Trajectory Optimization of Space Maneuver Vehicle Using a Hybrid Optimal Control Solver

Runqi Chai, *Student Member, IEEE,* Al Savvaris, *Member, IEEE,* Antonios Tsourdos, *Member, IEEE,* Senchun Chai, and Yuanqing Xia, *Senior Member, IEEE*

***Abstract*—In this paper, a constrained space maneuver vehicles trajectory optimization problem is formulated and solved using a new three-layer-hybrid optimal control solver. To decrease the sensitivity of the initial guess and enhance the stability of the algorithm, an initial guess generator based on a specific stochastic algorithm is applied. In addition, an improved gradient-based algorithm is used as the inner solver, which can offer the user more flexibility to control the optimization process. Furthermore, in order to analyze the quality of the solution, the optimality verification conditions are derived. Numerical simulations were carried out by using the proposed hybrid solver and the results indicate that the proposed strategy can have better performance in terms of convergence speed and convergence ability when compared with other typical optimal control solvers. A Monte-Carlo simulation was performed and the results show a robust performance of the proposed algorithm in dispersed conditions.**

***Index Terms*—Improved gradient-based algorithm, initial guess, optimal control, optimality verification, space maneuver vehicles (SMVs), trajectory optimization.**

I. Introduction

SPACECRAFT trajectories planning is usually recognized as an optimal control problem [1]–[4]. Due to the high nonlinear nature and strict path constraints, direct transcription algorithms are commonly used to approximate the optimal solution [5], [6]. Direct transcription algorithms [7] can be divided into two major schemes: 1) direct shooting scheme [8], in which only the control parameters are discretized at temporal nodes and 2) direct collocation scheme [9], where both the state and control are parameterized. Many both theoretical and experimental works have been carried out in this field. For example, Yakimenko [10] applied an inverse dynamics in the virtual domain collocation method to generate a near-optimal aircraft trajectory. The work of Albin *et al.* [11] presents a multiple-shooting-based model predictive control scheme for generating the optimal control sequence of a two-stage turbocharged gasoline airpath problem. In [12], considering the model physical constraints, an energy-optimal control trajectory was generated for a wave energy converter based on a specific collocation method. In addition, Benson *et al.* [7] developed a Gauss pseudospectral (orthogonal collocation) method for transcribing general optimal control problems. In their latest works, a hp-adaptive strategy was embedded in the framework of the original solver [13].

Normally, after the discretization process, the resulting nonlinear programming problem (NLP) can be solved numerically via well-developed algorithms, such as the sequential quadratic programming (SQP) [14] or interior point (IP) method [15]. However, the main challenge for gradient-based algorithms is that an initial guess value must be provided to the NLP solver. For the user to provide an initial guess is usually problematic since the guess sequences might be far from the optimal solution or close to a local optimal solution. It is likely for the optimizer to converge to a local solution or a neighborhood of the initial guess. Using optimal control software, such as GPOPS or ICLOCS, this process usually made by interpolating the boundary conditions provided by the user [16]. However, due to the lack of the physical knowledge, the initial guess value may not be physically meaningful and it can hardly satisfy the dynamic equations and path constraints. Therefore, the NLP solver may start at an infeasible point, where most of the constraints cannot be satisfied.

In recent years, evolutionary-based algorithms have become popular and have been widely investigated for solving trajectory optimization problems [6]. The main advantages of using evolutionary algorithms are that it is simple to understand and easy to implement. Furthermore, it is more likely than other methods to locate the global optimum solution. Contributions made to apply this type of methods can be found in literatures. For example, Yokoyama and Suzuki [17] implemented genetic algorithm to solve a constrained space plane reentry problem. In [18], a low-thrust interplanetary trajectory optimization problem was studied, and the main contribution is that a bilevel evolutionary-based optimal control solver was created. Kim and Lee [19] proposed a particle swarm optimization (PSO) to calculate the optimal trajectory for a manipulator motion planning problem. Similarly, a doubly fed induction generator optimal control problem was investigated in [2], wherein a modified PSO algorithm was applied to approximate the optimal control trajectory. Although all the aforementioned

Manuscript received September 11, 2017; accepted November 18, 2017. This paper was recommended by Associate Editor H. M. Schwartz. *(Corresponding author: Runqi Chai.)*

R. Chai, A. Savvaris, and A. Tsourdos are with the School of Aerospace, Transport and Manufacturing, Cranfield University, Cranfield MK43 0AL, U.K. (e-mail: r.chai@cranfield.ac.uk; a.savvaris@cranfield.ac.uk; a.tsourdos@cranfield.ac.uk).

S. Chai and Y. Xia are with the School of Automation, Beijing Institute of Technology, Beijing 100081, China (e-mail: chaisc97@163.com; xia_yuanqing@bit.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2017.2778195

works have shown the feasibility of applying stochastic algorithms for trajectory optimization problems, the verification of solution optimality will become difficult. Moreover, the computational burden caused by the evolutionary optimization process is usually high [20].

In order to combine the advantages of the traditional and stochastic-based methods, this paper presents a hybrid structure for solving the spacecraft trajectory optimization problem. So far to the best of our knowledge, there are fewer integrated design results have been reported to generate the optimal spacecraft flight trajectory. Therefore, this paper is an attempt to address this concern. The proposed structure contains an outer layer that can generate a reference trajectory with respect to the state and control variables. The algorithm used in this layer is an adaptive differential evolution algorithm and the fundamental framework of this stochastic algorithm is based on [21]. Following that, the reference trajectory is provided to the inner layer, where a new improved gradient-based solver is applied. Compared with other typical optimal control methods, using this approach can improve both the stability and efficiency. Moreover, this integrated design can offer the designers more flexibility in order to control the optimization process and verify the optimality.

The main contributions of the work reported in this paper include the following three aspects.

1) We propose a new optimal trajectory generation method by constructing an integrated framework that contains a newly developed initial guess generator and an inner gradient solver.
2) An improved gradient-based solver is designed and embedded in the proposed algorithm framework. This method can better control the solution-finding process and achieve faster convergence speed for solving the spacecraft trajectory optimization problem.
3) Both optimality verification and simulation results are provided to illustrated the effectiveness of the proposed design.

Some preliminary results have appeared in [21]; however, the results obtained in this paper are more integrated, including detailed formulation, analysis, and simulations.

The remainder of this paper is organized as follows. Section II presents the mathematical formulation of the space maneuver vehicles (SMVs) trajectory optimization problem. In Section III, the three-layer-hybrid optimal control solver is constructed. The optimality conditions of the SMV trajectory planning problem is derived and analyzed in Section IV. Section V presents the simulation results and verifies the optimality conditions numerically. The concluding remark is given in Section VI.

II. Space Maneuver Vehicle Trajectory Optimization Problem

In this section, the mission scenario investigated in this paper is presented. 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 50 km, 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. 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.

