JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 17, No. 4, July–August 1994

Nonsmooth Trajectory Optimization: An Approach Using Continuous Simulated Annealing

Ping Lu* and M. Asif Khan†
Iowa State University, Ames Iowa 50011

A continuous simulated annealing algorithm is introduced as a new global trajectory optimization tool for nonsmooth dynamic systems. Its properties are discussed. The algorithm is implemented in a trajectory optimization program. The difficult problem of nonsmooth trajectory optimization for a high-performance, rigid-body aircraft is successfully solved using this approach. The results show that the simulated annealing algorithm outperforms some other well-known conventional algorithms by a large margin.

I. Introduction

WITH the increasing demands on high performance of dynamic systems, the problem of finding an optimal solution to a control problem for a physical system has become a routine challenge to control engineers. Although optimal control theory has been well developed, most of today's optimal control problems, or alternatively referred to as trajectory optimization problems, are converted into mathematical programming problems because many algorithms exist that are well suited for such problems. Examples of general purpose trajectory optimization programs of this class include Program to Optimize Simulated Trajectories (POST)[1] and Optimal Trajectories by Implicit Simulation (OTIS).[2] Other programs based on sequential quadratic programming (SQP) techniques (e.g., Ref. 3) have also proved effective. Because all of these algorithms use gradient information for the search for a local optimum, the optimization problem needs to be at least once continuously differentiable. This assumption may, however, not be true for many complicated systems. For instance, the realistic model of an aerospace vehicle typically consists of a large amount of tabulated aerodynamic and propulsion data. Multidimensional linear interpolations are usually used for table look-up. In addition, actuator displacement and rate limits are generally present in a system, and the optimal solution frequently calls for maximum allowable control actions in certain intervals. Other internal nonlinearities such as dead-band and hysteresis often exist in a real system. All of these factors make trajectory optimization for such a system a continuous but nondifferentiable optimization problem. The application of one of the algorithms developed for smooth problems to such a nonsmooth problem will at best result in a solution the optimality of which is highly questionable. It is probable that the algorithm will fail to converge because of incorrect gradient information. Nongradient-type algorithms[4,5] are available; however, for the most part, they are very inefficient when the number of optimization parameters becomes moderately large. Often they fail to find even a local optimum for difficult problems if the initial estimates are not sufficiently close.

Recently, the genetic algorithm[6] (GA) has emerged as a promising tool in control and trajectory optimization.[7–8] The genetic algorithm generates a search procedure based on the natural law of survival-of-the-fittest. It does not require the gradients of the problem, and it is robust and capable of finding the global optimum. On the other hand, the genetic algorithm depends on educated choices of mutation and crossover rates and properly selected initial population to prevent premature termination of the algorithm; high resolution of the solution and increased number of optimization parameters tend to increase the computation time significantly. Although numerical experiences have indicated that genetic algorithms can find the global optimum, no theoretical study is available yet to prove the convergence. In this paper, we introduce a recently developed continuous simulated annealing algorithm as a new tool to trajectory optimization. It belongs to another class of nongradient global optimization algorithms, and is robust, guaranteed to converge to the global optimum with probability one. The application of this simulated annealing algorithm to trajectory optimization for a 6-DOF high-performance fighter aircraft demonstrates the superiority of the algorithm.

This paper is organized in the following structure. In Sect. II, simulated annealing algorithms are reviewed. In particular, a recently developed continuous simulated annealing algorithm, Hide-and-Seek, is introduced. A nonsmooth trajectory optimization problem for an advanced fighter aircraft is formulated in Sect. III. A realistic, nonlinear, rigid-body model is used for the aircraft. Optimal trajectories are obtained successfully by using Hide-and-Seek, whereas some conventional algorithms are shown either to have failed or yielded only local optimal solutions. Section IV summarizes the work.

II. Simulated Annealing

Simulated annealing (SA) is a class of stochastic optimization algorithms for the following problem

$$\min_{x \in S} f(\boldsymbol{x}) \tag{1}$$

where the feasible region $S \subset \boldsymbol{R}^n$ is a compact set, and f a continuous function defined on S. The problem is to find an $\boldsymbol{x}^* \in S$ so that $f^* \triangleq f(\boldsymbol{x}^*) \leq f(\boldsymbol{x})$ for all $\boldsymbol{x} \in S$. The algorithm searches for a global optimum by simulating the physical phenomenon of annealing.

A. Past Development and Applications

Simulated annealing algorithms were developed originally for discrete combinatorial optimization problems. A combinatorial optimization problem is one in which the design vector $\boldsymbol{x}$ has finite or countably infinite configurations.[9] The simulated annealing algorithm essentially imitates physical annealing. Physical annealing is a process by which a solid is first heated until it melts and then gradually cooled until it crystallizes into a state with a perfect lattice. The final state this process attains is a configuration that minimizes the free energy of the solid. At any given temperature T, the probability of a system being in a state r is given by the Boltzmann's distribution

Received Dec. 2, 1992; revision received Oct. 1, 1993; accepted for publication Oct. 28, 1993. Copyright © 1993 by the American Institute of Aeronautics and Astronautics, Inc. All rights reserved.

*Assistant Professor, Department of Aerospace Engineering and Engineering Mechanics. Senior Member AIAA.

