Research Article

Steepest-Ascent Revisited: Unconstrained Missile Trajectory

Elsayed M. Khalil, Hao Zhou, and Wanchun Chen

School of Astronautics, Beihang University (BUAA), Beijing 100083, China

Correspondence should be addressed to Elsayed M. Khalil; mykingboody@gmail.com

Received 11 June 2014; Accepted 31 August 2014; Published 15 September 2014

Academic Editor: Joseph Majdalani

Copyright © 2014 Elsayed M. Khalil et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A Steepest-Ascent numerical procedure for offline trajectory optimization of a surface-to-surface missile attacking a stationary target is presented. A detailed numerical solution, starting from building the mathematical formulation till generating an offline angle of attack control history, is illustrated. A novel approach for guessing a nominal control program is conducted. The formulation entails nonlinear 2-DOF missile flight dynamics with mixed boundary conditions. The technique of changing of variables is adopted to convert the free-final-time optimization problem to a fixed-final-time one. The influences of the optimization algorithm controlling parameters are investigated. A new form of the weighting matrix is proposed. A novel technique of relaxation factors for eliminating the terminal states violation is described. Finally, a comparison between the obtained control history and the one obtained by a nonlinear optimal control package “GPOPS” is presented. The results indicate that the use of Steepest-Ascent method, to a great extent, is comparable in accuracy and iteration time to the pseudospectral optimization package “GPOPS.” This paper, to a great extent, is a detailed procedure for the method of Steepest-Ascent analyzed and verified by the authors from many undetailed sources to disclosure of the main problems faced by user of Steepest-Ascent.

1. Introduction

One of the major problems in missile mission analysis is the construction of a feasible trajectory which allows a set of mission objectives to be met, subjected to operational and physical constraints not to be violated [1]. In the 1950s, there was a considerable progress in techniques for the determination of optimal flight paths which are optimal from the point of view of a performance criterion. Leitmann presented an excellent bibliography in his survey paper [2].

Optimization process can be broken down into two basic sequential steps: mathematical formulation and generation of a numerical solution. There are three mathematical formulations that are usually referred to in optimization theory, each of which seems to be suitable for some types of optimization problems: calculus of variations, Pontryagin maximum principle, and dynamic programming. The second step in optimization process, numerical solution, can be generated by using one of the three fundamental numerical procedures: neighboring extremal, Steepest-Ascent, and quasilinearization. Unfortunately, the formulation is a minor

part of the analysis. The major portion of the problem is considered to be the numerical solution [3].

The method of gradients or “method of Steepest-Ascent,” as it is sometimes called, is a basic concept for the solution of extremum problems. To find a local minimum of a function using *gradient descent*, one takes steps into the gradient function negative direction at the current point. If steps are taken into the positive of the gradient, one may approach a local maximum of that function; the iteration sequence is then known as *gradient ascent*. The history of Steepest-Ascent goes back to Cauchy in 1847. He was the first one to propose the use of gradients as one of the alternatives in solving nonlinear equations or a system of equations. The method of Steepest-Descent/or Steepest-Ascent, as one of the first-order gradient methods, has been applied successfully to the problem of optimizing the trajectory for a prescribed vehicle since the 1960s [4]. This success is thanks to the approach developed by Kelley and Bryson et al. [5, 6], to overcome the two-point boundary-value problems associated with the calculus of variations. Kelley [5] applied the gradient method to variational problems of flight paths optimization. A nearly

similar technique has been developed and adopted by Bryson and Denham [7].

Key objectives for any optimization technique are simplicity, ease of implementation, acceptable accuracy, and solution time. Steepest-Ascent methods, being of first-order, are, from a computational point of view, relatively simple to implement even for complex problems. Initially, when far from the optimal solution, Steepest-Ascent methods work well. The disadvantage of this method is its tendency to exhibit poor convergence rates as the optimum is reached. Several techniques for speeding up convergence were advised [8–10].

2. Problem Statement

A problem of interest is in the area of flight mechanics. It concerns the design of a trajectory along which the missile mission is accomplished in some sort of best fashion. The selection of a “best” trajectory is the determination of the optimum time history of the controllable forces [11]. The case study under investigation is a typical fin controlled missile flying in its power off phase of flight (i.e., for the coast-fall phase). Control over the missile trajectory will be achieved by controlling the aerodynamic forces acting on during flight which leads us to controlling of the time history of the angle of attack. The objective of optimization is to hit target with a specified terminal impact angle while maximizing the impact velocity.

The motivation for this paper, specifically choosing Steepest-Ascent, is its simplicity from the point of view of implementation. Besides, there is no enough detailed published work found in the open literature about this method and this encourages us to tackle this technique. The full detailed mathematical work by Kelley, Bryson, and Denham dated back to the 1960s. The problem of selection of algorithm controlling parameters in some basic functions in the formulation was faced during generating the optimization code. Gottlieb [12] suggested some clues to circumvent some of these parameters. This paper will deal basically with terminally constrained optimization problems with a Mayer type objective function.

The main contribution of this paper is to develop a trajectory optimization tool based on Steepest-Ascent method with superior features when compared to other existing optimization packages regarding the accuracy and solution speed. The work presented here deals with some problems faced by the designers when dealing with indirect methods: guessing a control and choosing a weighting function. Furthermore, a new technique of “relaxation factors” to deal with the violation in terminal constraints in a simple stable, but yet efficient, manner is suggested.

