Journal of Guidance, Control, and Dynamics, Vol. 40, No. 8 (2017), pp. 2093-2101
DOI:10.2514/1.G002183

Improved gradient-based algorithm for solving aeroassisted vehicle trajectory optimization problems

Runqi Chai[a], Al Savvaris[b] and Antonios Tsourdos[c]
Cranfield University, Bedfordshire, MK43 0AL, United Kingdom

Senchun Chai[d] and Yuanqing Xia[e]
Beijing Institute of Technology, Beijing, 100081, China

I. Introduction

The Space Maneuver Vehicles (SMV) [1, 2] will play an increasingly important role in the future exploration of space, since their on-orbit maneuverability can greatly increase the operational flexibility and are more difficult as a target to be tracked and intercepted. Therefore, a well-designed trajectory, particularly in skip entry phase, is a key for stable flight and for improved guidance control of the vehicle [3, 4]. Trajectory design for space vehicles can be treated as an optimal control problem. Due to the high nonlinear characteristics and strict path constraints of the problem, direct methods are usually applied to calculate the optimal trajectories, such as direct multiple shooting method [5], direct collocation method [5, 6], or hp-adaptive pseudospectral method [7, 8].

Nevertheless, all the direct methods aim to transcribe the continuous-time optimal control problems to a Nonlinear Programming Problem (NLP). The resulting NLP can be solved numerically by well-developed algorithms such as Sequential Quadratic Programming (SQP) and Interior Point method (IP) [9, 10]. SQP methods are used successfully for the solution of large scale NLPs. Each Newton iteration of the SQP requires the solution of a quadratic programming subproblem

[a] Ph.D. Student, School of Aerospace Transport and Manufacturing, r.chai@cranfield.ac.uk, Student Member AIAA.
[b] Reader, School of Aerospace Transport and Manufacturing, a.savvaris@cranfield.ac.uk.
[c] Professor, School of Aerospace Transport and Manufacturing, a.tsourdos@cranfield.ac.uk.
[d] Associate Professor, School of Automation, chaisc97@163.com.
[e] Professor, School of Automation, xia_yuanqing@bit.edu.cn.

Copyright © 2017 by the American Institute of Aeronautics and Astronautics, Inc. All rights reserved. All requests for copying and permission to reprint should be submitted to CCC at www.copyright.com; employ the ISSN 0731-5090 (print) or 1533-3884 (online) to initiate your request. See also AIAA Rights and Permissions www.aiaa.org/randp.

containing Jacobian and Hessian matrix. However, since the QP subproblem needs to be solved exactly, this could increase the computational burden of the solver. An alternative method is the IP method developed over the last decade. Detailed description of the IP method can be found in [9]. It transcribes the inequality constraints to equality constraints by introducing some slack variables so that the problem can be solved in a simpler form. The two step IPSQP approach proposed in this paper combines the advantages of SQP and IP. A special feature of the proposed method is that the user can control the iteration of inner loop so that the QP subproblem does not need to be exactly solved. By using the iterate solution calculated from the inner loop, it becomes more accurate to identify the active set, which will have positive influences in generating the Lagrangian multipliers and next iteration points. Moreover, compared with standard SQP, it tends to be more stable and can reduce the computational time (please refer to Table 3 in the simulation results section of this paper).

In recent years, derivative-free methods have become more popular in the application of optimal control problems [11]. The main difference between traditional gradient methods and derivative-free methods is that derivative-free methods such as Genetic Algorithm (GA) [12], Artificial Bee Colony (ABC) and Particle Swam Optimization (PSO) [11, 13], do not need the calculation of gradient information. Given the global convergence ability, the global optimization approaches are applied to generate the optimal trajectories. However, due to the properties of this type of algorithms, it may have poor behaviours for solving the SMV trajectory problem. This will be further discussed in the simulation section.

II. IPSQP optimization formulation

The general NLPs can be summarised as minimizing a performance index (cost function) subject to the equality constraints and inequality constraints,

$$\begin{aligned} &\min \quad f(x) \\ &h(x) = 0, g(x) \leq 0, \;\; x \in \Re^n \end{aligned} \tag{1}$$

where x is an n-dimensional parameter vector (e.g. x=$(x_1, x_2, ..., x_n)$). Similarly, the two types of constraints are $h(x)$= $(h_1(x), h_2(x), ..., h_l(x))^T$ and $g(x)$= $(g_1(x), g_2(x), ..., g_m(x))^T$, respectively. The cost function $f(x)$ and constraints are considered as twice continuously differential on the $\Re^n$.

A. Sequential Quadratic Programming and Interior Point method

SQP [10, 14] is among the most widely used algorithms for solving general nonlinear programming problems. The basic idea for SQP algorithm is to transform the original problem to a series of QP subproblems by approximating the augmented Lagrangian quadratically and linearizing the constraints using Taylor expansion. The resulting augmented Lagrangian is:

$$L(x, \lambda, u) = f(x) + \lambda^T h(x) + u^T g(x) \tag{2}$$

Using quadratic model to approximate Eq.(2), the QP subproblem can be written as:

$$\begin{aligned} \min \tfrac{1}{2} dx^T H(x_k, \lambda_k, u_k) dx + \nabla f(x_k)^T dx \\ h(x_k) + \nabla h(x_k) dx = 0 \\ g(x_k) + \nabla g(x_k) dx \leq 0 \end{aligned} \tag{3}$$

where $dx \in \Re^n$ and a (x_k, λ_k, u_k) represents the current iterate point, whereas $H(x_k, \lambda_k, u_k)$ is the Hessian of the Eq.(3). Commonly, the Hessian is calculated using $H(x_k, \lambda_k, u_k) = \nabla_{xx} L(x_k, \lambda_k, u_k)$ or a suitable approximation defined by the user (e.g. BFGS algorithm). $\nabla h(x_k)$ and $\nabla g(x_k)$ are the Jacobian matrix of the vector of equality constraints and inequality constraints, respectively. The index k stands for the number of iteration for the optimization algorithm and $k = 0, 1, 2, ...$

If the active set is defined as Λ, then by using Newton method to calculate the Karush-Kuhn-Tucker (KKT) condition of Eq.(3), a sequence of linear equations are constructed as KKT system as given in Eq.(4). The aim for defining active set is to determine which inequality constraint is active at the current iterate point so that the inequality constraints can either be removed from the optimization model or treated as equality constraint.

$$\begin{pmatrix} H_k & \nabla h(x_k)^T & \nabla \widetilde{g}(x_k)^T \\ \nabla h(x_k) & 0 & 0 \\ \nabla \widetilde{g}(x_k) & 0 & 0 \end{pmatrix} \begin{pmatrix} dx_k \\ d\lambda_k \\ d\widetilde{u}_k \end{pmatrix} = - \begin{pmatrix} \nabla f(x_k) + \nabla h(x_k)^T \lambda_k + \nabla \widetilde{g}(x_k)^T \widetilde{u}_k \\ h(x_k) \\ \widetilde{g}(x_k) \end{pmatrix} \tag{4}$$

In Eq.(4) above, $\widetilde{g}(x)$ denotes the subset of $g(x)$ corresponding to the active inequality constraints and $\nabla \widetilde{g}(x)$ is the corresponding Jacobian matrix for the constraints in Λ. The Lagrange multipliers corresponding to the equality and active inequality constraints are denotes by λ and $\widetilde{u}$, respectively.