A. Problem Formulation

The overall formulation of the time-optimal SMV trajectory programming problem containing the flight dynamics is given as follows:

$$\begin{aligned}
&\text{minimize} \quad J = t_f \\
&\text{subject to} \quad \dot{r} = V\sin\gamma \\
&\qquad \dot{\theta} = \frac{V\cos\gamma\sin\psi}{r\cos\phi} \\
&\qquad \dot{\phi} = \frac{V\cos\gamma\cos\psi}{r} \\
&\qquad \dot{V} = \frac{T\cos\alpha - D}{m} - g\sin\gamma \\
&\qquad \dot{\gamma} = \frac{L\cos\sigma + T\sin\alpha}{mV} + \left(\frac{V^2 - gr}{rV}\right)\cos\gamma \\
&\qquad \dot{\psi} = \frac{L\sin\sigma}{mV\cos\gamma} + \frac{V}{r}\cos\gamma\sin\psi\tan\phi \\
&\qquad \dot{m} = -\frac{T}{I_{sp}g} \\
&\qquad \left[r(0), \theta(0), \phi(0), V(0), \gamma(0), \psi(0), m(0)\right] \\
&\qquad = \left[r_0, \theta_0, \phi_0, V_0, \gamma_0, \psi_0, m_0\right] \qquad (1)
\end{aligned}$$

where t_f is the terminal time. r, θ, ϕ, V, γ, ψ, and m represent the radial distance, longitude, latitude, velocity, flight-path angle, heading angle, and vehicle's mass, respectively. The control variables are the angle of attack α, bank angle σ, and thrust T. In this paper, the dynamic model given by (1) is abbreviated as $\dot{x} = f(x,u), x(0) = x_0$, where $x \in \Re^7$ and $u \in \Re^3$ denote the state and control variables, respectively. Detailed information including the aerodynamic model and atmospheric model can be found in [22].

B. Constraints

The spacecraft reconnaissance mission should satisfy strict box and path constraints to protect the structure of the vehicle. These constraints usually depend on the mission requirements. Regarding the box constraints, during the mission, each state and control variable should vary in the tolerable region and this can be described as $x_{\min} \le x \le x_{\max}$ and $u_{\min} \le u \le u_{\max}$. Besides, the path constraints that the vehicle must satisfy during the entire mission are the aerodynamic heating, dynamic pressure, and load factor, which can be summarized as

$$\begin{aligned}
&\dot{Q}_d = K_Q\rho^{0.5}V^{3.07}\left(c_0 + c_1\alpha + c_2\alpha^2 + c_3\alpha^3\right) < \dot{Q}_{d\max} \\
&P_d = \frac{1}{2}\rho V^2 < P_{d\max} \\
&n_L = \frac{\sqrt{L^2 + D^2}}{mg} < n_{L\max} \qquad (2)
\end{aligned}$$

where $Q_{d\max}$, $P_{d\max}$, and $n_{L\max}$ represent acceptable maximum heating rate, dynamic pressure, and load factor, respectively.

In order to describe the angular rate and range of the control variables, three first-order lag equations are appended to the equations of motion. Consequently, the control variables are divided into the actual controls α, σ, T and the demanded controls α_c, σ_c, T_c. In this way, the discontinuity in the control profiles can be removed

$$\begin{cases} \dot{\alpha} = k_\alpha(\alpha_c - \alpha) \\ \dot{\sigma} = k_\sigma(\sigma_c - \sigma) \\ \dot{T} = k_T(T_c - T) \end{cases} \quad \begin{cases} \alpha_{c\min} \le \alpha_c \le \alpha_{c\max} \\ \sigma_{c\min} \le \sigma_c \le \sigma_{c\max} \\ T_{c\min} \le T_c \le T_{c\max}. \end{cases} \tag{3}$$

III. Three-Layer-Hybrid Optimal Control Solver

Traditional optimal control solvers tend to be sensitive with respect to the initial guess value provided by the users. However, according to the work carried-out by Conway [6], for solvers based on evolutionary algorithms or metaheuristics, the problem needs to be discretized by a relatively small size of temporal set, and there is no guarantee of optimality. To address these problems, a new three-layer-hybrid optimal control solver was designed by combining the advantages of the traditional solvers and evolutionary-based solvers.

A. Initial Guess Generator

In the first layer of the proposed solver, an initial guess generator is designed to generate a reference trajectory, which can then be provided to the inner gradient-based solver as a start point of Newton iterations. The method used in this layer is an adaptive differential evolution algorithm similar to the framework proposed in [21]. This algorithm is then combined with a self-learning strategy proposed in [23]–[25]. Moreover, to handle different types of constraints entailing in the problem, a V-based constraint handling strategy is embedded in the algorithm framework, hence the name violation learning differential evolution (VLDE). For completeness, a brief description of this stochastic-based method is introduced in this section.

By using the discretization technique, the continuous-time optimal control problem given by (1) is then converted to a series of static NLPs. The discretization method used in this layer is the direct Runge–Kutta method [8] and the resulting static model can be written as

$$\begin{aligned} &\text{minimize} \quad J = t_f \\ &\text{subject to} \quad x_{k+1} = x_k + h_k \sum_{i=1}^{s} b_i f(x_{ki}, u_{ki}) \\ &\qquad x_{ki} = x_k + h_k \sum_{j=1}^{s} a_{ij} f\left(x_{kj}, u_{kj}\right) \\ &\qquad g(x_{ki}, u_{ki}) \ge 0 \\ &\qquad x(0) = x_0 \\ &\qquad i = 1, \ldots, s \qquad k = 0, \ldots, N_{k-1} \end{aligned} \tag{4}$$

where N_k is the number of discretized time nodes, while g stands for the inequality constraints described in (2) and (3). It can be seen from (4) that only the control variables are parameterized at temporal nodes $[t_0, t_1, \ldots, t_f]$. Then, the equations of motion are integrated with a numerical integration method, e.g., fourth-order Runge–Kutta method. In this way, the advantage is that the control box constraints [described in (2)] and equations of motion (1) can be satisfied automatically by initializing all population members within the specified lower and upper bounds and by integrating the dynamic model forward via numerical integration. Specifically, if the initial population contains *NP* individuals, then all the decision variables can be generated randomly according to the limits of demanded angle of attack, bank angle, and thrust given in (2) such that every decision variable can be in the feasible zone

$$\begin{aligned} \alpha_c &= \alpha_c^{\min} + \text{rand}(\cdot) \times \left(\alpha_c^{\max} - \alpha_c^{\min}\right) \\ \sigma_c &= \sigma_c^{\min} + \text{rand}(\cdot) \times \left(\sigma_c^{\max} - \sigma_c^{\min}\right) \\ T_c &= T_c^{\min} + \text{rand}(\cdot) \times \left(T_c^{\max} - T_c^{\min}\right). \end{aligned} \tag{5}$$

Remark 1: It is worth noting that by combining a shooting-scheme with the evolutionary solver, it offers the user more flexibility to choose the temporal set (e.g., the designer can use a relatively large temporal set to parameterize the dynamics). This is because the resulting NLP problem contains less number of equality constraints compared with collocation discretization scheme. For collocation methods, both of the state and control variables will be discretized and consequently, the equations of motion are transcribed to a series of algebraic equations (equality constraints). This implies that the resulting number of optimization parameters and equality constraints tends to be larger and cannot be satisfied automatically. For instance, if an optimal control problem contains n_s state variables, n_c control variables and is parameterized using N_k collocation points, the number of decision variables of the NLP solver is $n_s \times (N_k + 1) + (n_c \times N_k) + 1$, which is large compared with shooting scheme since in a shooting method, only the control variable is parameterized. When an NLP problem contains too many optimization parameters and equality constraints, it tends to cost the stochastic-based solver significant number of iterations to capture the true behavior of the model and fail to satisfy all of the constraints.

It should be noted that the constraint handling strategy used in this paper is based on the violation degree of constraints V_{ol}. Using this approach, each individual among the population can be associated with all the constraints; and the value of the violation function can directly reflect the magnitude of the solution infeasibility. For example, the violation degree for relation "$\le$" [i.e., $g_j(u) \le g_j^*$, where g_j^* is the required upper bound] can be defined as

$$\mu_{g_j}(u) = \begin{cases} 0, & g_j(u) \le g_j^* \\ \frac{g_j(u) - g_j^*}{g_j\max - g_j^*}, & g_j^* \le g_j(u) \le g_j^{\max} \\ 1, & g_j(u) \ge g_j^{\max} \end{cases} \tag{6}$$

where $\mu_{g_j}(u)$ is a continuous function and $g_j(u)$ is the value of jth constraint for each individual, while the tolerance range should be $(g_j^*, g_j^{\max})$.

