, no matter the controls are imposed on angular accelerations or on actuator torques. The minimum-distance trajectory without obstacles is a straight line.

Since this algorithm generates true local minimum solution based on Pontryagin extremum principle, rather than based on approximations, the results provide a benchmark against which any other optimization can be measured.

The perspective research is to investigate the result of optimal solutions for robotic manipulators when the controls are imposed on actuator constraints, and/or with moving obstacles avoidance under different performance indices; and model-reference adaptive optimal feedback control.

Appendix A: Kinematics of a Two-Link Robotic Manipulator

In general, the kinematics, dynamics, control and constraints study of robot can be found in [37]. The kinematical equations are developed by geometrical relationship between Cartesian space and joint space [38]. They can be expressed in general as:

Forward kinematics

$$x(t)=F_1(\theta(t)) \tag{61}$$

$$v(t)=F_2(\theta(t), \omega(t)) \tag{62}$$

$$a(t)=F_3(\theta(t), \omega(t), \alpha(t)) \tag{63}$$

where x(t), v(t), and a(t) are vectors of positions, velocities and accelerations of the end-effector in Cartesian space. $\theta(t)$, $\omega(t)$, and $\alpha(t)$ are vectors of angles, angular velocities and angular accelerations in joint space.
F are functions. For a two-link planar robotic manipulator [40] (Fig.1.):

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} l_1\cos\theta_1+l_2\cos(\theta_1+\theta_2) \\ l_1\sin\theta_1+l_2\sin(\theta_1+\theta_2) \end{pmatrix} \tag{64}$$

$$\begin{pmatrix} v_x \\ v_y \end{pmatrix} = \begin{pmatrix} -l_1\sin\theta_1 - l_2\sin(\theta_1+\theta_2) & -l_2\sin(\theta_1+\theta_2) \\ l_1\cos\theta_1 + l_2\cos(\theta_1+\theta_2) & l_2\cos(\theta_1+\theta_2) \end{pmatrix} \begin{pmatrix} \omega_1 \\ \omega_2 \end{pmatrix} \tag{65}$$

$$\begin{pmatrix} a_x \\ a_y \end{pmatrix} = \begin{pmatrix} -l_1\sin\theta_1 & -l_2\sin(\theta_1+\theta_2) \\ l_1\cos\theta_1 & l_2\cos(\theta_1+\theta_2) \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_1+\alpha_2 \end{pmatrix}$$

$$- \begin{pmatrix} l_1\cos\theta_1 & l_2\cos(\theta_1+\theta_2) \\ l_1\sin\theta_1 & l_2\sin(\theta_1+\theta_2) \end{pmatrix} \begin{pmatrix} \omega_1^2 \\ (\omega_1+\omega_2)^2 \end{pmatrix} \tag{66}$$

Inverse kinematics $\theta(t)=\mathbf{G}_1(x(t))$ (67)

$$\omega(t)=\mathbf{G}_2(x(t),\ v(t)) \tag{68}$$

$$\alpha(t)=\mathbf{G}_3(x(t),\ v(t),\ a(t)) \tag{69}$$

G are functions which depend on the configuration (like: elbow-down). For a two-link robotic manipulator in elbow-down position (Fig.1.):

$$\theta_1=\tan^{-1}\left(\frac{y}{x}\right) - \tan^{-1}\left(\frac{l_2\sin\theta_2}{l_1+l_2\cos\theta_2}\right), \quad \text{where,} \tag{70}$$

$$\theta_2=\cos^{-1}\left(\frac{x^2+y^2-l_1^2-l_2^2}{2l_1l_2}\right) \tag{71}$$

$$\begin{pmatrix} \omega_1 \\ \omega_2 \end{pmatrix} = \frac{1}{l_1l_2\sin\theta_2} \begin{pmatrix} l_2\cos(\theta_1+\theta_2) & l_2\sin(\theta_1+\theta_2) \\ -l_1\cos\theta_1 - l_2\cos(\theta_1+\theta_2) & -l_1\sin\theta_1 - l_2\sin(\theta_1+\theta_2) \end{pmatrix}$$

$$\times \begin{pmatrix} v_x \\ v_y \end{pmatrix} \tag{72}$$

$$\begin{pmatrix} \alpha_1 \\ \alpha_1+\alpha_2 \end{pmatrix} = \frac{1}{l_1l_2\sin\theta_2} \begin{pmatrix} l_2\cos(\theta_1+\theta_2) & l_2\sin(\theta_1+\theta_2) \\ -l_1\cos\theta_1 & -l_1\sin\theta_1 \end{pmatrix} \begin{pmatrix} a_x \\ a_y \end{pmatrix}$$

$$+ \frac{1}{l_1l_2\sin\theta_2} \begin{pmatrix} l_1l_2\cos\theta_2 & l_2^2 \\ -l_1^2 & -l_1l_2\cos\theta_2 \end{pmatrix} \begin{pmatrix} \omega_1^2 \\ (\omega_1+\omega_2)^2 \end{pmatrix} \tag{73}$$

Reference

1. Holcomb, L.B., Montemerlo, M.D., "The NASA Automation and Robotics Technology Program", Acta Astronautica, Vol. 16, 1987, pp. 11-18.
2. Lerner, E.J., "Robots on the Space Station", Aerospace America, June 1987, pp. 42-45.