Another well-known and efficient algorithm is the IP method [9, 15]. Numerous updates and modifications have been made on this approach over the past decade. The ability of the IP method

to converge to a stationary point can be guaranteed theoretically. However, prior to applying this method to the general form of the problems, Eq.(1) should be transformed by introducing the Lagrange multipliers $\lambda > 0$ such that:

$$f_\lambda(x) = f(x) + \lambda \Sigma_{i=1}^{l} h_i(x) \tag{5}$$

The IP strategy consists of reducing the inequality constraints in Eq.(1) by using slack variables $s = (s_1, s_2, ..., s_m) \in \Re^m$, where all the elements in the vector should be positive. Then, the modified problem can be summarised as:

$$\begin{gathered} \min f_\lambda(x) - \mu \Sigma_{j=1}^{m} log(s_j) \\ g(x) + s = 0 \\ x \in \Re^n, s \in \Re^m \end{gathered} \tag{6}$$

And the augmented Lagrangian for Eq.(6) is:

$$L(x, s, \lambda, u) = f_\lambda(x) - \mu \Sigma_{j=1}^{m} log(s_j) + u^T (g(x) + s) \tag{7}$$

In Eq.(6), the term μ stands for a barrier variable and the smaller it is, the closer are the solutions. To solve the modified augmented Lagrangian model given by Eq.(7), m Lagrangian multipliers $u \in \Re^m$ need to be introduced for the equality constraints (e.g. $u^T(g(x) + s)$) and by applying Newton iterate, the KKT system for the modified IP can be written as:

$$\begin{pmatrix} H_k & 0 & \nabla h(x_k)^T & \nabla g(x_k)^T \\ 0 & S_k^{-1} & 0 & I \\ \nabla h(x_k) & 0 & 0 & 0 \\ \nabla g(x_k) & I & 0 & 0 \end{pmatrix} \begin{pmatrix} dx_k \\ ds_k \\ d\lambda_k \\ du_k \end{pmatrix} = - \begin{pmatrix} \nabla f(x_k) + \nabla h(x_k)^T \lambda_k + \nabla g(x_k)^T u_k \\ u_k - \mu_k S_k^{-1} e \\ h(x_k) \\ g(x_k) + s_k \end{pmatrix} \tag{8}$$

where I stands for the identity matrix and $e = (1, 1, ..., 1)^T$. S_k is a positive diagonal matrix constructed by the vectors of s_k. Eq.(8) is written in the symmetric form and the definition of S_k can be found in [9, 10, 14].

B. IPSQP optimization algorithm

In practice, both SQP and IP have some disadvantages. Most of the SQP methods use active set to determine the active inequality constraints so that the problem can be transcribed to equality

constraints programming. However, if the initial active set is chosen in an improper way, the computational burden may be increased significantly. On the other hand, for IP strategy, the most difficult part is to define the penalty functions and penalty factors in the augmented function so that it can reflect the true quality of the optimization process during the iteration. The general idea for IPSQP approach is to combine the advantages of SQP and IP. At a fixed iteration time k, the IP strategy is used to solve the QP model given by Eq.(3).

$$\begin{aligned} &\min \tfrac{1}{2} dx^T H(x_k, u_k) dx + \nabla f(x_k)^T dx - \mu_k \Sigma_{j=1}^m log(s_j^k + d_j^s) \\ &st. \quad g(x_k) + \nabla g(x_k) dx + s_k + e^T ds = 0 \\ &\qquad h(x_k) + \nabla h(x_k) dx = 0 \\ &\qquad dx \in \Re^n, ds \in \Re^m \end{aligned} \tag{9}$$

where the primal and dual variables are dx and ds, respectively. It is worth noting that the augmented Lagrangian should have the term containing the equality constraints (e.g. $\lambda_k \Sigma_{i=1}^l (h_i(x_k) + \nabla h_i(x_k)) dx$). The last two terms $s_k + e^T ds$ in the equality constraint are considered as slack variables. The IPSQP approach divides the complete optimization process into two iterations: the inner IP iteration and the outer SQP iteration. To distinguish these two iterations, the internal iteration index is defined as l while the external iteration number is defined as k. In the outer loop, the Lagrangian multipliers and slack variables should satisfy $u_k > 0$ and $s_k > 0$ at each k whereas the optimization parameters $dx_{k,l}, du_{k,l}$ and $ds_{k,l}$ in the inner circle must satisfy $s_k + e^T ds_{k,l} > 0$ and $du_{k,l} > 0$ correspondingly. After continuing the internal loop until termination or reaching the maximum number of l_{max} given by the user, a SQP solution at the next point can be achieved.

The main advantage of this two-step IPSQP approach is that the user can control the inner loop by setting the termination conditions or l_{max} at any time. Specifically, since the H_k is fixed at the internal circle, it is not required to solve the QP subproblem exactly, which means finding the time-consuming QP solution can be avoided. Also, the two-step IPSQP method does not need to approximate the initial active set at the first iteration.

Similarly with the IP and SQP cases, the KKT system of Eq.(9) given in Eq.(10) is solved by

using Newton iteration:

$$\begin{pmatrix} H_k & 0 & \nabla h(x_k)^T & \nabla g(x_k)^T \\ 0 & Du_{k,l} & 0 & Ds_{k,l} \\ \nabla h(x_k) & 0 & 0 & 0 \\ \nabla g(x_k) & I & 0 & 0 \end{pmatrix} \Delta d = - \begin{pmatrix} H_k dx_{k,l} + \nabla f(x_k) + \nabla h(x_k)^T d\lambda_{k,l} + \nabla g(x_k)^T du_{k,l} \\ Ds_{k,l} du_{k,l} - \mu_{k,l} e \\ h(x_k) + \nabla h(x_k) dx_{k,l} \\ g(x_k) + \nabla g(x_k) dx_{k,l} + s_k + e^T ds_{k,l} \end{pmatrix} \tag{10}$$

where $\Delta d = [\Delta dx_{k,l}, \Delta ds_{k,l}, \Delta d\lambda_{k,l}, \Delta du_{k,l}]^T$. $Ds_{k,l}$ and $Du_{k,l}$ are positive diagonal matrices corresponding to the slack variables and multipliers; while λ and μ are Lagrangian multipliers and penalty factors related to equality constraints and inequality constraints, respectively.

Solving the KKT system for IPSQP, the new iteration can be calculated by:

$$\begin{aligned} dx_{k,l+1} &= dx_{k,l} + \alpha_{k,l} \Delta dx_{k,l} \\ du_{k,l+1} &= du_{k,l} + \alpha_{k,l} \Delta du_{k,l} \\ ds_{k,l+1} &= ds_{k,l} + \alpha_{k,l} \Delta ds_{k,l} \\ d\lambda_{k,l+1} &= d\lambda_{k,l} + \alpha_{k,l} \Delta d\lambda_{k,l} \end{aligned} \tag{11}$$

where the step length parameter $\alpha_{k,l} \in (0, 1]$ should be chosen to ensure that the merit function achieves sufficient decrease.