†Graduate Research Assistant. Student Member AIAA.

$$P(r) \sim e^{-E(r)/k_bT}$$

where $E(r)$ is the energy associated with state r, and k_b is the Boltzmann's constant.[10] Thus, at equilibrium, the most probable state is that of the lowest energy. Simulated annealing uses this principle of physical annealing. The state of a system corresponds to the design vector $\boldsymbol{x}$; energy to the value of the cost function $f(\boldsymbol{x})$. The Boltzmann's distribution is replaced by the Metropolis criterion[11]

$$\beta_T = \min(1.0, e^{f(\boldsymbol{x}_2)-f(\boldsymbol{x}_1)/T}) \tag{2}$$

where $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are two different design points. The use of β_T is briefly described shortly. Simulated annealing avoids getting trapped in local optima by probabilistically accepting transitions corresponding to a deterioration in cost function value. These deteriorations make it possible to move away from local optima and explore the feasible region S in its entirety. As the optimization process progresses, the probability of accepting cost function deteriorations slowly reduces to zero.

Past applications of simulated annealing have been mainly in discrete optimization problems such as telephone network and integrated circuits design, test pattern generation, image restoration, and statistical optimization. Kirkpatrick et al. used an SA algorithm to solve the famous traveling salesman problem[12] in which the shortest itinerary is to be found for a salesman who must visit N cities in turn. The dimension of this problem is extraordinary because for only 20 cities the number of feasible tours is on the order of 10^{18}. Using a SA algorithm, Kirkpatrick et al. solved this problem for 400 cities. Genetic algorithms and simulated annealing algorithms have been compared at different levels.[13,14]

B. Hide-and-Seek Algorithm

For application in trajectory optimization for a continuous dynamic system, a continuous SA algorithm is required. One of the major differences between a discrete and continuous SA algorithm is the cooling schedule for the temperature parameter, which parametrizes the decrease of acceptance probabilities for deteriorations. For discrete SA, conditions for a cooling schedule that guarantees convergence to the global optimum are known.[15] Continuous SA algorithms have been proposed in the past, but most of them lack solid theoretical foundation. More recently, an SA algorithm for continuous optimization (maximization), called Hide-and-Seek, has been developed.[16] This algorithm has two distinct features: an adaptive cooling schedule and a continuous random walk process for generating a sequence of feasible points. Convergence of the algorithm to the global optimum is rigorously proved. The user supplies the bounds on the design vector. Within the bounded design space, the feasible region is specified by criteria set up by the user, and disjoint feasible regions are allowed. Another property of Hide-and-Seek is that the number of function evaluations is increased only linearly with the dimension of the problem, which makes it an attractive algorithm for large-scale optimization problems.

Hide-and-Seek proceeds roughly as follows.[16] The starting point, x_0, is generated randomly and a large initial temperature, T_0, is selected. In the kth step, a direction, θ_k, on the surface of the unit sphere in the search space is chosen from the uniform distribution. Then choose λ_k from the uniform distribution $\Lambda_k = (\lambda \in R : \boldsymbol{x}_k + \lambda\theta_k \in S)$. Set $\boldsymbol{y}_k + 1 = \boldsymbol{x}_k + \lambda_k\theta_k$. The next search point, $\boldsymbol{x}_{k+1}$, is determined by

$$\boldsymbol{x}_{k+1} = \begin{cases} \boldsymbol{y}_{k+1} & \text{if} \quad V_k \in [0, \beta_{T_k}(\boldsymbol{x}_k, \boldsymbol{y}_{k+1})], \\ \boldsymbol{x}_k & \text{if} \quad V_k \in [\beta_{T_k}(\boldsymbol{x}_k, \boldsymbol{y}_{k+1}), 1] \end{cases}$$

where V_k is a random variable with uniform distribution on [0, 1]; and T_k is the current temperature. It should be noted that from the above equation, even if $f(\boldsymbol{y}_{k+1})$ represents a deterioration in the objective function [i.e., $f(\boldsymbol{y}_{k+1}) < f(\boldsymbol{x}_k)$], the probability of acceptance of $\boldsymbol{y}_{k+1}$ as the next iteration point is high if the temperature T_k is high. T_k is updated (decreased) by the cooling schedule

$$T_k = 2[f^* - f(\boldsymbol{x}_k)]/\chi^2_{1-p}(n)$$

only when $f(\boldsymbol{x}_k)$ is greater than all previous objective function values, where $0 < p < 1$ and $\chi^2{}_{1-p}(n)$ is the $100(1 - p)$ percentile point of the chi-square distribution with n degree-of-freedom. This cooling schedule generates the next point that would give an improvement in function value over the current iteration point with probability at least p. The value of p for the results reported in this paper is set to 0.1. However, several other values of p were also tested, and the performance of the algorithm is found to be insensitive to different choices of p. When f^* is not known, the authors of Hide-and-Seek have developed a heuristic estimator $\hat{f}$ for f^*

$$\hat{f} = f_1 + \frac{f_1 - f_2}{(1 - p)^{-n/2} - 1} \tag{3}$$

where f_1 and f_2 are the current two largest function values. The algorithm stops when the difference between the current function value and f^* (or $\hat{f}$) is less than a prescribed value. (We refer the interested reader to Ref. 16 for more details and discussion of the algorithm.)

Hide-and-Seek is very easy to implement and use. The implementation of the algorithm itself takes a subroutine of only a few hundred lines in FORTRAN. Using it as an integral part of a trajectory optimization program, the user need only provide the cost function value at the current search point and feasibility check-up. Because all calculations can be done in double precision without extra computation burden, resolution of the solution is not a concern. On the other hand, the convergence rate of the algorithm does not accelerate even when the search point is near the global optimum, thus the algorithm is more expensive in terms of function evaluations than gradient-type algorithms. Also, when f^* is not known in advance, the use of $\hat{f}$ may result in even more function evaluations for convergence.