3. Kahn, M.E., Roth, B., "The Near-Minimum-Time Control of Open-Loop Articulated Kinematic Chains", Journal of Dynamic Systems, Measurement, and Control, Vol. 93, 1971, pp. 164-172.
4. Gilbert, E.G., Johnson, D.W. "Distance Functions and Their Application to Robot Path Planning in the Presence of Obstacles" IEEE Journal of Robotics and Automation, Vol. RA-1, No. 1, March, 1985, pp. 21-30.
5. Geering, H.P., Guzzella, L., Hepner, S.A.R., Onder, C.H., "Time-Optimal Motions of Robots in Assembly Tasks", IEEE Transactions on Automatic Control, Vol. AC-31, 1986, pp. 512-518.
6. Kim, Y.T., Jamshidi, M., Shahinpoor, M., "Near-Optimum Control of a Robot Manipulator", International Journal of Robotics and Automation, Vol. 2, 1987, pp. 15-20.
7. Ozaki H., Mohri, A., "Planning of Collision-Free Movements of a Manipulator with Dynamic Constraints", Robotica, Vol. 4, 1986, pp. 163-169.
8. Vukobratovic, M., Kircanski, M., "A Method for Optimal Synthesis of Manipulation Robot Trajectories", Journal of Dynamic Systems, Measurement, and Control, Vol. 104, 1982, pp. 188-193.
9. Singh, S., Leu, M.C., "Optimal Trajectory Generation for Robotic Manipulators Using Dynamic Programming", Journal of Dynamic Systems, Measurement, and Control, Vol. 109, 1987, pp. 88-96.
10. Shin, K.G., McKay, N.D., "A Dynamic Programming Approach to Trajectory Planning of Robotic Manipulators", IEEE Transactions on Automatic Control, Vol. AC-31, 1986, pp. 491-500.
11. Shin, K.G., McKay, N.D., "Minimum-Time Control of Robotic Manipulators with Geometric Path Constraints", IEEE Transactions on Automatic Control, Vol. AC-30, No. 6, 1985, pp. 531-541.
12. Shin, K.G., McKay, N.D., "Selection of Near-Minimum Time Geometric Paths for Robotic Manipulators", IEEE Transactions on Automatic Control, Vol. AC-31, 1986, pp. 501-511.
13. Bobrow, J.E., Dubowsky, S., Gibson, J.S., "On the Optimal Control of Robotic Manipulators with Actuator Constraints", Proceedings of the American Control Conference, San Francisco, California, June 22-24, 1983, pp. 782-787.
14. Dubowsky, S., Norris, M.A., Shiller, Z., "Time Optimal Trajectory Planning for Robotic Manipulators with Obstacle Avoidance: A CAD Approach" IEEE International Conference on Robotics and Automation, San Francisco, CA, April 1986, pp. 1906-1912.
15. Rajan, V.T., "Minimum Time Trajectory Planning", Proceedings of the IEEE International Conference on Robotics and Automation, St. Lious, Missouri, March 25-28, 1985, pp. 759-764.
16. Sahar, G., Hollerbach, J.M., "Planning of Minimum-Time Trajectories for Robot Arms", The International Journal of Robotics Research, Vol. 5, No. 3, 1986, pp. 90-100.