To terminate the two-step IPSQP approach, termination conditions for internal loop and external loop should be defined. Convergence can be obtained by setting the tolerance value ϵ_k for Eq.(12).

$$\begin{aligned} F_O(x_k, \lambda_k, \widetilde{u}_k) &= \begin{pmatrix} \nabla f(x_k) + \nabla h(x_k)^T \lambda_k + \nabla \widetilde{g}(x_k)^T \widetilde{u}_k \\ h(x_k) \\ \widetilde{g}(x_k) \end{pmatrix} < \epsilon_k \\ F_I(dx_{k,l}, ds_{k,l}, d\lambda_{k,l}, du_{k,l}) &= \begin{pmatrix} H_k dx_{k,l} + \nabla f(x_k) + \nabla h(x_k)^T d\lambda_{k,l} + \nabla g(x_k)^T du_{k,l} \\ Ds_{k,l} du_{k,l} - \mu_{k,l} e \\ h(x_k) + \nabla h(x_k) dx_{k,l} \\ g(x_k) + \nabla g(x_k) dx_{k,l} + s_k + e^T ds_{k,l} \end{pmatrix} < \epsilon_k \end{aligned} \tag{12}$$

where F_O is the terminate condition for the external layer whilst F_I is the criteria for the inner loop.

C. Merit function

Since the method proposed above is based on Newton iteration, the merit function should be designed to measure the progress of each iterate k. In this paper, the l_∞-merit function is taken into account and it can be written as:

$$M_{\mu,r}(x,s,\lambda,u) = f(x) + \lambda h(x) - \mu\Sigma_{i=1}^{m}\log s_i + r\max|\, g(x) + s\,| \tag{13}$$

where r is the penalty factor and μ is a barrier parameter. Based on Eq.(13), the merit function for Eq.(9) can be rewritten as:

$$\begin{aligned} &M_{\mu,r}(x,s,\lambda,u,dx,ds,d\lambda,du) \\ &= \tfrac{1}{2}dx^T H dx + \nabla f(x)^T dx - \mu_k\Sigma_{j=1}^{m} log(s_j + d_j^s) \\ &\quad +\lambda\Sigma_{i=1}^{l}(h_i(x) + \nabla h_i(x))dx + r\max|\, g(x) + \nabla g(x)dx + s + e^T ds\,| \end{aligned} \tag{14}$$

For each iterate k, it should be observed that:

$$\begin{aligned} &M_{\mu,r}(x_k,s_k,\lambda_k,u_k,dx_{k,l+1},ds_{k,l+1},d\lambda_{k,l+1},du_{k,l+1}) \\ &\leq M_{\mu,r}(x_k,s_k,\lambda_k,u_k,dx_{k,l},ds_{k,l},d\lambda_{k,l},du_{k,l}) \end{aligned} \tag{15}$$

The aim of Eq.(15) is to ensure that there should be a decrease on the merit function for each iteration. However, if a small value is chosen as the step length α, the computational burden can be increased significantly. To achieve a sufficient decrease in the merit function and a reasonable step length, the Goldstein conditions are applied in this paper [9]. It can be stated as two inequalities in the following way:

$$\begin{aligned} &M_{\mu,r}(x_k,s_k,\lambda_k,u_k) + c_1\alpha_k\nabla M_{\mu,r}^T\Delta d_k \\ &\leq M_{\mu,r}(x_k+\alpha_k dx_k, s_k+\alpha_k ds_k, \lambda_k+\alpha_k d\lambda_k, u_k+\alpha_k du_k) \\ &\leq M_{\mu,r}(x_k,s_k,\lambda_k,u_k) + c_2\alpha_k\nabla M_{\mu,r}^T\Delta d_k \end{aligned} \tag{16}$$

with $0 < c_1 < c_2 < 1$. Δd_k denotes the directional derivative $(dx_k, ds_k, d\lambda_k, du_k)$. The second term of the inequality is the general sufficient decrease condition while the first term inequality is to control the step length.

Commonly, the first initial guess and active set provided by the user are far from the optimal solution and therefore, the Lagrangian multipliers calculated by using SQP are inaccurate. If the quadratic model is solved using SQP and active set, it usually takes several iterations to converge.

However, after applying inner loop controlled by the number of l_{max} in the IPSQP, the identification of active set and the solutions can be more accurate and nearer to the optimal points. In this way, the general convergence ability can be improved and the computational burden can be decreased at the same time. The identification of the active set techniques can be found in [16, 17].

The overall procedures of the proposed two-step IPSQP approach are given as follows:

Step 1: Define starting values $x_0, u_0, \lambda_0, s_0 > 0$ and the tolerance variables. For $k = 0, 1, 2...$

Step 2: Check stopping criteria given by Eq.(12), if not satisfied, then go to Step 3.

Step 3: Choose the start points $dx_{k,0}, ds_{k,0}, d\lambda_{k,0}$ and $du_{k,0} > 0$, set $l_{max} = 10$, do the inner loop.

Step 4: Check the internal KKT condition given by Eq.(12), if satisfied, then break to Step 5.

Step 5: Identify the active set based on the solution from inner loop and solve the model given by Eq.(4).

Step 6: Calculate $x_{k+1} = x_k + \alpha_k dx_k$ and other optimization parameters using line search, then go to Step 2.

III. Trajectory optimization for Space Maneuver Vehicle

A. Problem formulation

This section presents the mission scenario simulated in this investigation. The space vehicle re-enters the atmosphere at a predetermined altitude for observation and gathering of information of inaccessible areas. The space vehicle descends down to a minimum allowable altitude of around $50km$, once this altitude point is reached, the spacecraft fires its engine and starts the ascent phase, exiting the atmosphere and returning back to Low Earth Orbit (LEO). The vehicle model used in this study is similar to that of the space shuttle and is modeled as a point mass with aerodynamic properties [18]. The only difference is that an engine model is embedded in the dynamics such that the vehicle can have enough kinetic energy to exit the atmosphere and return back into orbit. Based on the dynamic model and mission requirements, a time-optimal optimization problem can

be formulated as follows:

$$
\begin{aligned}
\min \quad & J = t_f \\
\text{s.t.} \quad & \dot{r} = V \sin\gamma \\
& \dot{\theta} = \tfrac{V \cos\gamma \sin\psi}{r \cos\phi} \\
& \dot{\phi} = \tfrac{V \cos\gamma \cos\psi}{r} \\
& \dot{V} = \tfrac{T\cos\alpha - D}{m} - g\sin\gamma \\
& \dot{\gamma} = \tfrac{L\cos\sigma + T\sin\alpha}{mV} + (\tfrac{V^2 - gr}{rV})\cos\gamma \\
& \dot{\psi} = \tfrac{L\sin\sigma}{mV\cos\gamma} + \tfrac{V}{r}\cos\gamma\sin\psi\tan\phi \\
& \dot{m} = -\tfrac{T}{I_{sp}g} \\
& [r(0), \phi(0), \theta(0), V(0), \gamma(0), \psi(0), m(0)] \\
& = [r_0, \phi_0, \theta_0, V_0, \gamma_0, \psi_0, m_0]
\end{aligned}
\tag{17}
$$

where $x = [r, \theta, \phi, V, \gamma, \psi, m]^T$ are state variables representing: radial distance, longitude, latitude, speed, flight path angle, heading angle and mass, respectively. $u = [\alpha, \sigma, T]^T$ are control variables of angle of attack, bank angle and thrust. $x_0 = [r_0, \theta_0, \phi_0, V_0, \gamma_0, \psi_0, m_0]$ are the initial conditions for the state variables.