Some numerical experiments conducted in Ref. 16 have shown that Hide-and-Seek significantly outperforms two multistart global optimization schemes in efficiency. Reference 17 compared Hide-and-Seek with two well-known conventional nongradient algorithms: Principal Axis,[4] which is based on Powell's method; and Nelder-Mead Simplex,[5] on some difficult test functions. Hide-and-Seek consistently performs much better. To illustrate this, consider the following test function[18]

$$f(x_1, x_2, \ldots, x_n) = -\frac{\pi}{n}\left\{10\sin^2(\pi x_1) + \sum_{i=1}^{n-1}(x_i - 1)^2\right.$$
$$\left.[1 + 10\sin^2(\pi x_{i+1})] + (x_n - 1)^2\right\}$$

For $-10 \le x_i \le 10, i = 1, \ldots, n$, the function has approximately 10^n local maxima, and one global maximum $f^* = 0$ at $x_i = 1$. A surface plot of the function with $n = 2$ is shown in Fig. 1. Tests were done using Hide-and-Seek for $n = 2$, 3, and 4. Table 1 summarizes the average function evaluations of ten runs for each case. Every run found the global optimum. The use of $\hat{f}$, the estimation of f^*, in the cooling schedule was also tested. It is seen that with f^* known, the number of function evaluations increases almost linearly with the problem dimension. With $\hat{f}$, the linearity is no longer true. Principal Axis and Nelder-Mead methods failed to find the global optimum unless the starting points were chosen to be extremely close.

III. Trajectory Optimization for a Rigid-Body Aircraft

The problem of finding an optimal flight path for an aircraft has been studied extensively. The classic work by Bryson and

Table 1 Test function evaluations

n	With f^*	With $\hat{f}$
2	3,068	11,213
3	5,191	13,738
4	7,663	6,616

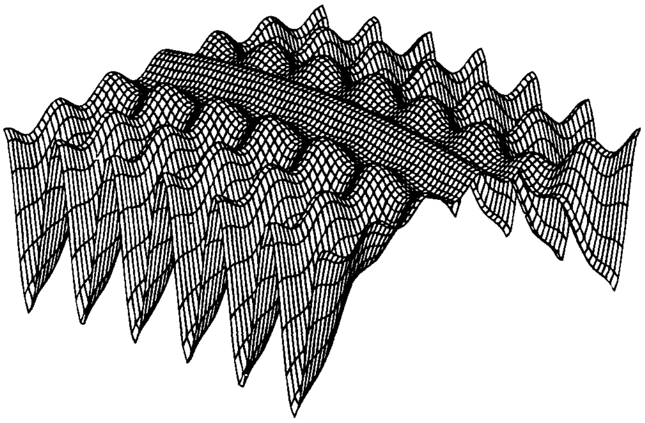

Fig. 1 Surface plot of the test function.

Table 2 Aircraft model parameters

Attribute	Symbol	Value
Weight	W	45,000.0 lb
Wing area	S	608.0 ft^2
Wing span	b	42.8 ft
Mean chord	$\bar{c}$	15.95 ft
Moments of inertia	I_x	28,700.0 slug/ft^2
	I_y	165,100.0 slug/ft^2
	I_z	187,900.0 slug/ft^2
Products of inertia	I_{xz}	−520.0 slug/ft^2
	I_{xy}	0.0 slug/ft^2
	I_{yz}	0.0 slug/ft^2

Denham[19] is very well known. Numerous investigations have been reported in the literature since then. However, most of the work involving trajectory optimization is limited to point-mass aircraft models. Because rotational dynamics are equally important for the flight of an aircraft but not included by point-mass models, a complete study of optimal maneuvers requires consideration of more realistic rigid-body models. A work by Stalford and Hoffman[20] computes optimal trajectories for a rigid-body aircraft model defined by smooth analytical expressions. Bocvarov et al.[21] solve minimum-time reorientation maneuvers for a rigid-body aircraft the aerodynamic coefficients of which are fitted by analytical smooth functions. The extension of trajectory optimization from a point-mass model to a nonlinear 6-DOF model is not nearly as trivial as we might think. Not only is the complexity of the problem increased with more state variables and controls, but more importantly, simultaneous control of the motion of the center of mass, c_m, and orientations of an aircraft tend to be much more demanding than control of the c_m alone. The latter makes the optimization problem more difficult to converge. An algorithm that is less sensitive to the increased complexity is necessary for any attempt to solve such a problem. When the model is not defined analytically, an efficient, nongradient algorithm is required.

A. Model of an Advanced Fighter Aircraft

The aircraft model used in this study is based on Ref. 22 with some minor modifications detailed in Ref. 17. The aircraft is representative of a modern, high-performance, supersonic fighter. In dimension, it is approximately the same size as the F-15 Eagle. Details of the aircraft characteristics are given in Table 2.

The aircraft has a maximum speed of Mach 2.5 and an absolute ceiling of approximately 60,000 ft. Each of the two turbofan engines of the aircraft can produce a maximum thrust of about 30,000 lb. The throttle ranges from 20 to 135 deg with the afterburner engaged beyond 83 deg. The primary control surfaces are two stabilators, two ailerons, and a rudder. The control surface deflection limits and conventions are given in Table 3.