3. Trajectory Optimization Problem Definition

In order to discuss the trajectory optimization problem, a general formulation and definitions of the problem must

first be given. For any dynamical system with the following assumed dynamics

$$\dot{\mathbf{X}} = \mathbf{f}(\mathbf{x}(\mathbf{t}), \mathbf{u}(\mathbf{t}), \mathbf{t}), \quad (1)$$

let “ $\mathbf{X}$ ” denote a $(n \times 1)$ state vector, where “ n ” is the minimum number of the states or the degrees of freedom of the system being optimized which can completely specify the system behavior for all future time in the absence of disturbance,

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_n \end{bmatrix}. \quad (2)$$

Let “ $\mathbf{U}$ ” denote a $(m \times 1)$ control vector, where “ m ” is the number of controls or decision variables, which can be changed arbitrarily to influence the future state of the system

$$\mathbf{U} = \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \vdots \\ \mathbf{u}_m \end{bmatrix}. \quad (3)$$

The first requirement of the optimization problem is that some quantity exists which can be used to compare the performance of one solution to another. This quantity is often called optimization criteria, pay-off function, objective function, cost function, or performance index. It can take the form of a scalar quantity or a final cost (*Mayer Problem*); integrand quantity or a running cost (*Lagrange Problem*); or a coupled form (*Bolza Problem*).

The trajectory optimization process solves for finding a trajectory of (1) in $[t_o, t_f]$, states and controls, that extremize any desired cost function of the Bolza form:

$$\mathbf{J} = \phi_f(\mathbf{x}_f, \mathbf{t}_f) + \int_{t_o}^{t_f} \mathbf{L}(\mathbf{x}, \mathbf{u}, \mathbf{t}) \cdot d\mathbf{t}. \quad (4)$$

Within the cost metric, the endpoint cost ϕ can be any function of the terminal states and time, while the integrated cost L can be any function of the states, controls, and time along the entire trajectory.

4. Necessary Conditions for Optimality

Bryson and Ho [13] and Lewis and Syrmos [14] derived the necessary conditions using the calculus of variations. Pontryagin et al. [15] showed that finding an extremal solution is equivalent to the requirement that the variational Hamiltonian be extremized (maximized or minimized), with respect to all admissible states and controls. For completeness of the illustration, the necessary conditions will be presented here without the need for reproducing the derivation.

Without loss of generality, we will assume that there is one control ($m = 1$). Define the Hamiltonian “ H ” and the auxiliary functions

$$\mathbf{H}(\mathbf{x}, \mathbf{u}, \lambda, \mu, \mathbf{t}) = \mathbf{L}(\mathbf{x}, \mathbf{u}, \mathbf{t}) + \lambda^T \cdot \mathbf{f}(\mathbf{x}, \mathbf{u}), \quad (5)$$

$$\Phi(\mathbf{x}_f, \mathbf{u}_f, \mathbf{t}_f, \mathbf{v}_f) = \phi_f(\mathbf{x}_f, \mathbf{u}_f, \mathbf{t}_f) + \mathbf{v}_f^T \cdot \psi_f(\mathbf{x}_f, \mathbf{u}_f, \mathbf{t}_f),$$

where $\lambda : [t_o, t_f] \rightarrow \mathbb{R}^n$ and $\nu_f \in \mathbb{R}^{N_f}$ are Lagrange multipliers associated with states and terminal conditions. To construct a numerical solution to an optimization problem, a solution is selected which satisfies only some of the following conditions:

Differential State Equations

$$\dot{\mathbf{X}} = \mathbf{f}(\mathbf{x}(\mathbf{t}), \mathbf{u}(\mathbf{t}), \mathbf{t}). \quad (6)$$

States Boundary Values

$$\psi_o = \psi_o^j(X_o, U_o, t_o) \quad \psi_f = \psi_f^i(X_f, U_f, t_f). \quad (7)$$

Differential Adjoint Equations

$$\dot{\lambda}^T = -H_t \quad \lambda^T(x_f) = \phi_x|_{t=t_f}. \quad (8)$$

Transversality Condition

$$(\dot{\Phi}_t + \mathbf{H})|_{t_f} = \mathbf{0}. \quad (9)$$

Weierstrass Condition

$$\mathbf{H}(X, x, \lambda, u^{\text{opt}}) \geq \mathbf{H}(X, x, \lambda, \bar{u}). \quad (10)$$

This solution is then iteratively corrected in a direction which tends to bring in the other conditions. What distinguish the Steepest-Ascent iteration process are the conditions which are satisfied at the start as opposed to those which are brought in. The iterative process of Steepest-Ascent is characterized by satisfaction of conditions (6), (7), and (8) in each successive iteration without the need to satisfy condition (10). Condition (9) holds only if the final time is not specified explicitly.

5. Steepest-Ascent Formulation

Power off trajectory optimization of a surface-to-surface missile attacking a stationary target involves first-order non-linear differential equations with full specification of initial conditions and partial specification of final conditions on the state variables. The optimal control problem is to determine, from all possible programs for the control variable, the one program that extremizes a terminal state while satisfying the required state-variable initial and final conditions. This is a two-point boundary-value problem (TPBVP) that, unfortunately, cannot be solved in a closed form. A numerical solution is required. Many methods are available to solve this type of problems and are widely reported in the literature. A Steepest-Ascent method, presented in detail by Bryson and Denham [7], was chosen to determine the optimal controls due to its simplicity and it is conveniently easy to be coded. The procedure begins with guessing a control program. The equations of motion are integrated forward from the initial conditions using the guessed control. In general, the resulting state-variable time histories may not satisfy the final conditions. Small perturbations of the control variables about the nominal trajectory are considered to drive the terminal quantities to their prespecified values while extremizing a pay-off function. For this problem, the pay-off function

is the final impact velocity. By continuing this iteration process, along the ascent direction in the control-variable hyper-space, a control-variable history, which maximizes final velocity while satisfying both initial and final boundary conditions, is obtained.

It should be noted that neither the Steepest-Ascent method nor any other method, for numerically solving this type of optimal control problems, is assured to find global optimal solutions. Only local optimal solutions may be found. Since all of the techniques considered have this same drawback, it was not a factor in the choice of which method to use.