Consequently, the value of violation function V_{ol} can be obtained via $V_{\text{ol}} = \sum_{j=1}^{I} \mu_{g_j}(u) + \sum_{k=1}^{E} \mu_{h_k}(u)$, where h

Algorithm 1 Main Framework of the Proposed VLDE

Input: the maximum number of generations G_{max}, number of temporal nodes N_k;
Output: the best candidate x_{best} among the final population $P_{G_{max}}$;
/*Initialization*/
Initialization: generate the initial population P_0 with N randomized individuals;
/*Main Loop*/
while $G < G_{max}$ **do**
    (a). choose the best number of the current population as P_G;
    (b). generate the offspring generation Q_G
        Q_G = VLDE offspring-creation (P_G);
    (c). L_G = learning strategy (P_G, Q_G);
    (d). set $P_G = P_G \cup Q_G \cup L_G$;
    (e). perform the elite selection based on the new dominant rule to get P_{G+1};
    (e). set $G = G + 1$;
end while

stands for the equality constraints. I and E are the number of inequality and equality constraints, respectively. As a result, the selection procedure is depended on the value of V_{ol}. The feasible individual ($V_{\text{ol}} = 0$) can always dominate the infeasible one ($V_{\text{ol}} \neq 0$), while the individual with small violation degree can always dominate the one with large violation degree and then be selected.

The main framework of the proposed VLDE method is constructed in Algorithm 1, from where it can be seen that VLDE applies the elitism strategy similar to that of traditional DE. The main new contributions in the VLDE algorithm lie in its offspring-creation component, where a simplex-based direct search operation is coupled with traditional generic operations, such as crossover and mutation. In addition, a learning strategy is also designed according to the nature of the SMV optimal control problem. In this way, the diversity of the offspring generation can be further improved, thus helping avoid the premature convergence of VLDE. It was shown in [21] that by applying these modifications, the convergence speed of the VLDE method for solving spacecraft trajectory optimization problem is effectively improved. Hence, this method is chosen as the outer initial guess generator in this paper.

Since the aim of the initial trajectory generator is to provide reference state and control sequences to the inner gradient solver, in order to keep a balance between the quality of solution and the computational burden, $G_{\max}$ is fixed as 100.

B. Improved Gradient-Based Inner Solver

After generating the reference state and control sequences, these results are provided to the inner gradient-based optimizer (layer 2). In the second layer, in order to achieve a higher accuracy, pseudospectral discretization scheme is applied to transform the dynamics. That is, both the control and state variables are parameterized at temporal nodes. It is discussed in [26] and [27] that the approximation accuracy of pseudospectral methods can be directly controlled via the number of collocation nodes. For example, the approximate error order for Gauss pseudospectral method is $\mathcal{O}(N_k^{3-p})$ [27], whereas for Radau pseudospectral method, this order becomes $\mathcal{O}(N_k^{2-\eta})$ [26].

The optimization methodology used in the second layer is an improved gradient algorithm based on the IP method and SQP. The general idea of the proposed algorithm (referred to as the IPSQP algorithm) is to use the SQP method, where the quadratic programming (QP) subproblem is solved using an IP method [28]. The general quadratic form of the transcribed trajectory optimization problem is given

$$\begin{aligned} \min\ & \frac{1}{2} dx^T H(x_k, \lambda_k, u_k) dx + \nabla f(x_k)^T dx \\ & h(x_k) + \nabla h(x_k) dx = 0 \\ & g(x_k) + \nabla g(x_k) dx \leq 0 \end{aligned} \tag{7}$$

where $h(\cdot) = (h_1(\cdot), h_2(\cdot), \ldots, h_E(\cdot))^T$ and $g(\cdot) = (g_1(\cdot), g_2(\cdot), \ldots, g_I(\cdot))^T$. H stands for the Hessian matrix. At a fixed iteration time k, the IP strategy is used to solve the QP model given by (7). Then, the formulation of the improved gradient method can be formulated as follows:

$$\begin{aligned} \min\ & \frac{1}{2} dx^T H(x_k, u_k) dx + \nabla f(x_k)^T dx - \mu_k \Sigma_{j=1}^{m} \log\left(s_j^k + d_j^s\right) \\ \text{s.t.}\ & g(x_k) + \nabla g(x_k) dx + s_k + e^T ds = 0 \\ & h(x_k) + \nabla h(x_k) dx = 0 \\ & dx \in \Re^n,\ ds \in \Re^m \end{aligned} \tag{8}$$

where the primal and dual variables are dx and ds, respectively. The last two terms $s_k + e^T ds$ in the equality constraint are considered as the slack variables. The algorithm consists of two nested loops and correspondingly, it contains two iteration indices. To distinguish these two iterations, the internal iteration index is defined as l while the external iteration number is defined as k. Specifically, the primal, slack, and dual variables in the inner iteration are $dx_{k,l}$, $ds_{k,l}$, $d\lambda_{k,l}$, and $du_{k,l}$, respectively. The main advantage of this algorithm is that the user can control the inner loop by setting the termination conditions at any time. 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.

The Karush–Kuhn–Tucker (KKT) system of (8) is described in (9), and it is solved iteratively

$$\begin{aligned} &\begin{pmatrix} H_k & 0 & \nabla h(x_k)^T & \nabla g(x_k)^T \\ 0 & Du_{k,l} & 0 & Ds_{k,l} \\ \nabla h(x_k) & 0 & 0 & 0 \\ \nabla g(x_k) & I & 0 & 0 \end{pmatrix} \begin{pmatrix} \Delta dx_{k,l} \\ \Delta ds_{k,l} \\ \Delta d\lambda_{k,l} \\ \Delta du_{k,l} \end{pmatrix} \\ &= \begin{pmatrix} -H_k dx_{k,l} - \nabla f(x_k) - \nabla h(x_k)^T d\lambda_{k,l} - \nabla g(x_k)^T du_{k,l} \\ -Ds_{k,l} du_{k,l} + \mu_{k,l} e \\ -h(x_k) - \nabla h(x_k) dx_{k,l} \\ -g(x_k) + \nabla g(x_k) dx_{k,l} - s_k + e^T ds_{k,l} \end{pmatrix} \end{aligned} \tag{9}$$

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, the new iteration can be calculated by

$$\begin{aligned} dx_{k,l+1} &= dx_{k,l} + \alpha_{k,l}\Delta dx_{k,l} \\ du_{k,l+1} &= du_{k,l} + \alpha_{k,l}\Delta du_{k,l} \\ ds_{k,l+1} &= ds_{k,l} + \alpha_{k,l}\Delta ds_{k,l} \\ d\lambda_{k,l+1} &= d\lambda_{k,l} + \alpha_{k,l}\Delta d\lambda_{k,l} \end{aligned} \tag{10}$$

where the step length parameter $\alpha_{k,l} \in (0, 1]$ should be chosen to ensure that the merit function achieves sufficient decrease but the step is not too short. In this paper, the l_∞-merit function is implemented in order to measure the progress of each iteration and it can be written as

$$\begin{aligned} &M_{\mu,r}(x, s, \lambda, u, dx, ds, d\lambda, du) \\ &\quad = \frac{1}{2}dx^T H dx + \nabla f(x)^T dx - \mu_k \Sigma_{j=1}^{m} \log\left(s_j + d_j^s\right) \\ &\quad\quad + \lambda \Sigma_{i=1}^{l}(h_i(x) + \nabla h_i(x))dx \\ &\quad\quad + r\|g(x) + \nabla g(x)dx + s + e^T ds\|_\infty \end{aligned} \tag{11}$$

where r is the penalty factor and μ is a barrier parameter. The line search process shown in (10) should ensure that there is a progress on the merit function for each iteration and therefore, the Goldstein conditions is applied

$$\begin{aligned} &M_{\mu,r}(x_k, s_k, \lambda_k, u_k) + c_1\alpha_k \nabla M_{\mu,r}^T \Delta d_k \\ &\quad \le M_{\mu,r}(x_k + \alpha_k dx_k, s_k + \alpha_k ds_k, \\ &\qquad\qquad \lambda_k + \alpha_k d\lambda_k, u_k + \alpha_k du_k) \\ &\quad \le M_{\mu,r}(x_k, s_k, \lambda_k, u_k) + c_2\alpha_k \nabla M_{\mu,r}^T \Delta d_k \end{aligned} \tag{12}$$

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 sufficient decrease condition while the first term of the inequality is to control the step length so that it will not be chosen too short.