17. Luh, J.Y.S., Lin, C.S., "Optimum Path Planning for Mechanical Manipulators", Journal of Dynamic Systems, Measurement, and Control, Vol. 103, June 1981, pp. 142-151.
18. Weinreb, A., Bryson, A.E., Jr., "Minimum-Time Control of a Two-Link Robot Arm", IFAC Control Applications of Nonlinear Programming and Optimization, Capri, Italy, 1985, pp. 195-199.
19. Meier E.B., Bryson A.E., Jr., "An Efficient Algorithm for Time Optimal Control of a Two-Link Manipulator", AIAA Conference on Guidance and Control, Monterey, CA, August 1987, pp. 204-212.
20. Zhang, W., Wang, P.K.C., "Collision-Free Time-Optimal Control of a Two-Link Manipulator", International Journal of Robotics and Automation, Vol. 1, No. 3, 1986, pp. 96-104.
21. Bobrow, J.E., "Optimal Robot Path Planning Using the Minimum-Time Criterion" IEEE Journal of Robotics and Automation, Vol. 4, No. 4, August 1988, pp. 443-450.
22. Wang, J.J., "Optimal Control of Robotic Manipulators", International Journal of Robotics and Automation, Vol. 4, No. 1, 1989, pp. 27-35.
23. Vetterling W.T., Teukolsky S.A., Press, W.H., Flannery, B.P., Numerical Recipes: The Art of Scientific Computing, Cambridge University Press, 1988.
24. Kelley, H. J., "Gradient Theory of Optimal Flight Paths", ARS Journal, Vol. 30, No. 10, 1960.
25. Gonzalez, S., Miele, A., "Sequential Gradient Restoration Algorithm for Optimal Control Problems with General Boundary Conditions", Journal of Optimization Theory and Applications, Vol. 26, No. 3, 1978, pp. 395-425.
26. Khatib, O., "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots", The International Journal of Robotics Research, Vol. 5, No. 1, 1986, pp. 90-98.
27. Miele, A., Venkataraman, P., "Optimal Trajectories for Aeroassisted Orbital Transfer", Acta Astronautica, Vol. 11, Nos. 7-8, 1984, pp. 423-433.
28. Miele, A., Basapur, V.K., Mease, K.D., "Nearly-Grazing Optimal Trajectories for Aeroassisted Orbital Transfer", Journal of the Astronautical Sciences, Vol. 34, No. 1, 1986, pp. 3-18.
29. Miele, A., Basapur, V.K., Lee, W.Y., "Optimal Trajectories for Aeroassisted Coplanar Orbital Transfer", Journal of Optimization Theory and Applications, Vol. 52, No. 1, 1987, pp. 1-24.
30. Miele, A., Wang, T., Melvin, W.W., "Optimal Take-Off Trajectories in the Presence of Windshear", Journal of Optimization Theory and Applications, Vol. 49, No. 1, 1986, pp. 1-45.
31. Wang, J.J., "Near-Optimal Guidance for Abort Landing Flight in a Windshear", M.S. Thesis, Rice University, Houston, TX, December 1987.
32. Miele, A., Wang, T., Wang, H., Melvin, W.W., "Optimal Penetration Landing Trajectories in the Presence of Windshear", Journal of Optimization Theory and Applications, Vol. 57, No. 1, 1988, pp. 1-40.
33. Lam., T.T., Bayazitoglu, Y., "Solution to the Orr-Sommerfeld Equation for Liquid Film Flowing Down an Inclined Plane: An Optimal Approach",

International Journal for Numerical Methods in Fluids, Vol. 6, 1986, pp. 883-894.

34. Bayazitoglu, Y., Lam., T.T., "Marangoni Convection in Radiating Fluids", Journal of Heat Transfer, Vol. 109, No. 3, 1987.
35. Wang, J.J., "Optimal Control of Robotic Manipulators with Actuator Constraints", AAC Professionals, Science, Engineering and Technology Seminars, Houston, TX, May 28-29, 1988
36. Wang, J.J., "Optimal Control of Robotic Manipulators with Obstacles Avoidance", SIAM Conference on Control in the 90's, San Francisco, CA, May 17-19, 1989
37. Vukobratovic, M., et al, "Introduction to Robotics", Springer-Verlag, New York, 1989.
38. Wolovich, W.A., Robotics: Basic Analysis and Design, CBS College Publishing, New York, NY, 1987.
39. Brady, M., Hollerbach, J.M., Johnson, T.L., Lozano-Perez, T., Mason, M.T., Robot Motion: Planning and Control, The MIT Press, Cambridge, Massachusetts, 1982.
40. Cheng, P.Y., Weng, C.I., Chen, C.K., "Symbolic Derivation of Dynamic Equations of Motion for Robot Manipulators Using Piogram Symbolic Method", IEEE Journal of Robotics and Automation, Vol. 4, No. 6, December 1988, pp. 599-609.
41. Rockafellar, R.T., "Duality in Optimal Control", Mathematical Control Theory, Edited by W. A. Coppel, Published by Springer-Verlag, New York, NY, 1978, pp. 219-257.
42. Miele, A., Wang, T., "Primal-Dual Properties of Sequential Gradient Restoration Algorithms for Optimal Control Problems", Part 2, General Problems, Journal of Mathematical Analysis and Applications, Vol. 119, Nos. 1-2, 1986, pp. 21-54.
43. Bryson, A.E., Jr., Ho, Y.C., Applied Optimal Control, Blaisdell Publishing Company, Waltham, Massachusetts, 1969.
44. Miele, A., "Method of Particular Solutions for Linear, Two-Point Boundary-Value Problems, Journal of Optimization Theory and Applications, Vol. 2, No. 4, 1968, pp. 260-273.
45. Miele, A., Iyer, R.R., "General Technique for Solving Nonlinear, Two-Point Boundary-Value Problems via the Method of Particular Solutions", Journal of Optimization Theory and Applications, Vol. 5, No. 5, 1970, pp. 382-399.
46. Asada, H., "Dynamic Analysis and Design of Robot Manipulators Using Inertia Ellipsoids", Proceedings of the IEEE Computer Society 1st International Conference on Robotics, Atlanta, Georgia, March 13-15, 1984, pp. 94-102.
47. Fu, K.S., Gonzalez R.C., Lee C.S.G., Robotics: "Control, Sensing, Vision, Intelligence", McGraw-Hill Publishing Company, NY, 1986
48. Lee, A.Y., Bryson, A.E., Jr., Hindson, W.S., "Optimal Landing of a Helicopter in Autorotation", Journal of Guidance, Control, and Dynamics, Vol. 11, No. 1, 1988, pp. 7-12