6. Physical and Dynamical Models

6.1. Modeling Assumptions

- (1) A point mass dynamical model, with trimmed angle of attack aerodynamics, is used.
- (2) The flight is limited to a vertical plane over a nonrotating flat earth with constant gravitational field.
- (3) Zero lag control system means that there is neither error nor delay during flight.
- (4) There are no in-flight disturbances, and consequently wind is neglected.

6.2. *Dynamic Model.* Consider the following:

$$\dot{\mathbf{F}}_t = \begin{pmatrix} \dot{\mathbf{X}} \\ \dot{\mathbf{Y}} \\ \dot{\dot{\gamma}} \\ \dot{\mathbf{V}} \end{pmatrix} = \begin{pmatrix} V \cdot \cos \gamma \\ V \cdot \sin \gamma \\ \frac{L}{m \cdot V} - \frac{g \cdot \cos \gamma}{V} \\ -\frac{D}{m} - g \cdot \sin \gamma \end{pmatrix}, \quad (11)$$

where g is the gravitational acceleration in (m/sec²); D is the aerodynamic drag force in N; L is the aerodynamic lift force in N; m is the missile mass in kg; V is the missile velocity in m/s in wind-frame; γ is the flight path angle in rad; and α is the missile angle of attack in rad.

The full definition of the vehicle states is represented by the four-element vector $(X, Y, \gamma, V)^T$ composed of down range, altitude, flight path angle, and velocity, respectively.

6.3. *Mission Boundary Conditions.* The problem under investigation has split boundary conditions; some boundary values are specified at initial time and the rest are specified at the terminal time. The missile boundary conditions during its power off phase are listed in Table 1, where the missile empty mass is $m = 1500$ kg with a reference cross-sectional area $S_{\text{ref}} = 0.23$ m².

There are four differential state constraint equations. Therefore, the problem has $(2n + 2)$ boundary values with a total of 10 boundary values, n initial conditions, and n final conditions, in addition to the initial and final time. The proposed mission is to attack a stationary target at a specific down range with a preselected impact angle. The case

FIGURE 1: Nominal control guess for different missions.

TABLE 1: Boundary conditions.

State		Fixed initial	Required terminal
Down range	(km)	10.0	From 45.0 to 130.0
Altitude	(km)	11.0	00.0
Flight path angle	(deg)	45.0	-80.0
Time	(sec)	00.0	Free
Velocity	(m/sec)	1000	Free, pay-off

under investigation is with fixed initial conditions ($n + 1$) and mixed terminal conditions; three terminal boundaries are fixed (terminal down range, terminal altitude, and terminal flight path angle) and two are free (terminal flight time and terminal impact velocity). Eight fixed boundary values exist, thus leaving two degrees of freedom for control variable optimization. This control variable is the one upon which some optimum requirements can be imposed.

6.4. Atmospheric Model. The standard US 1976 atmospheric model [16] is used. A sixth-order 1D polynomial, covering 50 km of altitude, of density (ρ) and speed of sound (a), is used. The used polynomial goodness has a root mean square error (RMSE) of 0.0001 for density and 0.001 for speed of sound:

$$\begin{aligned}\rho &= f_\rho(Y) = [Y_{1 \times 7}^{(6)}] \cdot [P_{7 \times 1}^\rho], \\ a &= f_a(Y) = [Y_{1 \times 7}^{(6)}] \cdot [P_{7 \times 1}^a],\end{aligned}\quad (12)$$

where $[P_{7 \times 1}]$ is the polynomial coefficient matrix.

6.5. Missile Aerodynamic Model. The drag coefficient (C_D) and lift coefficient (C_L), calculated by using Missile DATCOM, were arranged in lookup tables. A fifth-order 2D polynomial, with respect to Mach number (M) and angle of

attack (α), is generated using MATLAB. The used polynomial goodness has a RMSE of 0.045 for C_D and 0.015 for C_L :

$$\begin{aligned}C_D &= f(M, \alpha) = [M_{1 \times 6}^{(5)}] \cdot [P_{6 \times 6}^D] \cdot [\alpha_{6 \times 1}^{(5)}], \\ C_L &= f(M, \alpha) = [M_{1 \times 6}^{(5)}] \cdot [P_{6 \times 6}^L] \cdot [\alpha_{6 \times 1}^{(5)}],\end{aligned}\quad (13)$$

where $[P_{6 \times 6}]$ is the 2D polynomial coefficient matrix.

7. Optimization Procedure

7.1. Guessing a Nominal Control. The optimization process using Steepest-Ascent, as an indirect method, begins with a guess for a nominal control variable. This is one of the drawbacks of all indirect methods. Failure to generate a sufficiently accurate initial guess can prevent convergence to a solution, even if a solution exists. Furthermore, convergence rate of the optimization process depends on how this guess is close to the optimal one.

There is nothing in the literature talking about how to guess control for missile missions. This issue is totally left for the designer. It is not easy to give a good guess for many processes since its behavior may be not fully predicted. The nominal control used in this report depends mainly on the free flight trajectory parameters (free flight range and max altitude). Nominal controls guess for two different cases (one is greater and the other is lesser than free flight range) is illustrated in Figure 1.

In case of ranges greater than the free flight one, case (a) is used, where $X1$ represents the required range; $X2$ represents the down range at which the free flight trajectory attains the max altitude; $X3$ represents the downrange at which the nominal control begins to raise up, and it equals half of $X2$. In case of ranges lesser than the free flight one, case (b) is used, where $X1$ represents the required range. The guessed control in the two cases is piecewise continuous. The idea behind the construction of the given guess is that, in case of targets located at down range lesser than the free flight trajectory, the missile is forced to pitch down from the initial boundary to succeed in reaching the target. On the other hand, in case

of target located at down ranges greater than the free one, the missile needs to pitch up to generate lift to enlarge its reachable down ranges. Moreover, high pitching up before the summit point will decrease the reachable ranges and this is the reason for beginning the pitch-up program from the midpoint between the boundary point (burn-out point) and the summit point.