In order to better show the structure of this two nested gradient optimization algorithm, the overall procedure is illustrated in the pseudocode (see Algorithm 2).

Remark 2: It should be noted that commonly the first initial guess and active set provided by the user are far from the optimal solution. Therefore, the Lagrangian multipliers calculated by using SQP are inaccurate. If the quadratic model is solved using SQP and active set, it usually takes the solver a large number of iteration to converge. However, by applying the inner loop (controlled by the number of $l_{\max}$) in the proposed algorithm and the reference solution generated from layer 1, the identification of active set and the solution finding can be more accurate. Moreover, the convergence ability can also be improved and the computational burden can be decreased at the same time (less Newton iteration).

Algorithm 2 Pseudocode for the IPSQP Method

```
1:  procedure (Two nested structure)
2:      Choose starting values z_0 = (x_0, u_0, λ_0, s_0)
3:      for k := 0, 1, 2, ... do
4:          (a). Check stopping criteria for the outer loop
5:          (b). Choose dx_{k,0}, du_{k,0}, dλ_{k,0} and ds_{k,0}
6:          for l := 0, 1, 2, ..., l_max do
7:              i. Determine Du_{k,l}, Ds_{k,l} and μ_{k,l}.
8:              ii. Solve the KKT system described in Eq.(9).
9:              iii. Apply the line search algorithm shown in
10:                 Eq.(10).
11:             iv. If the inner loop solution can satisfy the
12:                 stopping condition of QP, break for-loop;
13:         end for
14:         (c). Find step length for the outer loop such that
15:           the merit function can have a proper
16:           improvement.
17:         (d). Update the current searching point and go back
18:           to line 3.
19:     end for
20:     Output the optimal solution
21: end procedure
```

C. Mesh Refinement

For direct transcription methods (e.g., direct multiple shooting and direct collocation), in general, the accuracy of the solution depends largely on the mesh refinement procedure [5], [29], [30]. The aim for carrying out mesh refinement is to determine whether the current mesh grid is proper and update the mesh grid so that the resulting mesh can contain a small number of temporal points and the mesh distribution is dense where the discontinuity order is high. Therefore, the third layer of the designed hybrid optimal control solver is coupled with the second layer (inner gradient-based optimizer) so as to refresh the mesh grid. The idea applied to do the mesh update is the *hp*-strategy. Detailed information including the analytical formulation of the *hp*-strategy can be found in [22] and [30].

Let ε stands for an accuracy tolerance for the discrete algebraic function constraint. The errors of the dynamic equations and path constraints at the sth collocation point, kth time interval are $a_s^{(k)}$ and $b_s^{(k)}$, respectively. Therefore, the maximum error $e_{\max}^{(k)}$ in the kth time interval can be approximated as $e_{\max}^{(k)} = \max[a_s^{(k)}, b_s^{(k)}]$. If the equation $e_{\max}^{(k)} \le \varepsilon$ can be satisfied, then the algorithm will stop the iteration because collocation points in the kth interval can reach the tolerance. Otherwise, it should be divided into subintervals or add more collocation points.

Suppose $k_{\max}^{(k)}$ and $\bar{k}^{(k)}$ are the maximum curvature and average curvature of all the nodes, respectively. Furthermore, let $r^{(k)} = ([k_{\max}^{(k)}]/[\bar{k}^{(k)}])$ be the ratio of the maximum to the mean curvature. Setting the tolerance of curvature as $r_{\max}$ and if $r^{(k)} \ge r_{\max}$, then the trajectory in this time interval tends to have oscillations and it should be divided into new subinterval. The number of the subinterval n_k is determined using $n_k = \text{ceil}(\log(e_{\max}^{(k)}/\epsilon))$, where the function of ceil$(\cdot)$ is to round a number to the next larger integer. On the other hand, if the tolerance can satisfy $r^{(k)} < r_{\max}$, the trajectory tends to be flat in this time interval and the accuracy can be improved by adding more collocation points. The number of points that should be added is determined by $N_k = N_k + \text{ceil}(\log(e_{\max}^{(k)}/\epsilon))$.

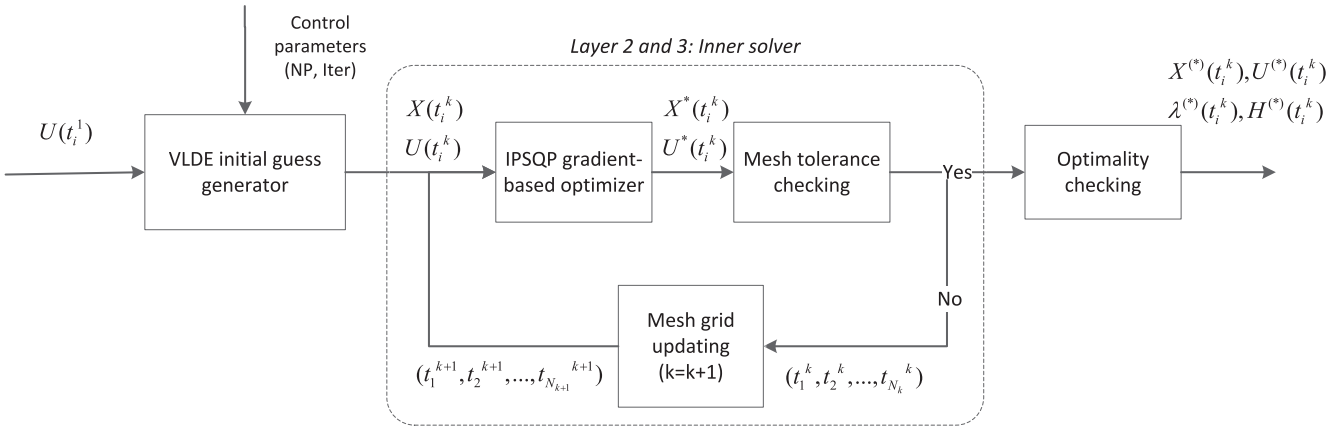

Fig. 1. Flow chart of the designed solver.

Remark 3: In order to improve the efficiency of the proposed solver, in the simulation, each refinement iteration will use the previous mesh history as the start point to do the solution finding. The stopping criteria depends on the maximum number of iteration and the accuracy tolerance. The algorithm will carry on until one of the stopping criteria can be satisfied.

D. Algorithm Framework

The framework of the proposed three-layer-hybrid optimal control solver is presented in Fig. 1.