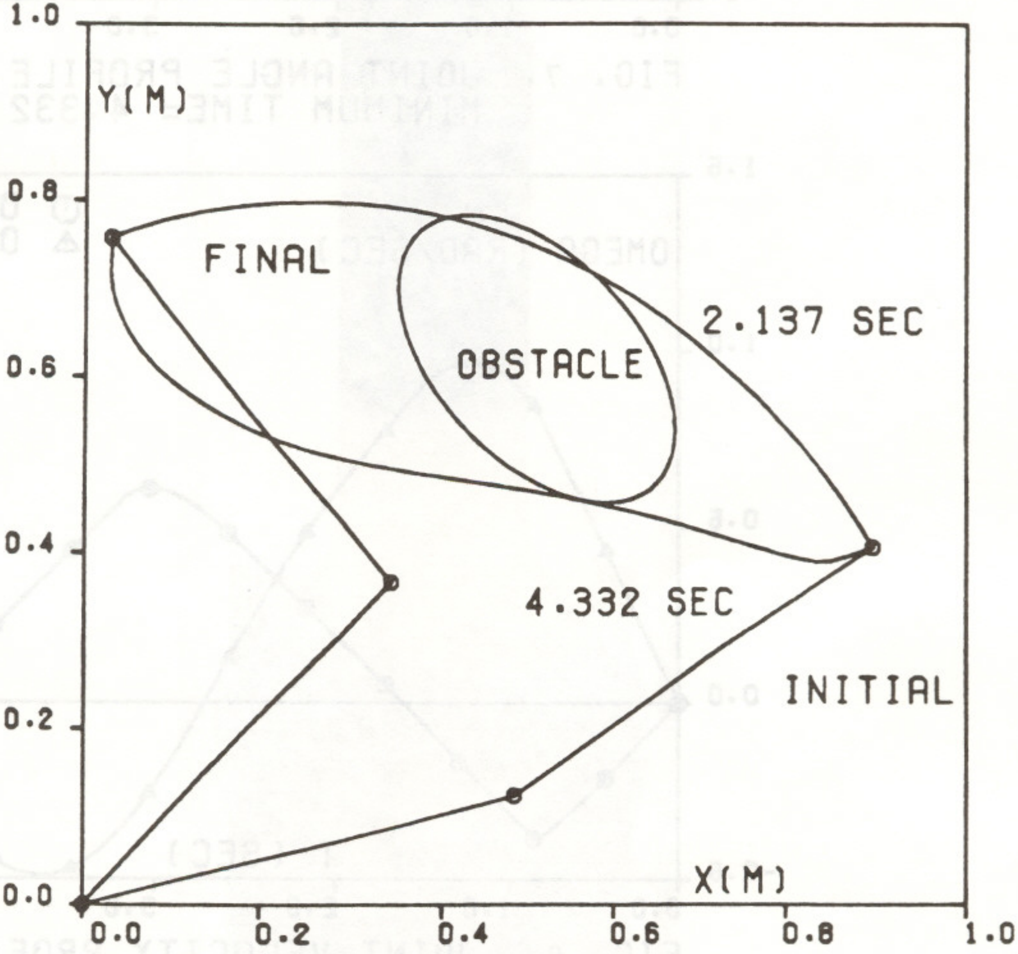

FIG. 6. OPTIMAL TRAJECTORY,
MINIMUM TIME= 4.332 SEC .