7.2. Changing of Variables. In most books, it is free-final-time problem that is being tackled first to derive the necessary conditions for optimal control. Fixed-final-time problems were treated as an equivalent variation with one more state for time. However, for numerical methods, fixed-final-time problems are the general form and free-final-time problem is solved by converting it into a fixed-final-time problem [17, 18]. The reason is that it is unavoidable to do numerical integration (either by indirect or by direct methods). Therefore, a time interval must be specified for these methods. For a certain class of free-final-time problems, the necessity of solving a sequence of fixed-final-time problems can be avoided by making an appropriate change of variables. To achieve this, a state variable needs to be identified or function of several state variables, whose initial and terminal values are known and whose time rate of change does not change sign (or become zero) along the optimal trajectory. By making this quantity the independent variable, the problem can be transformed from a “free-final-time” to a “fixed-final-time” problem in which time becomes a “state” variable, and one need not know its optimal terminal value. The requirement for the function relating the new independent variable and time to be strictly monotonic is imposed in order that the state differential equations and the cost function can be integrated with respect to the new independent variable [19].

Since “ X ” (down range) has fixed initial and terminal value and moreover “ X ” behaves monotonically, then its adoption as an independent variable provides a simple mean of avoiding complications of free end time optimization problems [20].

By eliminating the first equation and using the chain rule, the dynamical system can be converted to a new one with “ X ” as independent variable. By dividing all the equations by the first one and rearranging to place the terminal fixed constrained states first and then the free end terminal states, the dynamical system will take the following form:

$$\dot{\mathbf{F}}_{\mathbf{x}} = \begin{pmatrix} \dot{Y}_{\mathbf{x}} \\ \dot{\gamma}_{\mathbf{x}} \\ \dot{T}_{\mathbf{x}} \\ \dot{V}_{\mathbf{x}} \end{pmatrix} = \begin{pmatrix} \tan \gamma \\ \frac{L}{m \cdot V^2} \cdot \sec \gamma - \frac{g}{V^2} \\ \frac{1}{V} \cdot \sec \gamma \\ -\frac{D}{m \cdot V} \cdot \sec \gamma - \frac{g}{V} \cdot \tan \gamma \end{pmatrix}. \quad (14)$$

Now, the new four-element vector ($n \times 1$) is $(Y, \gamma, T, V)^T$, in which down range “ X ” is replaced by time “ T ”.

7.3. Scaling. For a poorly scaled optimization problem, that is, when the corresponding objective function has highly

elongated level sets, the Steepest-Ascent direction may not provide much reduction in the function [21]. This leads to forcing the algorithm to choose a very small step size to save stability and linearity. This is due to the fact that the gradient vector may be nearly orthogonal to the direction that leads to the extremum, producing zigzagging iterates that converge very slowly. Moreover, in such situations, even divergence may occur if one does not proceed carefully with the control of the step size [22]. Nevertheless, it is important to note that, if the objective function is poorly scaled, then Steepest-Ascent is a stiff differential system, which requires the integration step to remain small despite slow changes in the state variable; otherwise, numerical instabilities may produce drastic increases in the approximation error. This drawback holds for any ascent method that is highly sensitive to poor scaling [23]. For such functions, preconditioning, which changes the geometry of the space to circumvent the slow convergence, is required. Many authors like to entitle this preconditioning process with balancing, normalization, or squaring of the box [24, 25].

Ross and Gong [25] advise that the scales should be selected so that the scaled states and costates are roughly of the same order of magnitude. Many trials to scale the problem using scaling factors were extracted from the free flight simulation. The surprising conclusion is that doing the optimization without scaling should be the first choice. Steepest-Ascent is capable of circumventing the problem of scaling due to the gradual elimination of errors. Two trials were solved, with and without scaling. There is noticeable decrease in optimization time till convergence in case of scaled problem. There is no substantial gain in final velocity after scaling; contrarily, there is a slight reduction. GPOPS failed to converge without using the automatic scaling offered by the algorithm.

To simplify the development, three assumptions regarding the form of the optimization problem will be made.

- (1) The admissible control set U is the entire control space with no constraints.
- (2) The terminal range X_f is specified and, therefore, the transversality condition can be eliminated.
- (3) There is no need to specify a terminal state value as a stopping condition, since it is an optimization problem with specified end independent variable.

7.4. Iteration Procedure and Comments

7.4.1. Compute a Nominal Trajectory. By a forward integration of the dynamical model (14) using a guessed nominal (α) program with the specified initial conditions from Table 1 and store the states column wise in a matrix.

Calculate the terminal constraints violation, and then choose values of $\delta\psi$ to cause the next nominal solution to be closer to the desired values, where

$$\psi[x_i(x_f)] = x_i(x_f) - x_{if} \quad i = 1, \dots, 2, \quad (15)$$

where q is the number of terminally constrained states. Steepest-Ascent adopts the idea of gradual elimination of

FIGURE 2: Relaxation factor function with iteration number.

errors. It is not advised to eliminate the error in a single iteration. Simply, this will not work due to many reasons, first of which is the system nonlinearity, which means that there will be a difference between the predicted change and the actual change in the terminal values; secondly, this will not help with algorithm stability for subsequent iterations. Moreover, different ε , associated with every terminally constrained state, can be chosen depending on the nonlinearity inherent in the system in this direction. So, $\delta\psi$ can be chosen as

$$\delta\psi = \varepsilon_i \cdot \psi [x_i(x_f)] \quad 0.0 < \varepsilon \leq 1.0. \quad (16)$$

In the work presented here, ε can be called “relaxation factor” associated with q terminally constrained states. The main idea behind relaxation factor is to relax or weaken the correction of the terminal state violation and can be decreased gradually to reach a value lower than unity till the end of optimization. Figure 2 presents the suggested ε function with iteration number.