During the mission, each state and control variable should satisfy strict box constraints, which can be described as $x_{min} \leq x \leq x_{max}$ and $u_{min} \leq u \leq u_{max}$. In addition to taking the path constraints into account, in this investigation the heating rate Q, dynamic pressure P_d and load factor n_l are also considered. The detailed mathematical formulations of the path constraints can be found in [8].

B. Optimality of solution

The proposed method in Section.II is then applied to solve the spacecraft trajectory optimization problem described in Eq.(17). It should be noted that all the definition of parameters are the same with the definition discussed in [8]. Since the trajectory optimization problem is formulated as an optimal control problem, to demonstrate the optimality of the solution, the augmented Hamiltonian **H** is defined as:

$$
\begin{aligned}
\mathbf{H}(X, \lambda, u, \mu, t; t_0, t_f) = \; & g(X, u, t; t_0, t_f) + \lambda^T(t) f(X, u, t; t_0, t_f) \\
& - \mu^T(t) C(X, u, t; t_0, t_f)
\end{aligned}
\tag{18}
$$

where g is the Lagrange form cost function, f is the right hand side of the equations of motion, $\lambda(t) \in \Re^{10}$ is the costate corresponding to the dynamic equations and $\mu(t) \in \Re^3$ is the Lagrange multiplier associated with path constraints. Therefore, with:

$$\lambda = [\lambda_r, \lambda_\theta, \lambda_\phi, \lambda_V, \lambda_\gamma, \lambda_\psi, \lambda_m, \lambda_\alpha, \lambda_\sigma, \lambda_T] \in \Re^{10} \tag{19}$$

we have:

$$\dot{X} = \frac{d\mathbf{H}}{d\lambda} = f \tag{20}$$

The Hamiltonian minimization condition is based on the minimum principle such that the optimal control $u^* = [\alpha_c^*, \sigma_c^*, T_c^*]$ must minimize the Hamiltonian with respect to controls.

$$\frac{\partial \mathbf{H}}{\partial u} = \frac{\partial g}{\partial u} + \left(\frac{\partial f}{\partial u}\right)^T \lambda - \left(\frac{\partial C}{\partial u}\right)^T \mu = 0 \tag{21}$$

It is obvious that the first term $\frac{\partial g}{\partial u} = 0$ and therefore, Eq.(21) can be rewritten as:

$$\begin{cases} \frac{\partial \lambda_\alpha [K_\alpha(\alpha_c - \alpha)]}{\partial \alpha_c} - \mu_{\alpha_c}, & \text{(with respect to } \alpha_c) \text{ ;} \\ \frac{\partial \lambda_\sigma [K_\sigma(\sigma_c - \sigma)]}{\partial \sigma_c} - \mu_{\sigma_c}, & \text{(with respect to } \sigma_c) \text{ ;} \\ \frac{\partial \lambda_T [K_T(T_c - T)]}{\partial T_c} - \mu_{T_c}, & \text{(with respect to } T_c) \text{ .} \end{cases} \tag{22}$$

Also, the control multipliers should satisfy the conditions below:

$$\mu_{\alpha_c, \sigma_c, T_c} \begin{cases} \leq 0 \quad if, \ u_{\alpha,\sigma,T} = [\alpha_{min}, \sigma_{min}, T_{min}]; \\ = 0 \quad if, \ (\alpha_{min}, \sigma_{min}, T_{min}) < u_{\alpha,\sigma,T} < (\alpha_{max}, \sigma_{max}, T_{max}); \\ \geq 0 \quad if, \ u_{\alpha,\sigma,T} = [\alpha_{max}, \sigma_{max}, T_{max}]. \end{cases} \tag{23}$$

To calculate the final value of the Hamiltonian, the endpoint Lagrangian is formulated as:

$$\mathbf{H}(X(t_f), t_f, \upsilon) \ = -\frac{\partial \Phi}{\partial t_f} + \upsilon_r (r_f - r(t_f)) \ = -1 \tag{24}$$

where Φ is the Mayer form performance index and in the trajectory design problem $\Phi = t_f$. Eq.(24) indicates that the final value of the Hamiltonian should be zero for this problem. Moreover, the terminal transversality conditions are:

$$\lambda(t_f) = \frac{\partial \Phi}{\partial X(t_f)} - \upsilon^T \frac{\partial \phi}{\partial X(t_f)} \tag{25}$$

where ϕ stands for general boundary conditions. For state variables which have a free final condition, the corresponding final Lagrange multipliers should be zero. Otherwise, these multipliers at final

time are free. The Hamiltonian evolution equation is used to demonstrate the behaviour of the Hamiltonian with respect to time such that:

$$\frac{\partial \mathbf{H}}{\partial t} = 0 \tag{26}$$

Therefore, the Hamiltonian should be constant with regard to time. Combining Eq.(24) with Eq.(26), it is clear that the Hamiltonian should be -1 for all the time history. All the analysis made in this section are used to verify the quality of the numerical solution obtained using the proposed IPSQP method.

IV. Simulation results

A. Parameters setting

To investigate the feasibility and optimality of the proposed algorithm for solving a time-optimal aeroassisted vehicle trajectory problem, several simulation experiments were carried out. Results of comparative simulations using classical gradient optimization techniques and derivative-free optimization methods such as Deferential Evolution (DE) [19], Particle Swarm Optimization (PSO) [13], Genetic Algorithm (GA) [12], Ant Colony (AC) and Artificial Bee Colony (ABC) [20] are presented. All of the control parameters for the global methods are given in Table 1. The ini-

Table 1 Control parameters for global algorithms

GA		DE		PSO		AC		ABC	
NP	200	NP	200	NP	200	NP	200	NP	200
Iter	200	Iter	200	Iter	200	Iter	200	Iter	200
p_i	0.5	CR	0.7	w_{max}	0.8	α	0.5	Limit	10
p_{ij}	0.5	p_i	0.5	w_{min}	0.2	β	1.0		
CR	0.7	p_{ij}	0.5	c_1	2	ρ	0.8		
		F	0.7	c_2	2				

tial state conditions are set as $x_0 = [21162900ft, 0deg, 0deg, 25600ft/s, -1deg, 90deg, 6109.4slug]$. The minimum altitude point r_b and final point r_f (i.e. final point to return back into LEO) are $[r_b, r_f] = [21066900ft, 21162900ft]$. The box constraint values (lower and upper bounds for each variable) are given in Table 2. The path constraints maximum allowable values for heating, dynamic

pressure and load factor are set as follows: $Q_{max} = 200BTU$; $P_{dmax} = 13406.4583Pa$; $n_{lmax} = 2.5$.