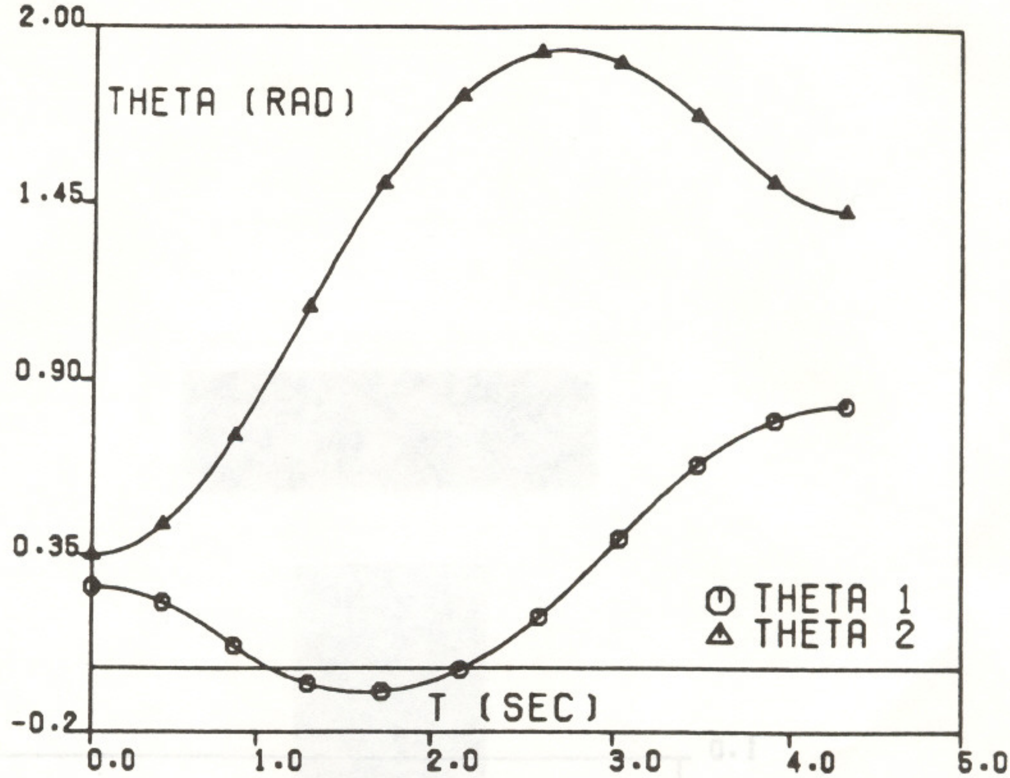

FIG. 7. JOINT ANGLE PROFILE,
MINIMUM TIME= 4.332 SEC .

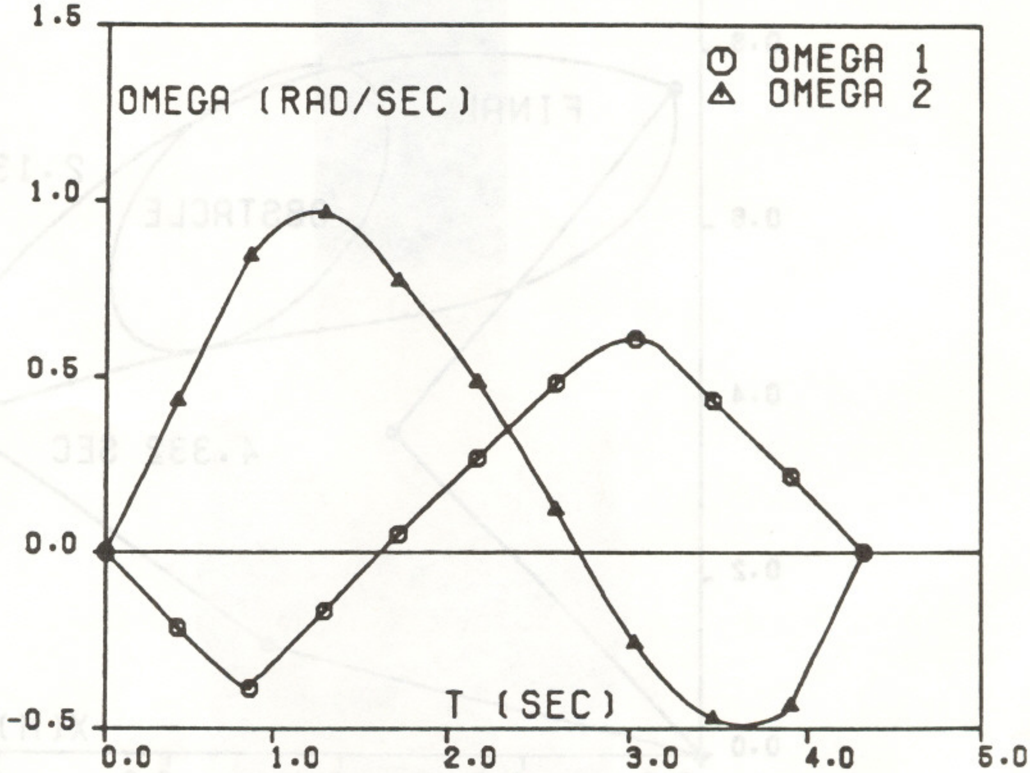

FIG. 8. JOINT VELOCITY PROFILE
MINIMUM TIME= 4.332 SEC .