7.4.2. Computation of Influence Functions [13]. By a backward integration of adjoint differential equations with its terminal values in (17), store and remember that it is now in a reverse order:

$$\lambda_x = - \int_{x_f}^{x_0} \left(\frac{\partial \dot{F}_x}{\partial x} \right)^T \cdot \lambda_x dx, \quad (17)$$

$$\lambda_i(x_f) = \begin{cases} 0, & i = 1, \dots, 2, \\ \left(\frac{\partial \phi}{\partial x_i} \right)_{x=x_f}, & i = 3, \dots, 4, \end{cases}$$

$$\mu_x = - \int_{x_f}^{x_0} \left(\frac{\partial \dot{F}_x}{\partial x} \right)^T \cdot \mu_x dx,$$

$$\mu_{ij}(x_f) = \begin{cases} 1, & i = j, \quad i = 1, \dots, 4, \\ 0, & i \neq j, \quad j = 1, \dots, 2, \end{cases}$$

where

$$\left(\frac{\partial \dot{F}_x}{\partial x} \right)^T = \begin{pmatrix} \frac{\partial f_1}{\partial Y} & \frac{\partial f_2}{\partial Y} & \frac{\partial f_3}{\partial Y} & \frac{\partial f_4}{\partial Y} \\ \frac{\partial f_1}{\partial \gamma} & \frac{\partial f_2}{\partial \gamma} & \frac{\partial f_3}{\partial \gamma} & \frac{\partial f_4}{\partial \gamma} \\ \frac{\partial f_1}{\partial T} & \frac{\partial f_2}{\partial T} & \frac{\partial f_3}{\partial T} & \frac{\partial f_4}{\partial T} \\ \frac{\partial f_1}{\partial V} & \frac{\partial f_2}{\partial V} & \frac{\partial f_3}{\partial V} & \frac{\partial f_4}{\partial V} \end{pmatrix}, \quad (18)$$

such that

$$\left(\frac{\partial \dot{F}_x}{\partial x} \right)^T = \begin{pmatrix} 0.0 & \frac{S_{\text{ref}}}{2m} \cdot \sec \gamma \cdot \left(C_L \cdot \frac{\partial \rho}{\partial Y} + \rho \cdot \frac{\partial C_L}{\partial Y} \right) & 0.0 & -\frac{S_{\text{ref}}}{2m} \cdot V \cdot \sec \gamma \cdot \left(C_D \cdot \frac{\partial \rho}{\partial Y} + \rho \cdot \frac{\partial C_D}{\partial Y} \right) \\ \sec^2 \gamma & \frac{S_{\text{ref}}}{2m} \cdot \rho \cdot C_L \cdot \sec \gamma \cdot \tan \gamma & \frac{1}{V} \cdot \sec \gamma \cdot \tan \gamma & -\frac{S_{\text{ref}}}{2m} \cdot \rho \cdot V \cdot C_D \cdot \sec \gamma \cdot \tan \gamma - \frac{g}{V} \cdot \sec^2 \gamma \\ 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & \frac{S_{\text{ref}}}{2m} \cdot \rho \cdot \sec \gamma \cdot \frac{\partial C_L}{\partial V} + \frac{2g}{V^3} & \frac{-1}{V^2} \cdot \sec \gamma & -\frac{S_{\text{ref}}}{2m} \cdot \rho \cdot \sec \gamma \cdot \left(C_D + V \cdot \frac{\partial C_D}{\partial V} \right) + \frac{g}{V^2} \tan \gamma \end{pmatrix}. \quad (19)$$

For fast and accurate differentiation, analytical derivatives are more efficient than performing functional evaluations and differencing. This can be referred to as the difference between algebraic Jacobians and analytic Jacobians. A Jacobians function is generated to evaluate the gradients of the Hamiltonian with respect to state variables.

7.5. Calculation of Weighting Matrix “W”. Primary concern in selecting weighting matrices is the effect that they will have on the objective function. The good selection of “W” may enable the algorithm to allow for larger step sizes in the areas of low sensitivities and reduced step sizes in high sensitivity areas. Sensitivity here accounts for nonlinearity which is

FIGURE 3: H_{uu} with down range X .

represented by the Hamiltonian second-order derivative with respect to control variable (α). There are many ideas for selecting the weighting matrix “ W ”. A new “ W ” depending on the absolute value of the Hessian of the Hamiltonian is proposed:

$$W(x) = \text{absolute}(H_{uu}), \quad (20)$$

where

$$H_{uu} = \left(\frac{\partial^2 \dot{F}_x}{\partial \alpha^2} \right)^T \cdot \lambda_x, \quad (21)$$

such that

$$\frac{\partial^2 \dot{F}_x}{\partial \alpha^2} = \begin{pmatrix} \frac{\partial^2 f_1}{\partial \alpha^2} \\ \frac{\partial^2 f_2}{\partial \alpha^2} \\ \frac{\partial^2 f_3}{\partial \alpha^2} \\ \frac{\partial^2 f_4}{\partial \alpha^2} \end{pmatrix} = \begin{pmatrix} 0.0 \\ \frac{S_{\text{ref}}}{2m} \cdot \rho \cdot \sec \gamma \cdot \frac{\partial^2 C_L}{\partial \alpha^2} \\ 0.0 \\ -\frac{S_{\text{ref}}}{2m} \cdot \rho \cdot \sec \gamma \cdot V \cdot \frac{\partial^2 C_D}{\partial \alpha^2} \end{pmatrix}^T. \quad (22)$$

The absolute value of H_{uu} is used because it is a maximization problem, so H_{uu} has a negative sign and “ W ” is a positive definite matrix. A limitation on the minimum value should be made in order not to have zero elements in “ W ”. The proposed “ W ” was checked with many trajectories and fast stable convergence was noticed in all trials. Figure 3 shows an illustration for the preparation of H_{uu} to use it in the weighting function W .