IV. Optimality Verification

Since the SMV trajectory planning problem is formulated as an optimal control problem, to judge the quality of the solution generated by the proposed three-layer-hybrid optimal control solver, the first-order necessary conditions should be used. According to the problem formulation given by (1), the Hamiltonian function H_{am} is constructed as $H_{\text{am}} = \Phi(x, u, t; t_0, t_f) + \lambda^T(t) f(x, u, t; t_0, t_f)$. Taking into account the path constraints, the augmented Hamiltonian is defined as

$$\begin{aligned} H_{\text{am}}(x, \lambda, u, \mu, t; t_0, t_f) = {} & \Phi(x, u, t; t_0, t_f) \\ & + \lambda^T(t) f(x, u, t; t_0, t_f) \\ & - \mu^T(t) C(x, u, t; t_0, t_f) \end{aligned} \quad (13)$$

where Φ is the Lagrange form cost function, f is the right hand side of the equations of motion, $\lambda(t) = [\lambda_r, \lambda_\theta, \lambda_\phi, \lambda_V, \lambda_\gamma, \lambda_\psi, \lambda_m, \lambda_\alpha, \lambda_\sigma, \lambda_T]^T \in \Re^{10}$ is the costate corresponding to the dynamic equations and $\mu(t) \in \Re^3$ is the Lagrange multiplier associated with the path constraints.

A. First-Order Necessary Conditions

The continuous-time first-order necessary conditions in terms of costate and hamiltonian can be written as

$$\begin{aligned} \lambda(t_0) &= -\frac{\partial J}{\partial x(t_0)} + \upsilon^T \frac{\partial \phi}{\partial x(t_0)} \\ \lambda(t_f) &= -\frac{\partial J}{\partial x(t_f)} + \upsilon^T \frac{\partial \phi}{\partial x(t_f)} \end{aligned}$$

$$\begin{aligned} H_{\text{am}}(t_0) &= \frac{\partial J}{\partial t_0} - \upsilon^T \frac{\partial \phi}{\partial t_0} \\ H_{\text{am}}(t_f) &= \frac{\partial J}{\partial t_f} - \upsilon^T \frac{\partial \phi}{\partial t_f} \end{aligned} \quad (14)$$

where ϕ is the boundary condition and υ is the Lagrange multiplier associated with the boundary condition. Besides, the first-order 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 control variables

$$\frac{\partial H_{\text{am}}}{\partial u} = \frac{\partial g}{\partial u} + \left(\frac{\partial f}{\partial u}\right)^T \lambda - \left(\frac{\partial C}{\partial u}\right)^T \mu = 0. \quad (15)$$

It is obvious that the first term $(\partial g/\partial u) = 0$ and therefore, (15) can be rewritten as

$$\begin{cases} \frac{\partial \lambda_\alpha [K_\alpha(\alpha_c - \alpha)]}{\partial \alpha_c} - q_{\alpha_c} = 0, & \text{(with respect to } \alpha_c) \\ \frac{\partial \lambda_\sigma [K_\sigma(\sigma_c - \sigma)]}{\partial \sigma_c} - q_{\sigma_c} = 0, & \text{(with respect to } \sigma_c) \\ \frac{\partial \lambda_T [K_T(T_c - T)]}{\partial T_c} - q_{T_c} = 0, & \text{(with respect to } T_c) \end{cases} \quad (16)$$

where $q_u = [q_{\alpha_c}, q_{\sigma_c}, q_{T_c}]^T$ is the Lagrange multiplier with respect to the control path constraints. Also, based on the KKT complementary condition, the control multipliers should satisfy the following equation:

$$q_u \begin{cases} \leq 0 & \text{if, } u = u_{\min} \\ = 0 & \text{if, } (u_{\min} < u < u_{\max}) \\ \geq 0 & \text{if, } u = u_{\max}. \end{cases} \quad (17)$$

Equation (17) implies when the control constraints become active, the corresponding multipliers become nonzero.

B. Terminal Transversality Conditions

The following proposition illustrates the property of costate variables at the terminal time instant.

Proposition 1: Consider the time-optimal spacecraft trajectory optimization problem has a feasible optimal solution: the costate value $[\lambda_\theta, \lambda_\phi, \lambda_V, \lambda_\psi, \lambda_m, \lambda_\alpha, \lambda_\sigma, \lambda_T]$ must be zero when $t = t_f$.

Proof: Based on (14), the transversality condition at the final time instant has the following form:

$$\lambda(t_f) + \frac{\partial J}{\partial x(t_f)} - \upsilon^T \frac{\partial \phi}{\partial x(t_f)} = 0.$$

For the time-optimal spacecraft trajectory planning problem, there are no constraints for $\theta, \phi, V, \psi, \alpha, \sigma, T$ at t_f. Therefore, the costate value for each state variable at the final time should hold

$$\begin{cases} \lambda_\theta(t_f) = \frac{\partial J}{\partial \theta} \mid_{t=t_f} = 0 \\ \lambda_\phi(t_f) = \frac{\partial J}{\partial \phi} \mid_{t=t_f} = 0 \\ \lambda_V(t_f) = \frac{\partial J}{\partial V} \mid_{t=t_f} = 0 \\ \lambda_\psi(t_f) = \frac{\partial J}{\partial \psi} \mid_{t=t_f} = 0 \\ \lambda_m(t_f) = \frac{\partial J}{\partial m} \mid_{t=t_f} = 0 \\ \lambda_\alpha(t_f) = \frac{\partial J}{\partial \alpha} \mid_{t=t_f} = 0 \\ \lambda_\sigma(t_f) = \frac{\partial J}{\partial \sigma} \mid_{t=t_f} = 0 \\ \lambda_T(t_f) = \frac{\partial J}{\partial T} \mid_{t=t_f} = 0 \end{cases} \tag{18}$$

which completes the proof. ■

Applying the terminal transversality condition can give some indications of the final value of the dual variables λ that can be used later to verify the numerical results.

C. Hamiltonian Function Condition

The following proposition gives the nature of the Hamiltonian function with respect to time.

Proposition 2: Assume $x^*(t)$ and $u^*(t)$ are the optimal state and control variables, the corresponding Hamiltonian function should satisfy $H_{\text{am}}(x^*(t), u^*(t)) = -1$ for $t \in [t_0, t_f]$.

Proof: Since the final time is a free variable, there is a stationary condition for the H_{am} function. This condition is obtained by applying the endpoint Lagrangian equation (e.g., $H_{\text{am}}(t_f) - (\partial J/\partial t_f) + \upsilon^T(\partial \phi/\partial t_f) = 0$). For the problem studied in this paper, this equation can be rewritten as

$$\begin{aligned} H_{\text{am}}(X(t_f), t_f, \upsilon) &= -\frac{\partial J}{\partial t_f} + \upsilon_r \frac{\partial(r_f - r(t_f))}{\partial t_f} \\ &\quad + \upsilon_\gamma \frac{\partial(\gamma_f - \gamma(t_f))}{\partial t_f} \\ &= -1. \end{aligned} \tag{19}$$

Equation (19) implies that the final value of the Hamiltonian function should be -1 for this problem. Then the Hamiltonian evolution equation is used to demonstrate the behavior of the Hamiltonian with respect to time such that

$$\frac{\partial H_{\text{am}}}{\partial t} = 0. \tag{20}$$

Equation (20) means H_{am} is not explicitly a function of time and $H_{\text{am}} = c$, where c is a constant. Combining (19) and (20), it is clear that the following equation should be satisfied:

$$H_{\text{am}}(x^*(t), u^*(t)) = -1, \quad \forall t \in [t_0, t_f]. \tag{21}$$

■

Therefore, it is obvious that the Hamiltonian function should be -1 during the entire time history. The aim of the analysis carried-out in this section is to verify the first-order quality of the numerical solution calculated using the designed optimal control solver.

D. Properties of the Control Variable

Similarly, the optimal control variable has the following property.

Proposition 3: If the time-optimal spacecraft trajectory optimization problem has the optimal control solution $u^*(t)$, then $u^*(t)$ can be expected to have a "bang–bang" behavior for all $t \in [t_0, t_f]$.