Table 2 Initial conditions and box constraints for each variable

	r(ft)	$\theta(deg)$	$\phi(deg)$	V(ft/s)	$\gamma(deg)$	$\psi(deg)$	m(slug)	$\alpha(deg)$	$\sigma(deg)$	$T(N)$
ICs	21162900	0	0	25600	-1	90	6109.4	17.43	-75	0
Minimum	21066900	-180	-70	2000	-80	-180	1370.4	0	-90	0
Maximum	21162900	180	70	30000	80	180	6109.4	40	1	2×10^6

When applying heuristic algorithms to solve trajectory optimization problems, a major challenge is to implement a constraint handling strategy so that it can directly reflect the magnitude of the solution infeasibility. The constraint handling procedure used for methods described in Table 1 is based on the violation degree of constraints V. Take GA as an example, the violation degree for inequality constraints "$\leq$" ($g_j \leq g_j^*$, e.g. path constraints and terminal constraints) and equality constraints ($h_k = h_k^*$) can be defined as follows:

$$\mu_{g_j} = \begin{cases} 0, & g_j \leq g_j^*; \\ \frac{g_j - g_i^*}{g_j max - g_j^*}, & g_j^* \leq g_j \leq g_j^{max}; \\ 1, & g_j \geq g_j max. \end{cases} \quad \mu_{h_k} = \begin{cases} 1, & h_k \geq h_k^{max}; \\ \frac{h_k - h_k^*}{h_k^{max} - h_k^*}, & h_k^* \leq h_k \leq h_k^{max}; \\ 0, & h_k = h_k^*; \\ \frac{h_k^* - h_k}{h_k^* - h_k^{min}}, & h_k^{min} \leq h_k \leq h_k^*; \\ 1, & h_k \leq h_k^{min}. \end{cases} \tag{27}$$

where g_j is the value of jth constraint for each individual, while (g_j^*, g_j^{max}) and (h_k^{min}, h_k^{max}) stand for the tolerance region. Consequently, the total violation degree for each individual among the population V can be obtained via $V = \sum_{j=1}^{I} \mu_{g_j} + \sum_{k=1}^{E} \mu_{h_k}$, in which I and E are the number of inequality and equality constraints, respectively. In this way, priorities can be given to feasible individuals and individuals with a small value of V in the selection part. In order to avoid designing a penalty factor, the penalty function used in this paper is given by Eq.(28), below:

$$J_i = \begin{cases} J_i, & \text{if } V_i = 0; \\ J_{max} + J_{max} V_i, & \text{if } V_i \neq 0. \end{cases} \tag{28}$$

where J_{max} is the worst objective value among the current population.

B. Time history of the state and control for different methods

Based on the dynamic model, the objective function and the constraints given in Section III of this paper, the optimal trajectories generated by using the gradient optimization techniques and derivative-free methods are shown in Fig.1 to 3.

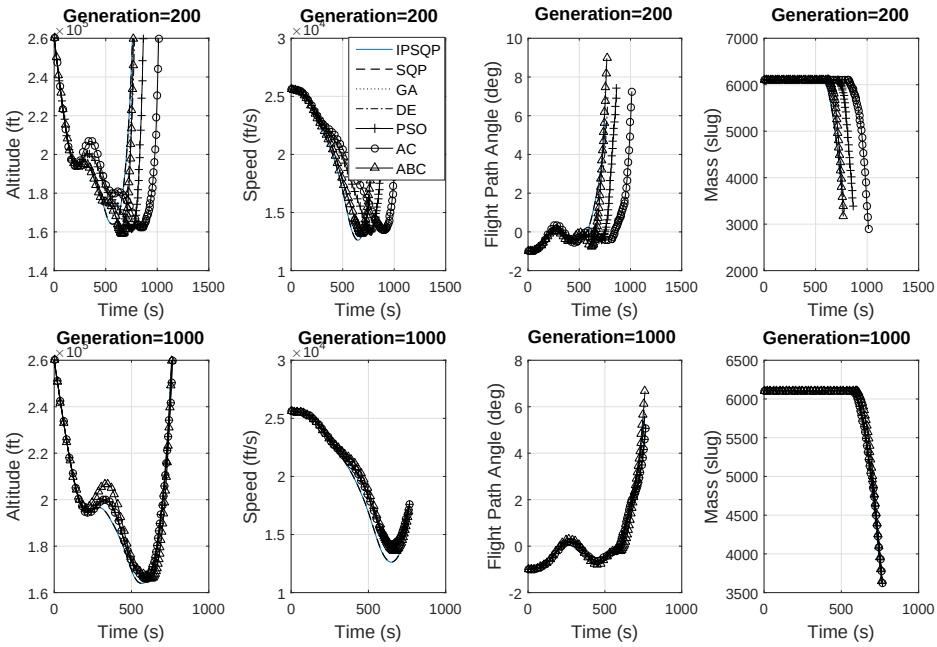

Fig. 1 State profiles obtained using gradient and stochastic methods

C. Analysis of the solutions

In this subsection, the characteristic arcs of the trajectories shown in Fig.1 to 3 are analyzed. The trajectory is split into two subintervals: descending and climbing.

1). Descending phase: In order to hit the target position (around 50km altitude) and minimize the final time, Fig.1 shows that the vehicle descends directly at the start of the mission. The angle of attack increases to slow down the vehicle so that the heating and dynamic pressure do not increase significantly, hence to avoid the path constraints from becoming active. While rapid descent is necessary, it should be noted that there is a slight dip in terms of the curvature of the altitude before arriving at the target point (50km altitude). This is because if the vehicle descends directly, then the dynamic pressure and load factor constraints may become active (see Fig.2). To decrease the path constraints values, the curvature of the altitude is decreased such that the SMV can have