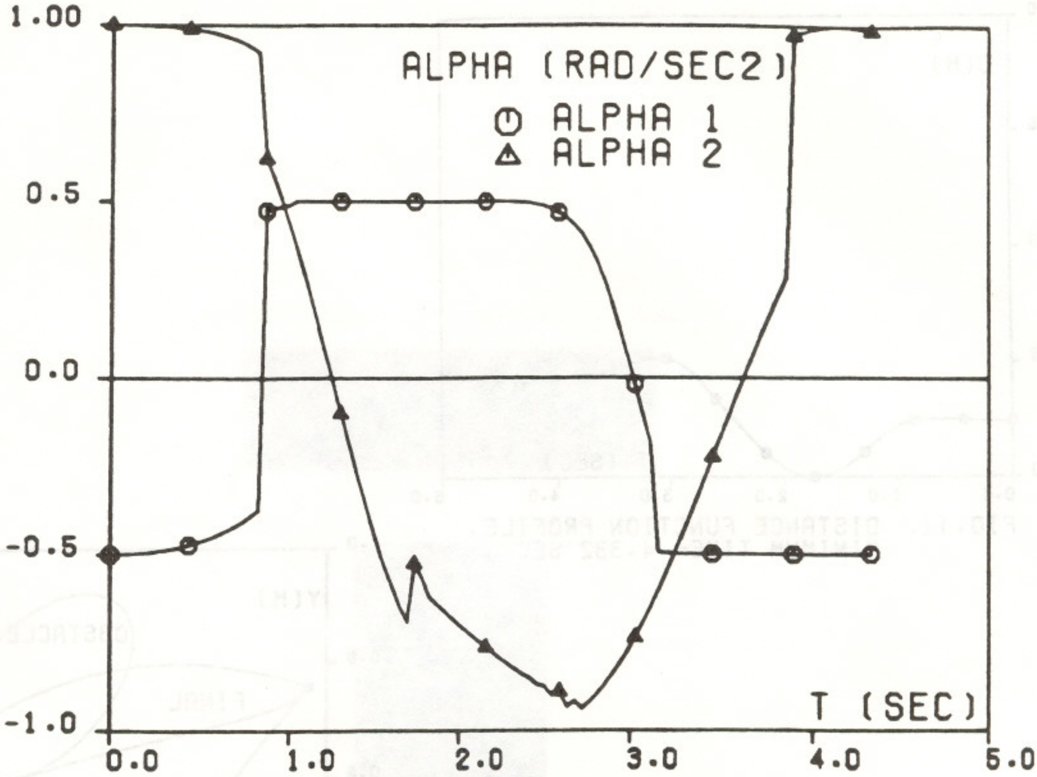

FIG. 9. JOINT ACCELERATION PROFILE, MINIMUM TIME= 4.332 SEC .