Proof: The control variables should be chosen such that the augmented Hamiltonian function can be minimized. Since the rate constraint of the control variable is achieved using the first-order lag equations [shown in (3)], the path constraints do not involve the control variables explicitly, which means the optimal solution may contain corners. Moreover, according to (3), the demanded control appears linearly in the differential equations. Therefore, to minimize the Hamiltonian function H_{am} with respect to the demanded control, 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, which can be expressed as

$$u^*(t) = \begin{cases} u_{\min} & \text{if}, \quad \rho < 0 \\ u_{\max} & \text{if}, \quad \rho > 0. \end{cases} \tag{22}$$

In (22), ρ is the switching function with the expression $\rho = -\lambda' K - q_u$, where $\lambda' = [\lambda_\alpha, \lambda_\sigma, \lambda_T]$ and $K = [k_\alpha, k_\sigma, k_T]$, respectively. The sign of the switching function dominates the magnitude of the controls according to (22). ■

Consequently, as for the SMV model considered in this paper, it can be expected to experience a switching structure in terms of the demanded angle of attack, bank angle, and thrust profiles.

E. Bellman's Principle

Another way to verify the optimality is Bellman's principle. 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. This principle is also the main theory of dynamic programming in terms of optimality. If several time points on the original trajectory are selected as initial conditions and the results show that there is no better or different solutions, then the Bellman's optimality principle can be satisfied.

V. Simulation Results

A. Parameter Setting

All the mission-dependent and vehicle-dependent parameters, including the initial, terminal boundary conditions, box constraints, and aerodynamic coefficients of the skip problem can be found in [22]. Regarding the path constraints for mission 1, the maximum allowable heating, dynamic pressure, and load factor are set as: $Q_{\max} = 200BTU$; $Pd_{\max} = 13406.4583Pa$; and $nl_{\max} = 2.5$, respectively.

B. Optimal Solution

According to the dynamic model, objective function and path constraints that were given in Section II, the numerical

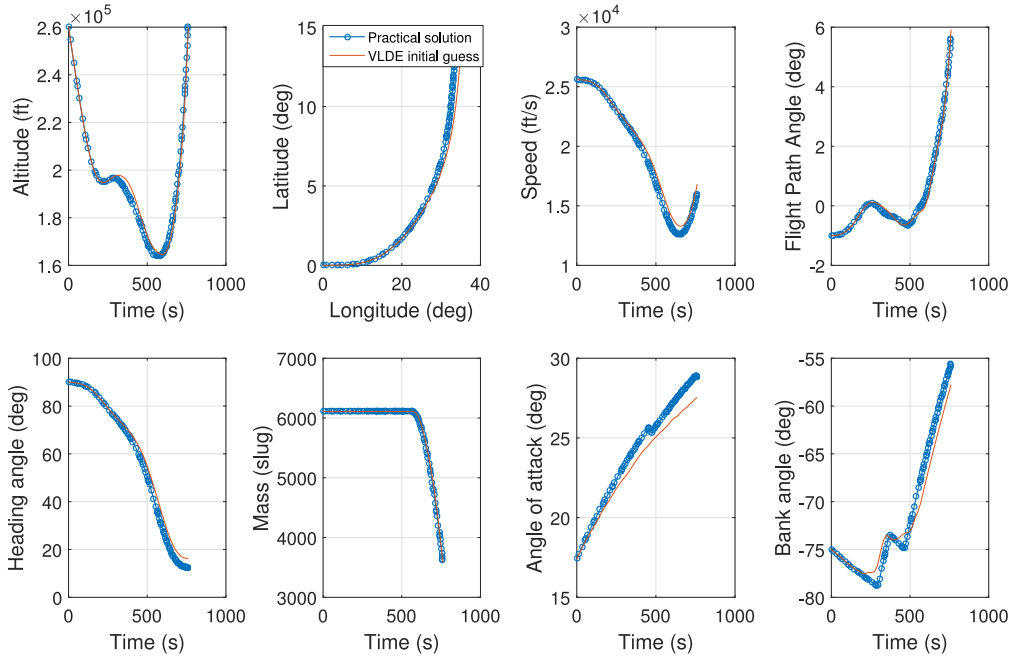

Fig. 2. Time history for states and controls (mission 1: $Q_{\max} = 200BTU$; $Pd_{\max} = 13406.4583Pa$; and $nl_{\max} = 2.5$).

solutions obtained using the VLDE-based initial guess generator and the hybrid solver are shown in Fig. 2. The characteristic arcs of the obtained trajectories are analyzed in this section. The trajectory is split into two subintervals: 1) descent and 2) climb. In the descent phase, to reach the minimum predetermined altitude (around 50 km above sea level) and minimize the objective function (final time), Fig. 2 shows that the SMV 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 (a small hop) before reaching the minimum altitude point. This is because if the spacecraft descends without increasing the flight path angle, then the dynamic pressure and load factor constraints may become active. To preserve kinetic energy and satisfy the path constraints, the curvature of the altitude is decreased such that the SMV can have sufficient time to use the drag force (i.e., slow its speed down). Therefore, the structure integrity of the vehicle can be guaranteed at the expense of the objective function.

In the climb phase, the SMV fires it engine so that the vehicle can accelerate and have enough kinetic energy to return back to orbit. The trend for the angle of attack can be seen in Fig. 2, where the angle of attack is increased during the whole climbing phase. The reason is that in the climbing phase, without violating the path constraints it can have positive influences in terms of acceleration.

To further test and analyze the performance of the proposed hybrid strategy, another mission scenario, which contains more strict limits on the path constraints was carried out using the designed hybrid optimal solver. In this case, the maximum allowable heating, dynamic pressure, and load factor for mission 2 are restricted to: $Q_{\max} = 150BTU$; $Pd_{\max} = 11970.05Pa$; and $nl_{\max} = 2.0$, respectively. The time history of the optimization variables and path constraints are shown in Figs. 3 and 4.

As can be seen from Figs. 3 and 4, the proposed hybrid strategy can still generate high quality solutions without violating path constraints and box constraints. Compared with the results shown in Fig. 2, the solution obtained in Figs. 3 and 4 shows a slight difference. The reason of the additional hop in the altitude profile after around 400 s is to prevent the load factor constraint from becoming active. Therefore, based on there observations and results, it can be concluded that the initial guess generator embedded in the algorithm framework can have positive influences for the inner gradient-based solver in terms of increasing convergence ability and getting rid of infeasibility.

Regarding to the mesh refinement process, the way that this strategy adds points in a specific time interval or divides an interval into subintervals was described in Section III-C. Using the *hp* strategy, the time history for the states and controls can be much smoother. This can be seen in Figs. 2–4, where the distribution of grid points tends to be dense at the areas having a high value of curvature (the trajectory at those areas tends to have more oscillations), while the distribution of temporal points tends to be sparse at those flat areas. It is important to combine the mesh refinement with a specific optimal control solver since for most real-world problems, the optimal state and control profiles may contain discontinuous points or nonsmooth segments. If a fixed temporal set is used, it is usually hard to detect all the discontinuity or nonsmoothness, and the obtained solution may fail to describe the true behaviors of the state and control variables.