There are 24 aerodynamic coefficients that define the nonlinear aerodynamics of the aircraft. Each coefficient is defined by multidimensional tabulated data. The thrust level is also dependent on another two-dimensional table. Because of the sheer size of the data (more than 30,000 data), any smooth interpolation scheme will cause the optimization process to take an unrealistically long time and large memories. Linear interpolations are, therefore, used for table look-up. Thus, this is a nonsmooth optimization problem, and indirect trajectory optimization methods based on the calculus of variations are not applicable. The trajectory optimization problem is best transformed into a nonlinear programming problem and solved using an appropriate algorithm. Two proved sequential quadratic programming algorithms[3,23] were tried for this problem but both failed, rightfully, because of erroneous gradient information.

B. Optimal Trajectories Using Hide-and-Seek

The objective of the optimization is to find the optimal control histories for the aircraft to minimize the flight time for a specified maneuver. Thus, the performance index of minimization is

$$J = (t_f - t_0) \tag{4}$$

subject to some constraints on the trajectory

$$g_i(x(t_f)) = 0, \qquad i = 1, \ldots, l \tag{5}$$

where $x(t)$ in Eq. (5) is the state of the system, and t_0 and t_f are the starting and final time of the maneuver. The specific forms of the g_i depend on the maneuvers. The nonlinear rigid-body dynamic equations involve twelve state variables for three-dimensional flight, representing position, velocity, angles, and angular rates. The five controls are the engine throttle, the deflections of the symmetric and differential stabilators, symmetric ailerons, and rudder. They are parameterized by cubic spline functions. The nodes of the cubic splines are the design parameters. The flight time t_f (with $t_0 = 0$) is another optimization parameter. Through numerical integration of the dynamic equations of the aircraft, the original optimal control problem now becomes a nonlinear programming problem with cost function Eq. (4) and constraints Eq. (5). Hide-and-Seek was chosen to be implemented in the trajectory optimization program. Because Hide-and-Seek is a nongradient algorithm, conditioning of the problem is not a concern. A penalty function approach is therefore used for its simplicity to handle the equality constraints (5):

$$f = -t_f - \sum_{i=1}^{l} k_i|g_i| \tag{6}$$

where $k_i > 0$ are the penalty coefficients. Hide-and-Seek was used to find the global maximum of f in the design space. The choices of k_i only affect the accuracy of the ith constraint. The performance of Hide-and-Seek is not sensitive to $k_{i.}$ Several optimal maneuvers have been investigated.

1. Two-Dimensional Minimum Time-to-Half Loop

To gain some quick comparison of the performance of various algorithms, we first investigated two-dimensional motion in a vertical plane. The initial conditions are

Table 3 Control surface limits and sign conventions

Control surface	Symbol	Limit, deg	Positive deflection
Symmetric stabilator	δ_H	+15/−25	Trailing edge down
Differential stabilator	δ_D	±20	Left trailing edge down
Aileron	δ_A	±20	Left trailing edge down
Rudder	δ_R	±30	Trailing edge left

Table 4 Two-dimensional half-loop results

Method	Time-of-flight	Function evaluations
Hide-and-Seek	24.9 s	5518
Nelder-Mead	39.4 s	2084
Principal Axis	35.1 s	4063

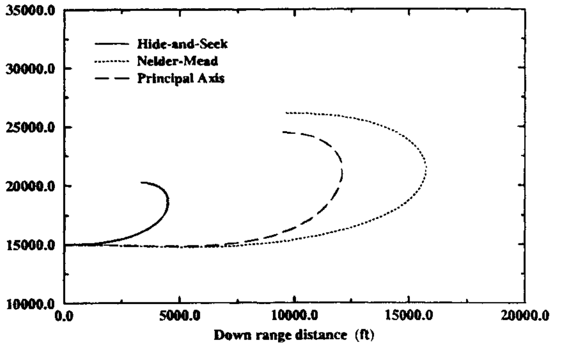

Fig. 2 Two-dimensional half-loop trajectories.

$$v_0 = \text{Mach } 0.6, \qquad \alpha_0 = 2.3 \text{ deg},$$
$$\theta_0 = 2.3 \text{ deg}, \qquad h_0 = 15{,}000 \text{ ft} \tag{7}$$

where v stands for velocity, α for angle of attack, θ for pitch angle, and h for altitude. All other state variables are initially zero. The terminal conditions are on the flight path angle γ and pitch rate q

$$\gamma_f = 180 \text{ deg}, \qquad q_f = 0 \tag{8}$$

The other state variables are free at the final time. The two controls in this case are symmetric stabilator deflection and engine throttle. First, each control is parametrized by cubic splines of 6 nodes, totaling 13 design parameters, including the parameter for the flight time. To compare the performance of Hide-and-Seek, the Principal Axis method[4] and the Nelder-Mead Simplex method[5] were also used to solve the problem. The results are listed in Table 4.

The number of function evaluations for Hide-and-Seek is an average of five solutions because the initial search point is randomly generated. All five solutions yield the same flight time. The results listed for Principal Axis and Nelder-Mead Simplex methods are the best among the solutions obtained from 20 different starting points. Both methods at best yield local optimal solutions for this problem that require 41 and 58% longer flight time, respectively. Figure 2 shows the three trajectories. The histories of angle of attack, symmetric stabilator deflection, and throttle setting of the solution found by Hide-and-Seek are plotted in Figs. 3–5.