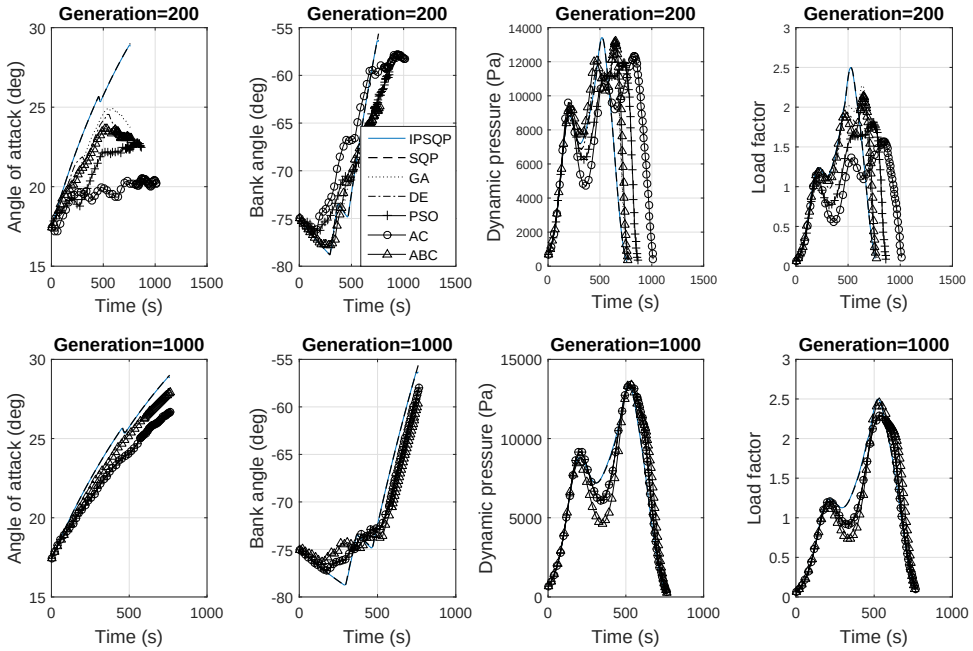

Fig. 2 Control and constraint profiles obtained using gradient and stochastic methods

sufficient time and use the drag force to slow its speed down. In this way, the structure integrity of the vehicle can be guaranteed at the expense of the objective function.

2). Climbing phase: Once the target point is reached, the vehicle fires its engine so that the vehicle can have enough kinetic energy to return back to LEO. The decreasing of air density and mass will also result in a decrease in the aerodynamic heating, dynamic pressure and load factor during the climbing phase. The trend of angle for the attack can be seen in Fig.2, where the angle of attack increases during the whole climbing phase. This is because in the climbing phase, without violating the path constraints it can have positive influences in terms of acceleration. The thrust curves calculated by IPSQP and SQP are illustrated in Fig.3. To minimize the cost function, the vehicle will use the maximum acceleration climbing back to the final boundary condition.

To analyze the performance of the proposed IPSQP algorithm, the results are compared with the "**fmincon**" subroutine in Matlab Optimization Toolbox (standard SQP). The options for initializing **fmincon** function are set as follows:

- **options**=optimset('Algorithm', 'active-set', 'Hessian', 'bfgs', 'Jacobian', 'on', 'JacobPattern', 'sparse matrix', 'TolCon', 1e-6, 'TolFun', 1e-6)

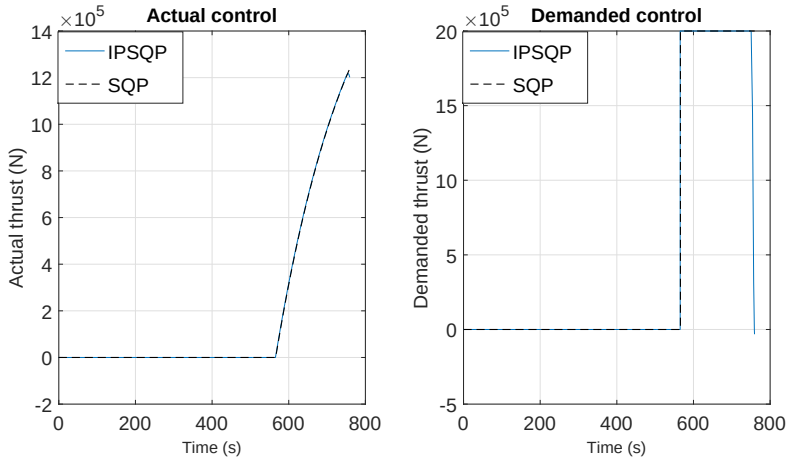

Fig. 3 Thrust generated by using IPSQP and SQP

By setting the maximum number of inner iteration for the proposed solver as 10; ten cases were tested using the IPSQP and standard SQP methods with different initial guesses, the results of which are tabulated in Table 3. Attention is given to three key performance measures that include total number of iterations, number of calls to function evaluations and execution time required to converge to the optimum solution point. The results are shown in Table 3.

Table 3 Results of IPSQP and SQP

	Newton iteration		Calls for function		Execution time(sec)	
Case number	IPSQP	SQP (fmincon)	IPSQP	SQP	IPSQP	SQP
1	15773	24378	368	486	11.47	20.51
2	18876	23749	374	488	13.49	22.17
3	14515	20916	311	444	12.01	18.49
4	11173	25898	298	492	10.44	19.46
5	16577	19663	330	391	15.07	20.37
6	424	985	12	20	2.58	4.37
7	980	980	8	12	2.75	4.16
8	503	1006	4	8	2.30	4.25
9	1025	1026	2	4	3.56	3.68
10	772	1021	4	8	2.42	3.91

As can be seen from Table 3, the proposed IPSQP algorithm is considerably faster than the

standard SQP method. This can be attributed to the properties of IPSQP algorithm that overcome the burden for choosing a starting active set. By using the solution from the inner loop, the approximation of active set can be more accurate so that the convergence ability can be improved. Moreover, compared with the standard SQP algorithm, the IPSQP converges to the optimum solution with less iteration for 8 cases, equal iterations for 2 cases. The number of calls to derivative function required by IPSQP algorithm is also smaller than its counterpart for most cases. Therefore, the IPSQP method is more time-saving and iteration-wise than the standard SQP algorithm for solving discrete optimal trajectory problems.

As for the quality of solution, there are two ways to verify optimality. One is to check if the necessary condition is satisfied. This can be achieved by comparing numerical solutions with analytical results. In Section III, the Hamiltonian value condition shows that the Hamiltonian should be -1 at the final time (i.e. $\mathbf{H}(t_f) = -1$). From the Hamiltonian evolution equation, it can be proved that the Hamiltonian is constant during the whole time history. Therefore, combining these two conditions, the Hamiltonian should keep -1 theoretically. As can be seen from Fig.4, the Hamiltonian profile keeps around -1 at $[0s, 500s]$ and $[600s, 769.06s]$, but it varies from -0.7 to -1.6 at around $[500s, 600s]$. This is because at this time period, the dynamic pressure, load factor path constraints become active as can be seen in Fig.2. That implies that the multipliers $\mu(t)$ associated with the path constraints in the Hamiltonian equation (see Eq.(18)) become non-zero. In other words, during this time domain, the numerical solution will lose some optimality and the Hamiltonian profile may contain some oscillations. In addition, the Lagrange multipliers for the dynamic equations at the final time match the theoretical results shown in terminal transversality conditions. Another way is the Bellman's principle described in [21]. The main idea of this principle is that the optimal result will not change if several points on the original optimal trajectory are selected as the initial condition to a new problem. Ten time points on the original trajectory were selected as initial conditions. The results showed that there was no better or different solution.