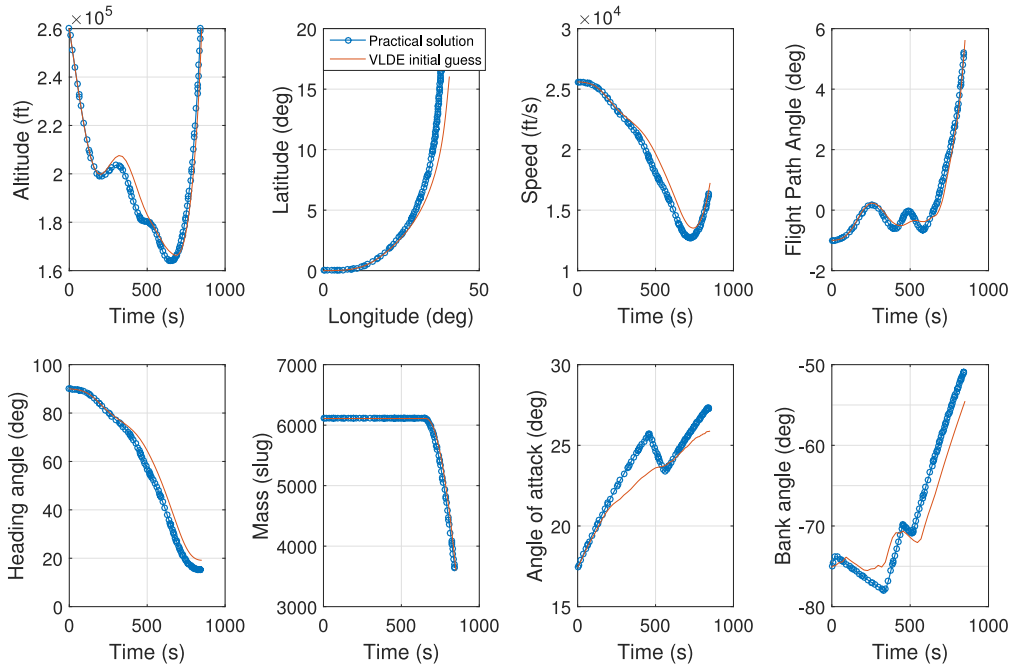

Fig. 3. Time history for states and controls (mission 2: $Q_{\max} = 150BTU$; $Pd_{\max} = 11970.05Pa$; and $nl_{\max} = 2.0$).

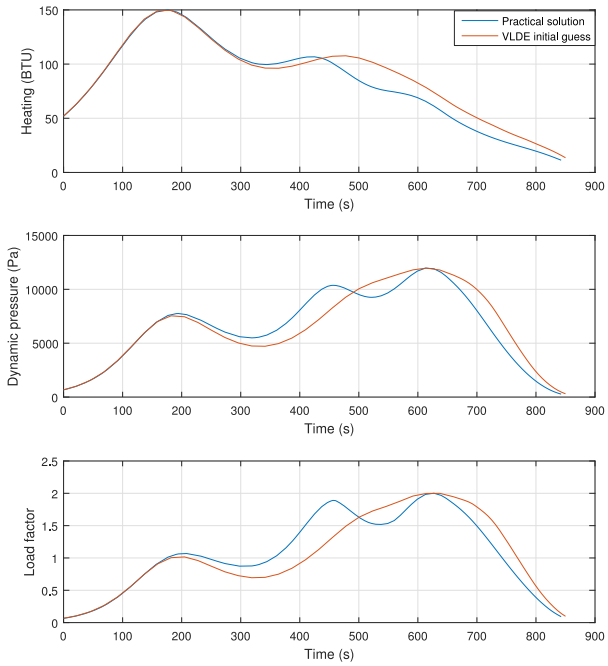

Fig. 4. Time history for three path constraints (mission 2).

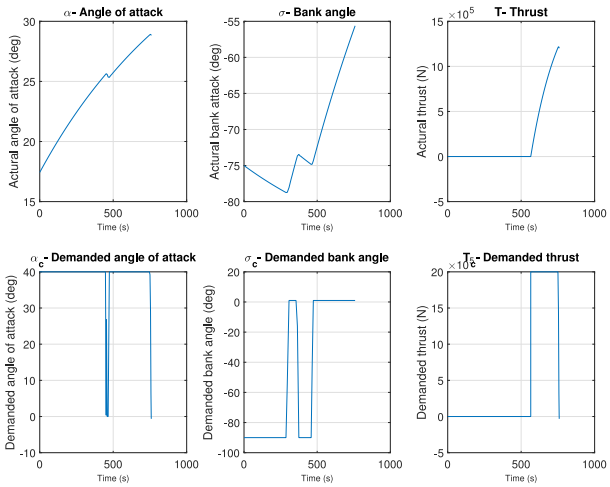

Fig. 5. Actual and demanded control profiles (mission 1).

C. Verification of Optimality

After generating all the state and control profiles using the proposed solver, the next step is to verify the optimality of the obtained solutions. The conditions used to analyze the optimality were presented in Section IV. For the simulation results shown in Fig. 2, the corresponding actual and demanded control profiles, including angle of attack, bank angle, and thrust are plotted in Fig. 5. As discussed in Section IV, since the demanded control variables that appear on the right hand side of the equations of motion are linear and do not appear in the path constraints, a bang–bang behavior with respect to the demanded control profile can be expected. This behavior can be used partly to show the optimality of the calculated trajectories. Moreover, by introducing the three lag equations (3), the actual controls become much smoother (no discontinuous point), which can offer more flexibilities in terms of the design of online guidance law.

Moreover, to check the terminal transversality conditions, the final values of the costate (λ) corresponding to the state obtained from the optimization process are calculated, such that $\lambda_\theta(t_f) = -5.33\times10^{-10}$, $\lambda_\phi(t_f) = -2.28\times10^{-9}$, $\lambda_V(t_f) = -4.93\times10^{-2}$, $\lambda_\psi(t_f) = 7.52\times10^{-10}$, $\lambda_\alpha(t_f) = 6.09\times10^{-21}$, $\lambda_\sigma(t_f) = 2.17\times10^{-17}$, and $\lambda_T(t_f) = -3.14\times10^{-23}$.

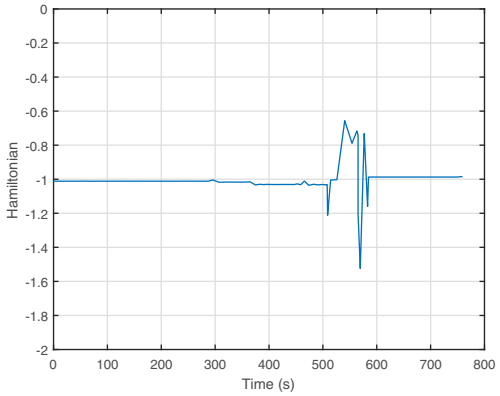

Fig. 6. Hamiltonian function for mission scenario 1.

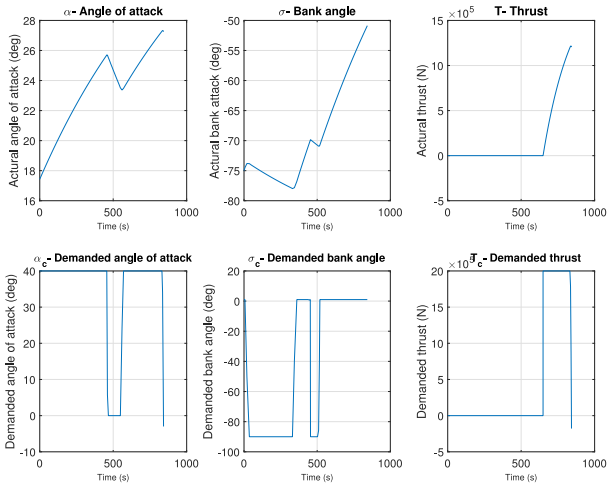

Fig. 7. Actual and demanded control profiles (mission 2).