Next, to investigate the effects of the dimensionality of the problem on Hide-and-Seek, the two controls are parametrized with more nodes. When each control is parametrized with 10 and then 15 nodes, the total design parameters are 23, and then 33. The problem is resolved using Hide-and-Seek. Table 5 compares the numbers of function evaluations in three cases (again, they are averages of five repeated solutions for each case). The flight times for three cases are very close (with a difference of 0.1 s). We observe that the number of function evaluations is increased by only 81% when the dimensionality is increased by 154%. This not-so-linear behavior is attributable to the fact that $\hat{f}$ in Eq. (3) was used in the cooling schedule instead of f^*, which is not known a priori. The Nelder-Mead Simplex and Principal Axis methods were also tried for the 33-parameter case. For a very small bounded search region containing the optimum point, the algorithms used over 100,000

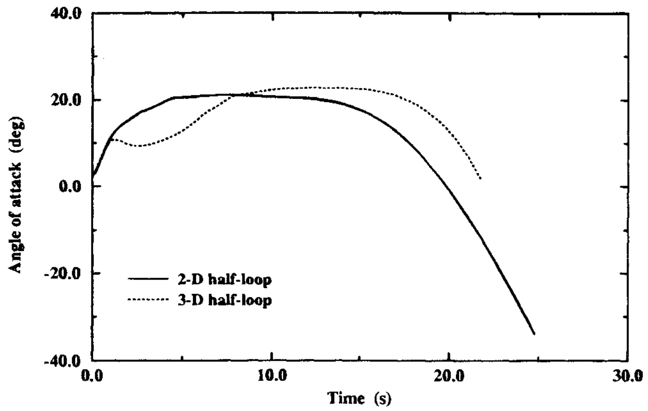

Fig. 3 Angle-of-attack histories for half-loop maneuvers.

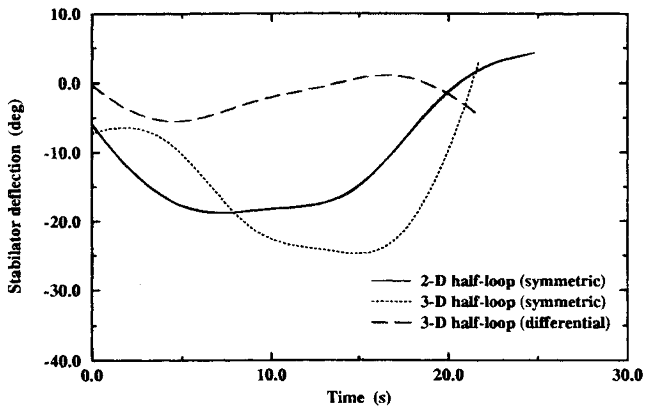

Fig. 4 Stabilator histories for half-loop maneuvers.

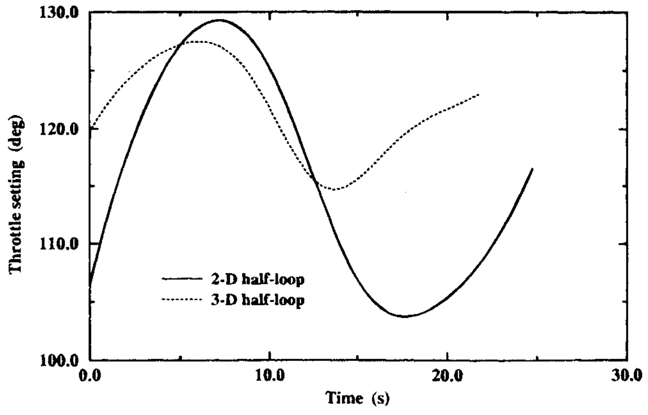

Fig. 5 Throttle histories for half-loop.

Table 5 Number of function evaluations for two-dimensional half-loop

Number of variables	Hide-and-Seek
13	5518
23	5713
33	10015

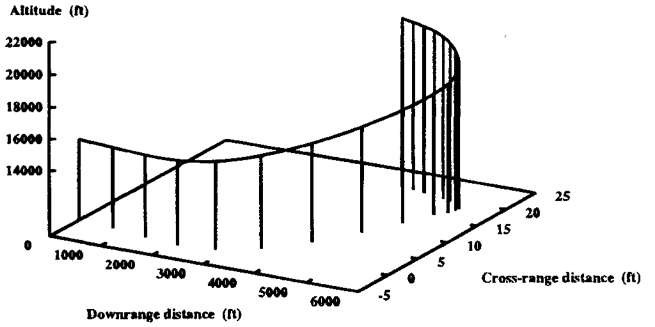

Fig. 6 Three-dimensional half-loop trajectory.

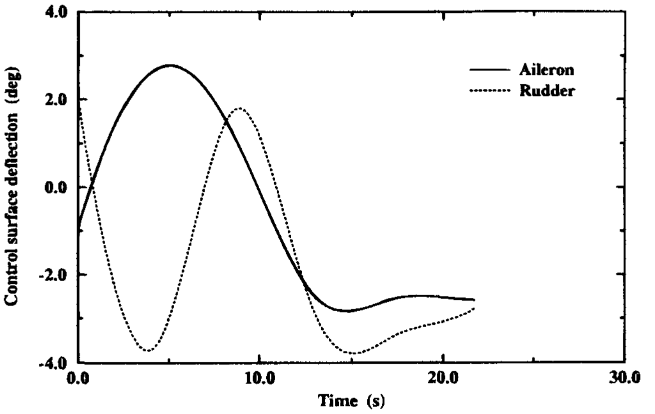

Fig. 7 Aileron and rudder histories for three-dimensional half-loop.

function evaluations and showed no sign of convergence, and the process had to be terminated.