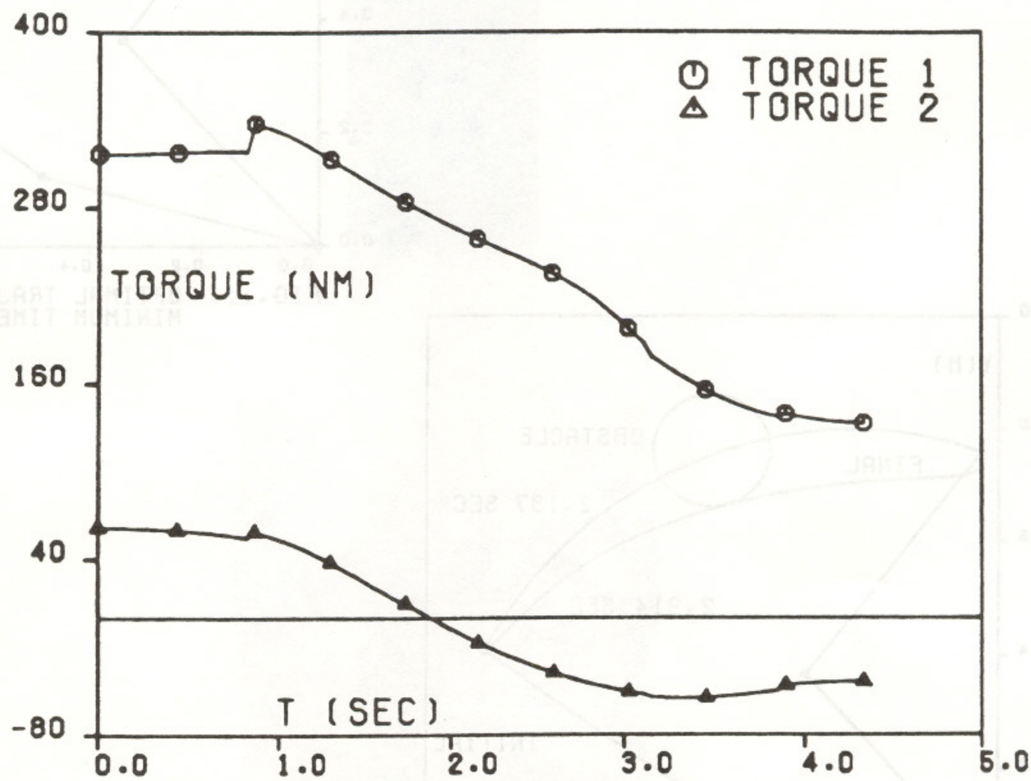

FIG. 10. TORQUE PROFILE, MINIMUM TIME= 4.332 SEC .

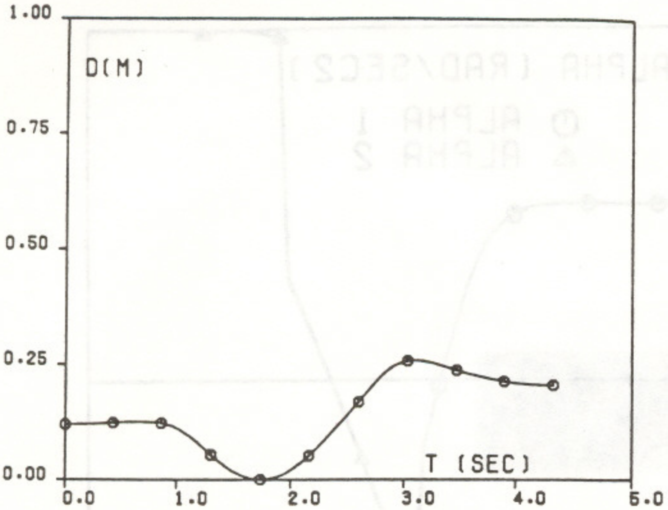

FIG. 11. DISTANCE FUNCTION PROFILE,
MINIMUM TIME= 4.332 SEC .

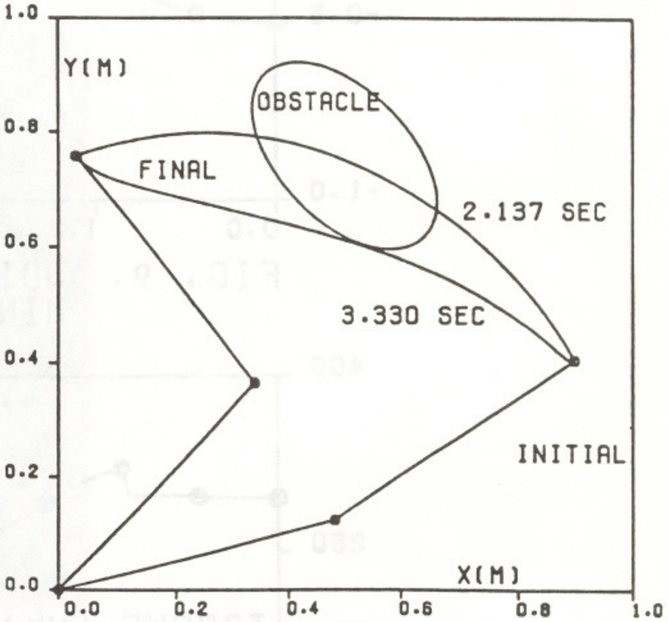

FIG. 12. OPTIMAL TRAJECTORY,
MINIMUM TIME= 3.330 SEC .