It is shown in Figure 3 that H_{uu} can take positive and negative values along the trajectory (Figure 3(a)). Since it will

be used in a weighting function, it must take positive definite values (Figure 3(b)). Moreover, it must not take zero in order not to have infinite values in the integration process and the minimum value should be limited to a value greater than zero (Figure 3(c)).

7.5.1. Compute Impulse Response. By a forward integration using zero initial values, take the final values

$$I_{\psi\psi} = \int_{x_0}^{x_f} \mu_x^T \cdot \left(\frac{\partial \dot{F}_x}{\partial \alpha} \right) \cdot W^{-1} \cdot \left(\frac{\partial \dot{F}_x}{\partial \alpha} \right)^T \cdot \mu_x dx \quad (2 \times 2),$$

$$I_{J\psi} = \int_{x_0}^{x_f} \lambda_x^T \cdot \left(\frac{\partial \dot{F}_x}{\partial \alpha} \right) \cdot W^{-1} \cdot \left(\frac{\partial \dot{F}_x}{\partial \alpha} \right)^T \cdot \mu_x dx \quad (1 \times 2),$$

$$I_{JJ} = \int_{x_0}^{x_f} \lambda_x^T \cdot \left(\frac{\partial \dot{F}_x}{\partial \alpha} \right) \cdot W^{-1} \cdot \left(\frac{\partial \dot{F}_x}{\partial \alpha} \right)^T \cdot \lambda_x dx \quad (1 \times 1), \quad (23)$$

where

$$\frac{\partial \dot{F}_x}{\partial \alpha} = \begin{pmatrix} \frac{\partial f_1}{\partial \alpha} \\ \frac{\partial f_2}{\partial \alpha} \\ \frac{\partial f_3}{\partial \alpha} \\ \frac{\partial f_4}{\partial \alpha} \end{pmatrix}^T = \begin{pmatrix} 0.0 \\ \frac{S_{\text{ref}}}{2m} \cdot \rho \cdot \sec \gamma \cdot \frac{\partial C_L}{\partial \alpha} \\ 0.0 \\ -\frac{S_{\text{ref}}}{2m} \cdot \rho \cdot \sec \gamma \cdot V \cdot \frac{\partial C_D}{\partial \alpha} \end{pmatrix}^T. \quad (24)$$

7.5.2. Select a Step Size. There are two strategies in decision making situations. One may prefer to take a decision directly

FIGURE 4: Final velocity comparison with different target ranges.

FIGURE 5: Iteration time comparison with different target ranges.

by figuring out the correct course to be followed or indirectly by thinking of the courses not to be followed. Steepest-Ascent takes the former choice and chooses the steepest gradient direction towards the optimal solution, but the decision concerning the step sizes taken in this direction, to a great extent, is a matter of the designer choice regarding not violating the assumption of linearization. Simply, a value of “ P ” in (26) can be chosen instead of selecting a reasonable value of “ dP ” in (25). Basically, choice of “ dP ” is made to insure that the correction in the control program is, reasonably, small enough to insure that the linearization theory is not violated:

$$(dP)^2 = \int_{x_0}^{x_f} \delta\alpha(x)^T \cdot W(x) \cdot \delta\alpha(x) dx, \quad (25)$$

$$P = \left[\frac{(dP)^2 - \delta\psi^T \cdot I_{\psi\psi}^{-1} \cdot \delta\psi}{I_{JJ} - I_{J\psi} \cdot I_{\psi\psi}^{-1} \cdot I_{\psi J}} \right]^{1/2}, \quad (26)$$

where $0.0 \leq P \leq 1.0$. Bryson and Ho [13] were followed, in which they suggest a value of one for “ P ”. Extensive investigation for the choice of “ P ” reveals that, if “zero” is chosen, the method is forced to deviate all the controlling energy to meet the terminal constraints and to neglect the pay-off function, while choosing “one” means that the method is forced to deviate equal energies to meet the terminal

FIGURE 6: Load factor change with down range for 45 km trajectory.

FIGURE 7: Load factor change with down range for 130 km trajectory.

constraints and simultaneously maximize the pay-off function. Although the whole process seems to be a systematic numerical procedure, it strongly requires a suitable initial guess and reasonable selection of the mean square perturbation of the control variable program (dP), which is named “step size.”

7.5.3. Control Adjustment. Use (27) to update the nominal control program. The updating vector $\delta\alpha(x)$ is composed of two parts. The formal one is to extremize the pay-off function, so positive sign is used in case of maximization, and negative sign is for minimization. The later one is to reduce terminal constraints error, so it has negative sign. Consider the following:

$$\begin{aligned} \delta\alpha(x) = & \pm [W(x)]^{-1} \cdot \left(\frac{\partial f}{\partial \alpha} \right)^T \cdot [\lambda(x) - \mu(x) \cdot I_{\psi\psi}^{-1} \cdot I_{\psi J}] \\ & \cdot P - [W(x)]^{-1} \cdot \left(\frac{\partial f}{\partial \alpha} \right)^T \cdot \mu(x) \cdot I_{\psi\psi}^{-1} \cdot \delta\psi, \\ \alpha_{\text{new}}(x) = & \alpha_{\text{old}}(x) + \delta\alpha(x). \end{aligned} \quad (27)$$

After obtaining a new nominal trajectory, repeating of the previous steps till reaching an optimal trajectory is required.