2. Three-Dimensional Minimum Time-to-Half Loop

A three-dimensional problem is not restricted to planar motion. The initial conditions are the same as Eq. (7) with all other variables initially equal to zero. Because of different ranges of definition for various angles, the terminal conditions are now

$$\gamma_f = 0 \text{ deg}, \quad \psi_f = 180 \text{ deg}, \quad \phi_f = 180 \text{ deg}, \quad q_f = 0 \quad (9)$$

where ψ and ϕ are the heading and roll angle, respectively. In this case, all five controls are used. Each control is parametrized by cubic splines 0f five nodes. There is a total of 31 design parameters. Hide-and-Seek was again successful in finding the optimal solution. The time-of-flight is 21.7 s with an average of 29,246 function evaluations. The trajectory is shown in Fig. 6. We note that the trajectory is almost within the vertical plane (the maximum crossrange is only 21.5 ft). This is because out-of-plane motion requires expenditure of additional energy, and hence it tends to extend the flight time. However, with the additional control surfaces, the time-of-flight is reduced by 14.7% as compared to the two-dimensional case (in which t_f = 24.9 s). The histories of angle of attack, symmetric and differential stabilator deflections, and throttle setting are plotted in Figs. 3–5. The variations of aileron and rudder deflections are depicted in Fig. 7. Note that, against intuition, the throttle settings in both the two- and three-dimensional cases are not at the maximum value (135 deg). This suggests a singular control for the thrust. The explanation is that a full-throttle setting increases the velocity too rapidly, which in turn causes the time required to reorient the velocity vector to increase.

3. Three-Dimensional Minimum Time-to-Turn

The differences between this maneuver and the three-dimensional half-loop maneuver are in the terminal constraints

$$\gamma_f = 0 \text{ deg}, \quad \psi_f = 180 \text{ deg}, \quad \phi_f = 0 \text{ deg}, \quad h_f = h_0 \quad (10)$$

The initial conditions and design parameters are the same as those employed in the three-dimensional half-loop maneuver. Hide-and-Seek obtained an optimal flight time of 30.6 s with an average of 51,258 function evaluations. A three-dimensional view of the optimal trajectory is shown in Fig. 8. Unlike the half-loop maneuver, this flight path contains considerable lateral motion. Figure 9 displays the histories of various angles. Figure 10 contains the variations of the four control surfaces. It is seen

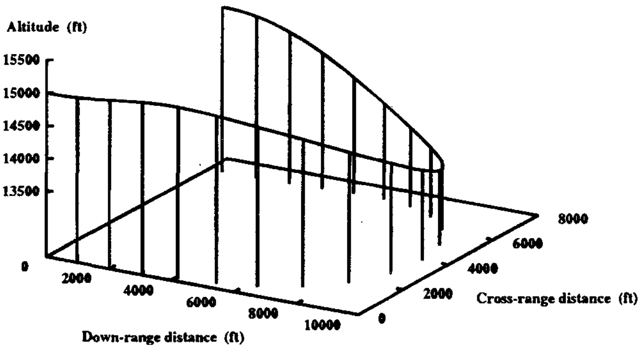

Fig. 8 Three-dimensional turn trajectory.

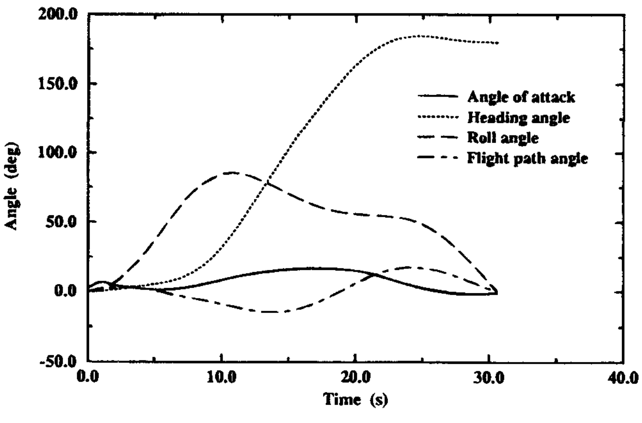

Fig. 9 Angle histories for three-dimensional turn.

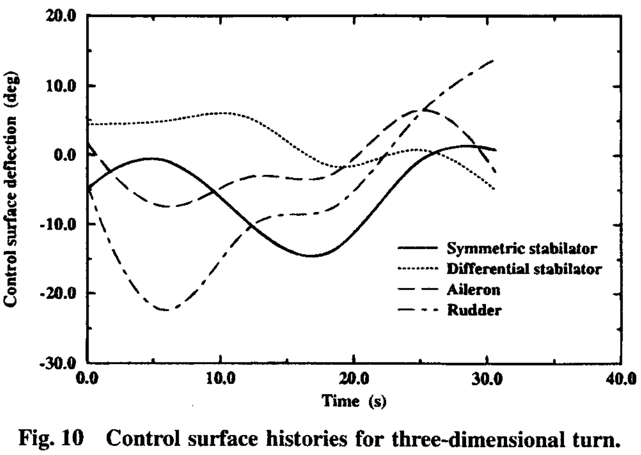

Fig. 10 Control surface histories for three-dimensional turn.

that large rudder deflections are used for the turn. Again, the throttle setting (not shown) is not saturated. For the three-dimensional maneuvers considered in this paper, the sideslip angle is always found to be very small (< 0.2 deg).

The Nelder-Mead Simplex and Principal Axis algorithms were also used for three-dimensional maneuvers with many starting points. Both failed to give even local optima because of the size of the problems.