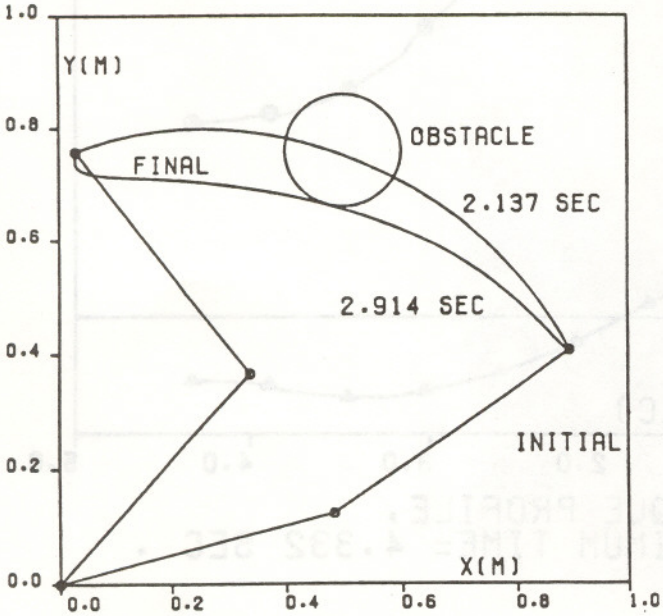

FIG. 13. OPTIMAL TRAJECTORY,
MINIMUM TIME= 2.914 SEC .

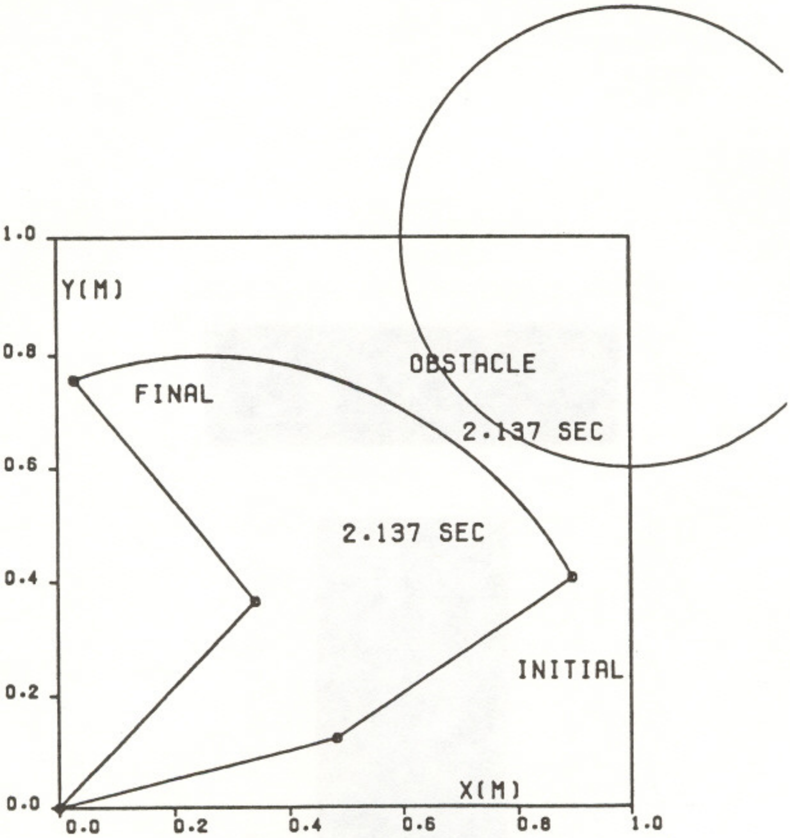

FIG.14. OPTIMAL TRAJECTORY,
MINIMUM TIME= 2.137 SEC .

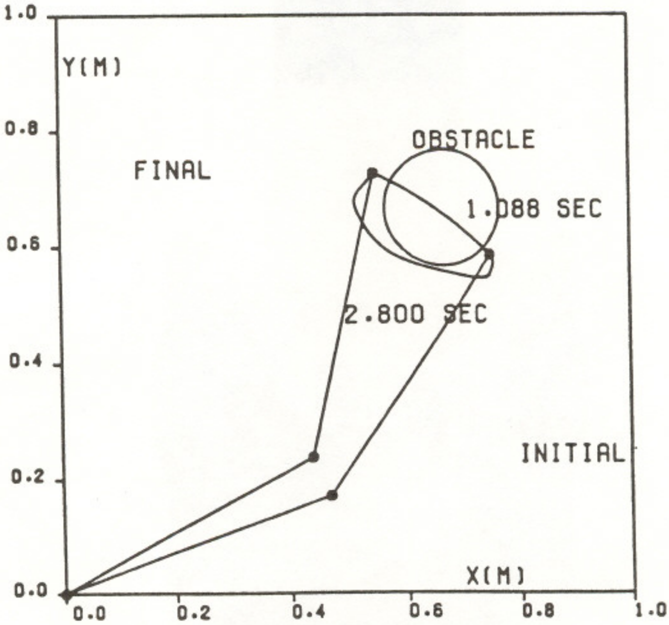

FIG.15. OPTIMAL TRAJECTORY,
MINIMUM TIME= 2.800 SEC .