With regard to the derivative-free methods, it can be seen from Fig.1 and Fig.2 (the first line) that under limited computing power (small population size and number of generations), the results display a significant difference between the optimal trajectory found by gradient-based methods

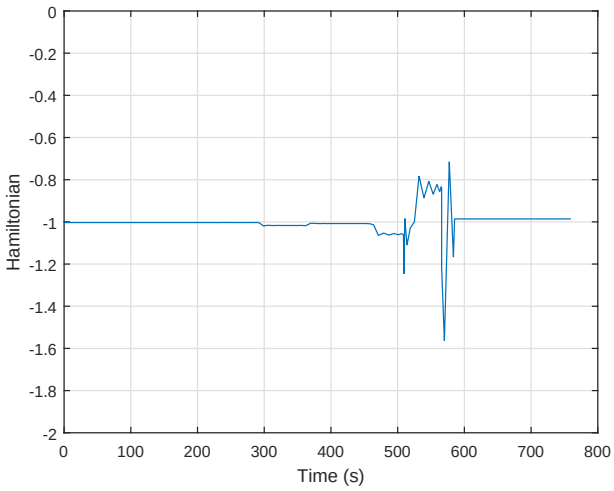

Fig. 4 Hamiltonian profile generated applying IPSQP

and derivative free algorithms. For the problem carried out in this paper, since the rate constraint of the control variable is achieved using the first order lag equations, the path constraints do not involve the control variables explicitly, which means the optimal solution may contain corners. Moreover, when the demanded control appears linearly in the differential equations, the optimal control solutions can be expected to have a "bang-bang" behaviour theoretically. That is, to minimize the Hamiltonian function **H** (e.g. Eq.(18)) with respect to the demanded control u_c, taking into account the control variable constraints, the demanded control variables should move from one point from the boundary of the feasible control region to another point on the boundary. For the solution obtained using the proposed IPSQP algorithm, this behaviour can be reflected by the actual control profiles shown in Fig.2. More precisely, the actual and demanded thrust profiles have been plotted in Fig.3 to show the "bang-bang" behaviour with respect to the demanded thrust variable. However, with heuristic algorithms, it is difficult for the control variables to converge to its bounds using limited computational efforts especially when the stochastic processes are embedded in the optimization algorithm. Therefore, although the results calculated using heuristic methods based on stochastic processes can perform a similar trend with the solution obtained using gradient-based solvers, it tends to have more oscillations in terms of the control variables, which implies that the proposed algorithm can better capture the theoretical behaviours of the control variables than the heuristic methods discussed in this paper. Further simulations were carried out by increasing the

computational effort of the heuristic approaches (e.g. number of generations and population size etc.). The solutions can be seen in Fig.1 and Fig.2 (the second line). It was found that the time history with respect to the state and control variables tends to get close to the solution generated by applying the proposed algorithm, but the oscillations with respect to the demanded control variables cannot be avoided.

It should be noted that all the global approaches were able to generate skip reentry trajectories between the predetermined initial position and terminal position without violating the path constraints, as can be seen from Table 4. Clearly, all the solutions calculated by applying derivative-free methods can be accepted as feasible solutions. In addition, when the nonlinearity of the cost functions or path constraints become higher, which means it is difficult to calculate the gradient information using gradient techniques, the global methods become the only way to solve the trajectory optimization problem. However, there are some limits for global strategies. Firstly, to combine the optimization processes with discrete methods, global techniques cannot be as flexible as gradient methods. Specifically, the IPSQP approach can be connected with either multiple shooting or direct collocation methods by discretizing the control variables or both the state and control variables. While, global approaches can be combined with multiple shooting method, however it is more challenging to combine with collocation schemes since the stochastic optimization parameters tend to increase. Following a large amount of evolutionary iterations, stochastic methods still failed to catch the behaviour of the equations of motion and satisfy all the constraints. This is because the larger the stochastic parameters become, the harder a true evolutionary direction can be found. Therefore, as suggested in [11], to combine the collocation scheme with heuristic algorithms, the number of optimization parameters should be restricted. Secondly, it is hard to verify the optimality for the solutions from global techniques whereas the proposed IPSQP and other gradient-based methods have strong theoretical verifications. For example, gradient-based methods can use the first order optimality conditions to verify the solution. Moreover, in trajectory optimization, the quality of solution largely depends on the mesh refinement process. The aim for doing mesh refinement is to determine whether the current mesh grid is proper and update the mesh grid. However, it is unrealistic for global optimization methods to check the optimality error and constraints violation

between each current mesh point and carry out the mesh refinement procedure. This will largely influence the quality and flexibility of the time histories with respect to the states and controls.

Table 4 Results of different optimization methods in detail

Iter=200	GA	DE	PSO	AC	ABC	SQP	IPSQP
J	765.04	779.67	865.27	1013.44	769.06	759.72	759.25
$Q_{max}(BTU)$	161.44	159.43	165.56	170.12	163.74	157.02	157.01
$P_{dmax}(Pa)$	12756.34	12981.36	11929.45	12344.63	13187.34	13406.47	13406.47
n_{Lmax}	2.2686	2.1229	1.8072	1.5669	2.1368	2.5019	2.5019
$e_{r_f}(ft)$	43.77	50.29	11.06	92.82	11.68	0	0
Iter=1000	GA	DE	PSO	AC	ABC	SQP	IPSQP
J	761.23	760.03	765.05	762.18	761.37	759.72	759.25
$Q_{max}(BTU)$	161.99	162.01	161.90	161.84	157.67	157.02	157.01
$P_{dmax}(Pa)$	13305.24	13363.91	13390.65	13402.02	13370.75	13406.47	13406.47
n_{Lmax}	2.2731	2.2825	2.2836	2.2901	2.4563	2.5019	2.5019
$e_{r_f}(ft)$	0.81	2.00	1.47	0.16	1.15	0	0

In terms of algorithm stability, the IPSQP algorithm can converge to the same optimum solutions for different initial settings illustrated in Table 3. Whilst there are some slight differences between the solutions generated by global strategies each time, in practice, it is desired to have a stable and efficient solver and therefore, gradient method like IPSQP and SQP can have a better performance in terms of efficiency and stability than derivative-free strategies. In addition, to further illustrate the improved performance of the proposed method over the standard SQP algorithm, an experiment was conducted by setting the IPSQP solution (time history with respect to the state and control variables and the corresponding Lagrange multipliers) as the initial guess to the standard SQP solver. Simulation results demonstrate that the SQP solver can converge to the IPSQP solution since the optimality and constraint violation tolerances can be achieved directly using the solution found by the proposed method as an initial guess, which means the method designed in this study can have an advantage over the standard SQP method. Moreover, the sensitivity of the gradient-based solver with respect to the initial guess was analyzed. The results are tabulated in

Table 3, where the last 5 experiments were carried out by setting the solution obtained using global techniques studied in this paper as the initial guess to the SQP and IPSQP algorithms.

In summary, all the results provided earlier confirm the feasibility of the proposed gradient and derivative-free algorithms. By using different optimization strategies, the vehicle can reach the target position without violating three path constraints and boundary conditions. In addition, although the solutions generated from derivative-free methods can be accepted, there is still room for improvement in terms of using these techniques in trajectory optimization. Moreover, it is shown that the IPSQP algorithm has better convergence ability and convergence speed than its counterparts, SQP method and other global techniques for trajectory optimization problems.