FIGURE 8: Angle of attack history with down range for 45 km trajectory.

FIGURE 9: Angle of attack history with down range for 130 km trajectory.

7.6. Convergence Check and Iteration Termination. The most serious problems faced in the optimization by a Steepest-Ascent method or any other methods are failure to converge and/or false convergence. The former type of failure is easy to realize from the first few iterations, but the latter one is hard to be detected. At least five iteration cycles are needed to decide whether the algorithm is steering its direction to the ascent direction or not. The ascent direction here includes the terminal constraint violation suppression and the objective function maximization. Nearly, all nonlinear systems exhibit a coupling effect between terminal constrained states and pay-off function. This means that meeting the terminal constraint reduces the pay-off function and vice versa. Generally, there are three main conditions upon which the termination decision is taken:

$$\Psi_{\text{index}} = \psi \left[x(x_f) \right] \approx \varepsilon,$$

$$\Phi_{\text{index}} = I_{JJ} - I_{J\psi} \cdot I_{\psi\psi}^{-1} \cdot I_{\psi J} \approx 10^{-\epsilon}$$

$$\Phi_{\text{index}} \text{ is not improving relative to previous iteration.} \quad (28)$$

The first index Ψ_{index} considers meeting terminal constraints to a prespecified tolerance. Since it is not easy to get zero error,

FIGURE 10: Altitude versus down range for 45 km trajectory.

FIGURE 11: Altitude versus down range for 130 km trajectory.

due to numerical errors and system nonlinearities, ± 50 m is chosen to be an accepted error in altitude “ Y ” and $\pm 0.5^\circ$ error to be accepted for flight path angle “ γ ”. The second index Φ_{index} is about the improvement in pay-off function. This improvement will reach zero as the optimal trajectory is reached and $I_{\psi\psi}$ will be a singular matrix. So, limiting the minimum value is required in order not to allow the optimized trajectory to degenerate. This index takes different values in case of using the scaled problem or the no-scaled one. A value of 10^{-4} is accepted for scaled and 10^{-1} for no-scaled problem. The last check is when the Φ_{index} changes sign, so the last trajectory is the optimal one.

8. Optimization Results

Results are obtained by running a MATLAB m-code on a DELL laptop with a Windows platform and utilizing core i7 processor with 6 GB of RAM.

Fourth-order Runge-Kutta with variable step size is used to minimize the number of time steps required to integrate the trajectory to its final state and also for the integration of the influence functions and the step response. The fourth-order Runge-Kutta algorithm requires four evaluations of the acceleration per time step but permits a time step more than four times as large as an algorithm requiring only one acceleration evaluation per time step. A relative and total

FIGURE 12: Velocity change with down range for 45 km trajectory.

FIGURE 14: Flight path angle with down range for 45 km trajectory.

FIGURE 13: Velocity change with down range for 130 km trajectory.

FIGURE 15: Flight path angle with down range for 130 km trajectory.

accuracy of 10^{-4} is chosen in the setup of the ODE45 used with MATLAB package.

The feasibility of a trajectory can be validated by comparing the discretized trajectory from the optimization results with a propagated trajectory. Because the optimization results are only defined at discrete nodes, to generate a continuous reference trajectory, the control variables must first be interpolated at the node points; then the differential equations defining the system dynamics can be propagated from the initial conditions as an initial value problem. This provides a continuous, propagated reference trajectory for comparison with the discretized trajectory from the optimization. A feasible trajectory would have very little error between the two trajectories.

The no-scaled model was used in both Steepest-Ascent code and GPOPS package. The problem setup for GPOPS is with automatic scaling and using 50 nodes per interval as a minimum number of nodes. Another scaled model was used in a scaled Steepest-Ascent code. INTLAB package is used in conjunction with GPOPS for automatic differentiation and integration; besides, SNOPT is used as nonlinear programming solver (NLP). A value of 10^{-4} is used with GPOPS for the required mesh accuracy to be forced during solution.

Figure 4 shows a comparison between the pay-off obtained by the developed scaled and no-scaled codes and

GPOPS, while Figure 5 shows a comparison for the iteration time taken with the three codes. Figures 6 and 7 illustrate the optimized load factor history for the two extremum trajectories (minimum and maximum ranges). Figures 8 and 9 show angle of attack time history for 45 km and 130 km trajectories. Figures 10 and 11 show the trajectories of 45 km and 130 km. Figures 12 and 13 present the velocity profile for both 45 km and 130 km trajectories, respectively. Figures 14 and 15 prescribe the flight path angle profile for both 45 km and 130 km trajectories, respectively.

9. Conclusions and Discussion

In this paper, an attempt is made to tackle the problem of trajectory optimization from a new, simple, and efficient point of view. This attempt, although it is not yet a complete perspective, is found to be a simple efficient alternative in the area of trajectory design and guidance.

The main contribution of this paper is to revisit the method of Steepest-Ascent. This method, in the authors' opinion, did not take a fair amount of researcher's attention and care that it deserves. For nearly four decades now, the great portion of attention is devoted to optimization by direct methods due to its simplicity and ease of implementation

without giving a satisfactory amount of attention to circumvent the problems associated with optimization by indirect methods. Some of these problems directly depend on the designer and his innovation in dealing with algorithm key parameters which, indeed, needs a good experience and good judgment on the algorithm behavior with different missions. Some others need looking to the problem from a new point of view.