4. Minimum Time-to-Climb

The minimum-time climb is a classic aircraft performance problem. The flight trajectory is restricted to a vertical plane. Unlike the half-loop and turn maneuvers, it has been found in the past[19] and confirmed by our numerical experiments that for the time-optimal climb, the throttle operates at maximum allowable value. Therefore, we, set the throttle at a constant value of 127 deg. The initial conditions are the takeoff conditions from the ground

$$v_0 = 400 \text{ ft/s}, \qquad \theta_0 = 5 \text{ deg}, \qquad \alpha_0 = 5 \text{ deg} \tag{11}$$

All of the other initial conditions are zero. The terminal conditions are

$$h_f = 50{,}000 \text{ ft} \tag{12}$$

$$v_f = \text{Mach } 1.0 \tag{13}$$

$$\gamma_f = 0 \text{ deg} \tag{14}$$

Condition (12) is always required. The corresponding trajectories are called one-, two-, and three-constraint solutions, depending upon whether or not conditions (13) and (14) are enforced. To prevent the trajectory from intersecting the ground after takeoff, another terminal constraint

$$|\min\{0.0, h_{\min}\}| = 0 \tag{15}$$

is always imposed, where $h_{\min}$ is the minimum altitude after takeoff. The only control for this problem is the symmetric stabilator deflection. It is first parametrized by cubic spline over 10 equally spaced intervals, giving a 12-parameter problem. Then, to allow greater maneuverability in the initial phase of the climb, the first interval is divided further into 4 subintervals, yielding a 15-parameter problem.

The Hide-and-Seek, Principal Axis, and Nelder-Mead Simplex algorithms were tested on this problem. Table 6 summarizes the results for one- to three-constraint solutions. The solutions listed by Principal Axis and Nelder-Mead Simplex are the best ones from 10 different starting points. Again, Hide-and-Seek consistently obtained solutions better than those found by Principal Axis and Nelder-Mead Simplex. Improvements ranged from 19.4 to 44.8%. Note from Table 6, that a finer parametrization provides negligible reductions in flight time. But aircraft ground clearance after takeoff is improved. Figure 11 shows the three climb trajectories found by Hide-and-Seek subject to increasingly stringent constraints. The histories of various angles and stabilator deflection for the 3-constraint solution are depicted in Fig. 12. Bryson and Denham[19] found that for their aircraft model, the minimum-time-to-climb trajectory required the aircraft to dive in order to break the sound barrier. Our two- and three-constraint trajectories for which the final velocity is constrained also exhibit an obvious diving segment. But in both cases, this counterintuitive diving occurs approximately around Mach 1.2 where the drag coefficient of our aircraft model reaches its maximum value. In the dive, the aircraft quickly accelerates through this high-drag region at the price of a slight loss of altitude. However, the overall time-of-flight is shortened by the increased velocity.

IV. Conclusions

The use of realistic model for a dynamic system often renders trajectory optimization for such a system to be a nonsmooth problem. A continuous, simulated annealing algorithm, Hide-and-Seek, is introduced as a powerful tool for nonsmooth trajectory optimization problems. This algorithm has been rigorously proved to converge to the global optimum. It is robust, relatively efficient, and very easy to implement and apply. The trajectory optimization problem for a realistic, nonanalytic model of an advanced fighter is solved successfully using Hide-and-Seek,

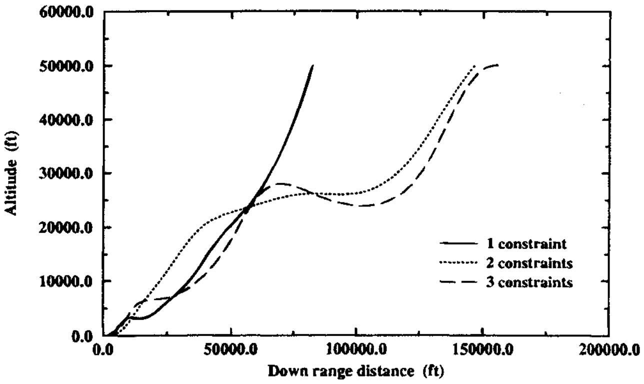

Fig. 11 Minimum-time climb trajectories.

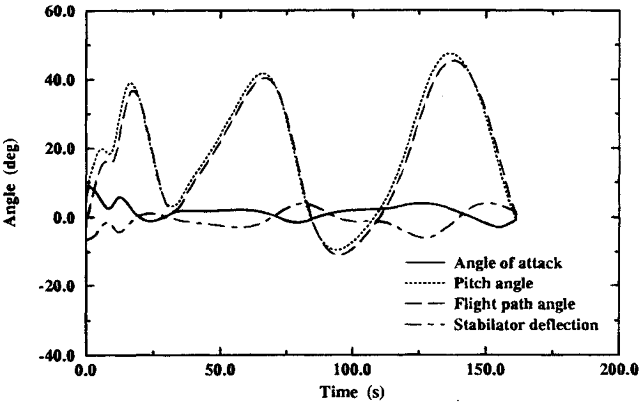

Fig. 12 Angles and control surface histories for 3-constraint climb.