V. Conclusion

In this work, a two step gradient-based algorithm is applied to solve the space vehicle trajectory optimization problem. In order to effectively evaluate an active set and initial guess that can lead to accurate calculation of the Lagrange multipliers and new iteration points, an inner loop procedure is designed based on the Interior Point method. This approach allows users to define the inner iterates and using a fix Hessian, thereby ensuring a considerable reduction in the required number of iterations. This is unlike a standard SQP method, where the quadratic programming problem is required to be solved exactly. In addition, to assess the quality of Newton iterations, a merit function and descent condition are introduced. Simulation results indicate that by applying the proposed algorithm, the number of iterations, function evaluations, and computational time can be improved for most of the test cases. Comparative simulations with derivative-free methods were conducted and the results show that the proposed method can have a better performance in terms of convergence ability and stability when compared with other methods. Moreover, it was shown that the numerical solution from the proposed method can match the analytical Hamiltonian profile. Therefore, for solving space vehicle trajectory optimization problems the proposed method could provide effective and more computationally efficient solutions.

References

[1] Pienkowski, J., Whitmore, S., and Spencer, M., *Analysis of the Aerodynamic Orbital Transfer Capabilities of the X-37 Space Maneuvering Vehicle (SMV)*, American Institute of Aeronautics and Astronautics, Aerospace Sciences Meetings, 2003, doi:doi:10.2514/6.2003-908.

[2] McNabb, D. J., *Investigation of atmospheric reentry for the space maneuver vehicle*, MSc. Thesis, Air Force Institute of Technology, 2004.

[3] Brunner, C. W. and Lu, P., "Skip Entry Trajectory Planning and Guidance," *Journal of Guidance, Control, and Dynamics*, Vol. 31, No. 5, 2008, pp. 1210–1219, doi:10.2514/1.35055.

[4] Liu, X. and Lu, P., "Solving Nonconvex Optimal Control Problems by Convex Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 3, 2014, pp. 750–765, doi:10.2514/1.62110.

[5] Betts, J. T., "Survey of Numerical Methods for Trajectory Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 2, 1998, pp. 193–207, doi:10.2514/2.4231.

[6] Reddien, G. W., "Collocation at Gauss Points as a Discretization in Optimal Control," *SIAM Journal on Control and Optimization*, Vol. 17, No. 2, 1979, pp. 298–306, doi:doi:10.1137/0317023.

[7] Benson, D. A., Huntington, G. T., Thorvaldsen, T. P., and Rao, A. V., "Direct Trajectory Optimization and Costate Estimation via an Orthogonal Collocation Method," *Journal of Guidance, Control, and Dynamics*, Vol. 29, No. 6, 2006, pp. 1435–1440, doi:10.2514/1.20478.

[8] Chai, R., Savvaris, A., and Tsourdos, A., "Fuzzy physical programming for Space Manoeuvre Vehicles trajectory optimization based on hp-adaptive pseudospectral method," *Acta Astronautica*, Vol. 123, 2016, pp. 62–70.

[9] Nocedal, J. and Wright, S. J., "Numerical Optimization, Springer series in operations research," *Siam J Optimization*.

[10] Sargent, R. W. H. and Ding, M., "A New SQP Algorithm for Large-Scale Nonlinear Programming," *SIAM J. on Optimization*, Vol. 11, No. 3, 2000, pp. 716–747, doi:10.1137/s1052623496297012.

[11] Conway, B. A., "A Survey of Methods Available for the Numerical Optimization of Continuous Dynamic Systems," *Journal of Optimization Theory and Applications*, Vol. 152, No. 2, 2012, pp. 271–306, doi:10.1007/s10957-011-9918-z.

[12] Wall, B. J. and Conway, B. A., "Genetic algorithms applied to the solution of hybrid optimal control problems in astrodynamics," *Journal of Global Optimization*, Vol. 44, No. 4, 2008, pp. 493–508, doi:10.1007/s10898-008-9352-4.

[13] Pontani, M. and Conway, B. A., "Particle Swarm Optimization Applied to Space Trajectories," *Journal of Guidance, Control, and Dynamics*, Vol. 33, No. 5, 2010, pp. 1429–1441, doi:10.2514/1.48475.

[14] Heinkenschloss, M. and Ridzal, D., "A Matrix-Free Trust-Region SQP Method for Equality Constrained Optimization," *SIAM Journal on Optimization*, Vol. 24, No. 3, 2014, pp. 1507–1541, doi:doi:10.1137/130921738.

[15] Laurent-Varin, J., Bonnans, F., Berend, N., Haddou, M., and Talbot, C., "Interior-Point Approach to Trajectory Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 5, 2007, pp. 1228–1238, doi:10.2514/1.18196.

[16] Burger, M. and MÃijhlhuber, W., "Numerical Approximation of an SQP-Type Method for Parameter Identification," *SIAM Journal on Numerical Analysis*, Vol. 40, No. 5, 2002, pp. 1775–1797, doi:doi:10.1137/S0036142901389980.

[17] Sachsenberg, B. and Schittkowski, K., "A combined SQP-IPM algorithm for solving large-scale nonlinear optimization problems," *Optimization Letters*, Vol. 9, No. 7, 2015, pp. 1271–1282.

[18] Senses, B. and Rao, A. V., "Optimal Finite-Thrust Small Spacecraft Aeroassisted Orbital Transfer," *Journal of Guidance, Control, and Dynamics*, Vol. 36, No. 6, 2013, pp. 1802–1810, doi:10.2514/1.58977.

[19] Rajesh, A., *Reentry Trajectory Optimization: Evolutionary Approach*, American Institute of Aeronautics and Astronautics, Multidisciplinary Analysis Optimization Conferences, 2002.

[20] Duan, H. and Li, S., "Artificial bee colony based direct collocation for reentry trajectory optimization of hypersonic vehicle," *IEEE Transactions on Aerospace, Electronic Systems*, Vol. 51, No. 1, 2015, pp. 615–626.

[21] Kevin, B., Michael, R., and David, D., *Optimal Nonlinear Feedback Guidance for Reentry Vehicles*, American Institute of Aeronautics and Astronautics, Guidance, Navigation, and Control and Co-located Conferences, 2006.

https://dspace.org/

Improved gradient-based algorithm for solving aeroassisted vehicle trajectory optimization problems

Chai, Runqi; Savvaris, Al; Tsourdos, Antonios; Chai, Senchun; Xia, Yuanqing

2017-05-04
AIAA
http://dx.doi.org/10.2514/1.G002183
Runqi Chai, Al Savvaris, Antonios Tsourdos, Senchun Chai, and Yuanqing Xia. Improved gradient-based algorithm for solving aeroassisted vehicle trajectory optimization problems. Journal of Guidance, Control, and Dynamics, Vol. 40, No. 8 (2017), pp2093-2101

Downloaded from a DSpace repository. https://dspace.org/