Superiority of Steepest-Ascent codes is obvious from the comparison for the iteration time taken by the codes. A comparative time study is conducted. A maximum reduction in iteration time occurs at the 100 km trajectory. For the 45 km and 130 km cases, Steepest-Ascent reduces the required solution time by 42% and 74%, respectively. The comparison for the required simulation time shows a good performance for all the optimized trajectories by scaled Steepest-Ascent except for 70 km trajectory, while the no-scaled one is a step behind GPOPS in case of 60 km and 70 km trajectories. The comparison for the objective function and the impact velocity shows that the final velocity obtained by Steepest-Ascent is lower than GPOPS by 0.38% and 0.49% for 45 km and 130 km trajectory. By analyzing the obtained spatial trajectories from both codes, it is apparent that the trajectory optimized by Steepest-Ascent reaches higher altitudes (17.7 km and 31.9 km for 45 km and 130 km trajectories) than that of GPOPS (17.5 km and 31.3 km for 45 km and 130 km trajectories). Steepest-Ascent reaches those altitudes in a larger down range (25.9 km and 54.8 km for 45 km and 130 km trajectories) than for GPOPS (24.0 km and 53.4 km for 45 km and 130 km trajectories).

The results presented in this paper show a promising potential of the used approach as a rapid trajectory optimization tool for the class of surface-to-surface tactical missiles. A good agreement is noticed between the Steepest-Ascent method and GPOPS in different cases. Nevertheless, superior performance of Steepest-Ascent may be detected in some cases.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

- [1] J. Campbell, W. E. Moore, and H. Wolf, "A general method for selection and optimization of trajectories," *Journal of AIAA*, AIAA-65-697, 1965.
- [2] G. Leitmann, "The optimization of rocket trajectories—a survey," in *Progress in Astronautical Sciences*, North-Holland, Amsterdam, The Netherlands, 1961.
- [3] J. E. McIntyre, *Guidance, Flight Mechanics, and Trajectory Optimization. Volume VII: The Pontryagin Maximum Principle*, NASA-CR-1006, NASA, Washington, DC, USA, 1968.
- [4] R. T. Stancil, "A new approach to steepest-ascent trajectory optimization," *The American Institute of Aeronautics and Astronautics*, vol. 2, no. 8, pp. 1365–1370, 1964.
- [5] H. J. Kelley, "Gradient theory of optimal flight paths," *Journal of ARS*, vol. 30, no. 10, pp. 947–954, 1960.
- [6] A. E. Bryson, W. F. Denham, F. J. Carroll, and K. Mikami, "Lift or drag programs that minimize re-entry heating," *Journal of Aerospace Science*, vol. 29, no. 4, pp. 420–430, 1962.
- [7] A. E. Bryson and W. F. Denham, "A steepest-ascent method for solving optimum programming problems," *Journal of Applied Mechanics*, vol. 29, no. 2, pp. 247–257, 1962.
- [8] R. Rosenbaum, "Convergence technique for the steepest descent method of trajectory optimization," *Journal of AIAA*, vol. 1, no. 7, pp. 1703–1705, 1963.
- [9] W. E. Williamson and W. T. Fowler, "A segmented weighting scheme for steepest-ascent Optimization," *Journal of AIAA*, vol. 6, no. 5, pp. 976–977, 1968.
- [10] S. D. Kenton, "An efficient algorithm for optimum trajectory computation," in *Proceedings of the Fall Joint Computer Conference (AFIPS '70)*, pp. 17–19, 1970.
- [11] N. X. Vinh, *Optimal Trajectories in Atmospheric Flight*, Elsevier, New York, NY, USA, 1981.
- [12] R. G. Gottlieb, "Rapid convergence to optimum solutions using a min-H strategy," *Journal of AIAA*, vol. 5, no. 2, pp. 322–329, 1967.
- [13] A. E. Bryson and Y. C. Ho, *Applied Optimal Control*, Taylor & Francis, New York, NY, USA, 1975.
- [14] F. L. Lewis and V. L. Syrmos, *Optimal Control*, John Wiley & Sons, New York, NY, USA, 1995.
- [15] L. S. Pontryagin, V. G. Boltyanskii, R. V. Gamkrelidze, and E. F. Mishchenko, *The Mathematical Theory of Optimal Processes*, John Wiley & Sons, London, UK, 1962.
- [16] "U.S. 1976 Standard Atmosphere," NASA-TM-X-74335, Washington, DC, USA, 1976.
- [17] X. Wang, "Solving optimal control problems with MATLAB: indirect methods," 2009, http://profs.basu.ac.ir/ganjehfar/free_space/optimal%20control%20problems.pdf.
- [18] V. H. Quintana and E. J. Davison, "A numerical method for solving optimal control problems with unspecified terminal time," *International Journal of Control*, vol. 17, no. 1, pp. 97–115, 1973.
- [19] R. J. Norbutas, *Optimal Intercept Guidance for Multiple Target Sets*, MIT-ESL-R-333, Massachusetts Institute of Technology (MIT), Cambridge, Mass, USA, 1968.
- [20] H. J. Kelley, *Method of Gradients*, Elsevier Scientific, New York, NY, USA, 1962.
- [21] L. D. Peterson, "Trajectory optimization by method of steepest descent," AFFDL TR-67-108, McDonnell Douglas Corporation, 1967.
- [22] F. Alvarez and A. Cabot, "Steepest descent with curvature dynamical system," *Journal of Optimization Theory and Applications*, vol. 120, no. 2, pp. 247–273, 2004.
- [23] W. L. Miranker, *Numerical Methods for Stiff Equations and Singular Perturbation Problems*, Reidel, Dordrecht, The Netherlands, 5th edition, 1981.
- [24] H. B. Oza and R. Padhi, "Impact-angle-constrained suboptimal model predictive static programming guidance of air-to-ground missiles," *Journal of Guidance, Control, and Dynamics*, vol. 35, no. 1, pp. 153–164, 2012.
- [25] I. M. Ross and Q. Gong, "Guess-free trajectory optimization," in *Proceedings of the AIAA/AAS Astrodynamics Specialist Conference*, Honolulu, Hawaii, USA, August 2008.