Table 6 Minimum time-to-climb trajectories : flight times

Constraints	Nelder-Mead, 12 variables	Principal Axis, 12 variables	Hide-and-Seek, 12 variables	Hide-and-Seek, 15 variables
1	171.3 s	140.9 s	118.3 s	117.3 s
2	186.3 s	169.8 s	156.0 s	153.2 s
3	199.3 s	172.8 s	161.2 s	160.4 s

whereas some other well-known algorithms either failed or yielded only local optima. Several common two- and three-dimensional time-optimal maneuvers were investigated. The study revealed some interesting features of the time-optimal trajectories. They include variable instead of full-throttle settings for half-loop and turn maneuvers, almost planar motion for three-dimensional half-loop maneuver, and a dive phase in minimum-time-to-climb trajectories. With these applications, this paper demonstrated the great potential of simulated annealing algorithms in trajectory optimization for highly complicated systems.

Acknowledgments

This work was supported in part by NASA Langley Research Center under grant NAG-1-1255. The authors thank Lin-Lin Chen for bringing the Hide-and-Seek algorithm to the authors' attention. The authors are grateful to Robert L. Smith for letting them use the Hide-and-Seek program.

References

[1] Brauer, G. L., Cornick, D. E., and Stevenson, R., "Capabilities and Applications of the Program to Optimize Simulated Trajectories (POST)," NASA CR-2770, Feb. 1977.

[2] Hargraves, C. R., and Paris, S. W., "Direct Trajectory Optimization Using Nonlinear Programming and Collocation," *Journal of Guidance, Control, and Dynamics,* Vol. 10, No. 4, 1987, pp. 338–342.

[3] Gill, P. E., Murray, W., Sanders, M. A., and Wright, M. H., "User's Guide For NPSOL (Version 4.0): A Fortran Package For Nonlinear Programming," System Optimization Laboratory, Department of Operation Research, Stanford University, Stanford, California, 1986.

[4] Brent, R. P., *Algorithms for Minimization without Derivatives,* Prentice-Hall, 1973, pp. 128–133.

[5] Parkinson, J. M., and Hutchinson, D., *Numerical Methods in Nonlinear Optimization,* Academic Press, 1972, pp. 99–114.

[6] Goldberg, D. E., *Genetic Algorithm in Search, Optimization, and Machine Learning,* Addison Wesley, Reading, MA, 1989.

[7] Krishnakumar, K., and Goldberg, D. E., "Control System Optimization Using Genetic Algorithms," *Journal of Guidance, Control, and Dynamics,* Vol. 15, No. 3, 1992, pp. 735–740.

[8] Katragadda, L. K., and Pierson, B. L., "Low-Thrust Space Trajectory Optimization Using Genetic Algorithms" (in preparation).

[9] Otten, R. H. J. M., and van Ginneken, L. P. P. P., *The Annealing Algorithm,* Kluwer Academic Publishers, The Netherlands, 1989, pp. 1–19.

[10] Wong, D. F., Leong, H. W., and Liu, C. L., *Simulated Annealing for VLSI Design,* Kluwer Academic Publishers, The Netherlands, 1988, pp. 1–7.

[11] Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E., "Equations of State Calculations by Fast Computing Machines," *Journal of Chemical Physics,* Vol 21, 1953, pp. 1087–1092.

[12] Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P., "Optimization by Simulated Annealing," *Science,* Vol. 220, 1983, pp. 671–680.

[13] Mahfoud, S. W., and Goldberg, D. E., "Parallel Recombinative Simulated Annealing: A Genetic Algorithm," Genetic Algorithm Laboratory, University of Illinois, IlliGAl Report No. 92002, Urbana, IL.

[14] Petersen, C., "Parallel Distributed Approaches to Combinatorial Optimization," *Neural Computation,* Vol. 2, No. 3, 1990, pp. 261–269.

[15] Hajek, B., "Cooling Schedules for Optimal Annealing," *Mathematics of Operations Research,* Vol. 13, 1988, pp. 311–329.

[16] Bélisle, C. J. P., Romeijn, H. E., and Smith, R. L., "Hide-and-Seek: A Simulated Annealing Algorithm for Global Optimization," Department of Industrial and Operations Engineering, TR 90-25, University of Michigan, Ann Arbor, 1990.

[17] Khan, M. A., "New Techniques in Trajectory Optimization and Guidance," M. S. Thesis, Iowa State University, Ames, IA, 1992.

[18] Aluffi-Pentini, F., Parisi, V., and Zirili, F., "Global Optimization and Stochastic Differential Equations," *Journal of Optimization Theory and Applications,* Vol. 47, 1985, pp. 1–16.

[19] Bryson, A. E., and Denham, W. F., "A Steepest-Ascent Method for Solving Optimum Programming Problems," *Journal of Applied Mechanics,* Vol. 29, 1967, pp. 247–257.

[20] Stalford, H., and Hoffman, E., "Maximum Principle Solutions for Time-Optimal Half-Loop Maneuvers of a High Alpha Fighter Aircraft," *Proceedings of American Control Conference,* Pittsburgh, PA, June 1989.

[21] Bocvarov, S., Lutze, F. H., and Cliff, E. M., "Time-Optimal Reorientation Maneuvers for a Combat Aircraft," *Journal of Guidance, Control, and Dynamics,* Vol. 16, No. 2, 1993, pp. 232–240.

[22] Brumbaugh, R. W., "An Aircraft Model for the AIAA Controls Design Challenge," *Proceedings of the AIAA Guidance, Navigation, and Control Conference* New Orleans, LA, AIAA, Washington, DC, 1991 (AIAA Paper 91-263).

[23] Pouliot, M. R., "CONOPT2: A Rapidly Convergent Constrained Trajectory Optimization Program for TRAJEX," General Dynamics, Convair Division, Rept. GDC-SP-82008, San Diego, CA, 1982.