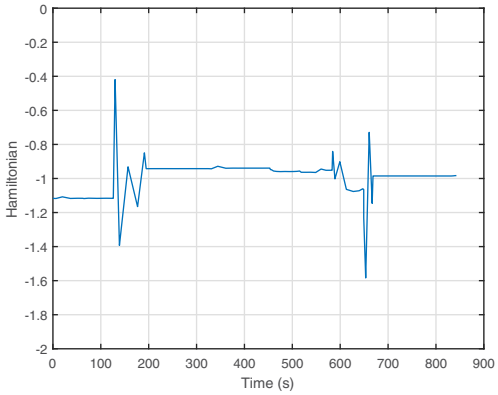

Fig. 8. Hamiltonian function for mission scenario 2.

The Hamiltonian value condition shows that the Hamiltonian function should be -1 at the final time [i.e., $H_{\text{am}}(t_f) = -1$], see Section IV-C. From the Hamiltonian evolution equation, it can be shown that the Hamiltonian is constant during the whole time history. Therefore, combining these two conditions, the Hamiltonian should keep -1, as illustrated in Fig. 6.

Fig. 6 and the final costate values illustrate the optimality of the obtained solutions. Specifically, the value of the Hamiltonian function generated using the proposed approach tends to remain flat around -1 with small variance, while the final costate values for states which have free final conditions are equal to 0 or approximately 0. That means the performance of the designed strategy is closer to the theoretical behavior and it further confirms that the newly designed solver is feasible and effective for handling SMV trajectory optimization problem.

For the mission scenario containing strict path constraints, the control profiles are plotted in Fig. 7. It is clear that the numerical solution can still satisfy the control profile conditions discussed in Section IV. In terms of the Hamiltonian profile shown in Fig. 8, again, the theoretical behaviors discussed in Section IV can be satisfied.

In order to verify Bellman's principle described in [31], and presented in Section IV of this paper, 15 time points on the obtained trajectory were selected randomly as initial conditions for the simulation. The results show that there is no better or different solution for both mission cases.

Remark 4: It is clear from Figs. 6 and 8 that the Hamiltonian profiles do not stay flat during the entire time period and contain some oscillations. For example, in Fig. 6, there are some fluctuations around 500 s, while in Fig. 8, the trajectory tend to oscillate in the [100 s, 200 s] and [550 s, 650 s] regions. This is because during these time periods, the path constraints (see Fig. 4) become active. That is, in the Hamiltonian equation, the multipliers $\mu^T(t)$ associated with C (path constraints) become nonzero. More precisely, at these time periods, the numerical solutions will lose some optimality.

Remark 5: One main advantage of the proposed optimal control solver is that it is possible for the user to analyze the optimality of the obtained solution. Since the inner solver is based on gradient-based optimization methods, it is possible to calculate the information about the Lagrange multipliers, which implies that the adjoint variable associated with each state can then be calculated via the mapping principle discussed in [7]. Therefore, the first-order necessary conditions for the general optimal control problems can be verified numerically.

D. Comparison With Existing Evolutionary Solvers

In this section, comparative studies were performed to analyze the optimal trajectories achieved by applying the VLDE evolutionary solver with other typical heuristic strategies. For example, the PSO method reported in [19], and a DE design studied in [32]. These strategies use stochastic optimization processes and were shown as promising methods to calculate the optimal state and control trajectories. For the purpose of comparison, all the methods are applied to solve the first mission scenario. Fig. 9 illustrates the time history with respect to the state variable calculated using different evolutionary algorithms, whereas Fig. 10 gives the control and path constraint trajectories.

As can be seen from Figs. 9 and 10, all the solutions calculated by applying these three methods can be accepted as feasible solutions. The VLDE approach can perform better

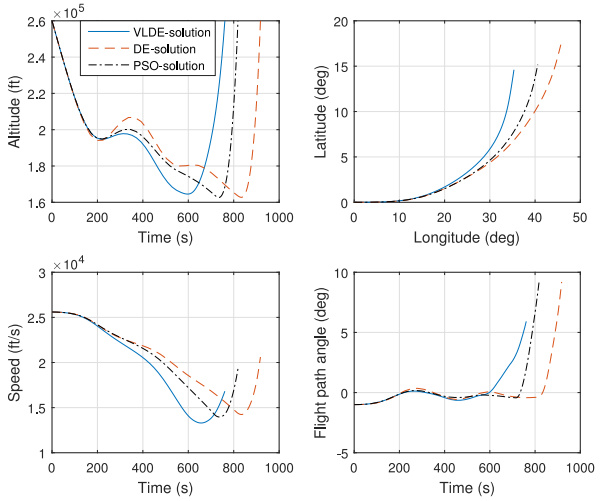

Fig. 9. Results obtained using different heuristic solvers.

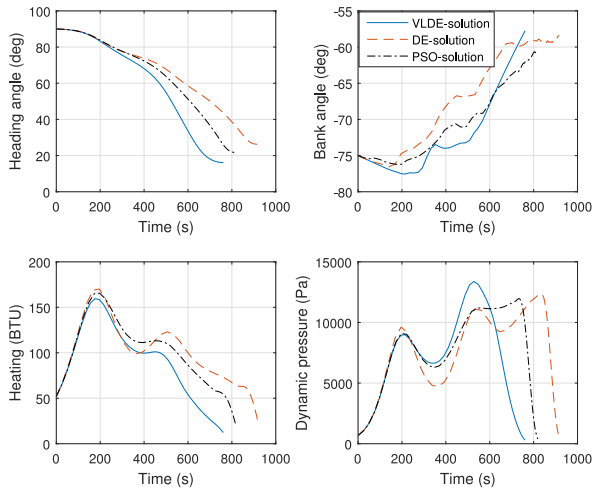

Fig. 10. Results obtained using different heuristic solvers.

than its counterparts in terms of achieving better fitness values for this mission case. Moreover, it is obtained that the final boundary error values calculated using different global optimization methods are $e_{r\_\mathrm{VLDE}} = 0.08$ ft, $e_{r\_\mathrm{PSO}} = 14.16$ ft, and $e_{r\_DE} = 203.88$ ft, respectively. This further confirms that the VLDE method investigated in this paper can better lead the boundary error to a small value without violating various path constraints for the SMV trajectory planning problem.

E. Dispersion Models

The aim of dispersion simulations is to illustrate the stability of the proposed three-layer-hybrid optimal control solver in the presence of significant deviations in trajectory initial state variables (x_0) and vehicle uncertainties [33]. All the random initialization data used in the dispersion model are tabulated in Table I. Furthermore, the vehicle mass was perturbed uniformly up to 5% with the nominal mass, which gives a range of value of $95\%m_0$–$105\%m_0$.

The dispersion simulation was carried out for the mission scenario 1 and 1000 Monte-Carlo simulations were performed.

TABLE I
DISPERSIONS IN THE ENTRY INITIAL CONDITIONS

State/parameter	Distribution	3-σ range
Altitude,ft	Zero-mean Gaussian	500
Longitude,deg	Zero-mean Gaussian	0.0749
Latitude,deg	Zero-mean Gaussian	0.3202
Velocity,ft/s	Zero-mean Gaussian	100
Flight-path angle,deg	Zero-mean Gaussian	0.1084
Heading angle,deg	Zero-mean Gaussian	0.0973
Vehicle's mass,slug	Uniform	$\pm 5\%$

Simulation results show that most of the cases can successfully converge to the optimal solution and it is not sensitive with respect to the random initialization. The time histories of 100 dispersed trajectories in terms of the state variables obtained using the proposed hybrid optimal control algorithm are plotted in Fig. 11. Correspondingly, Fig. 12 shows the Hamiltonian profile in order to illustrate the optimality of the calculated solutions. As can be seen from Fig. 11, all the trajectories can be accepted as feasible solutions. Moreover, based on the time history of the Hamiltonian function demonstrated in Fig. 12, the optimality of the dispersion simulation can be guaranteed.

A comparative study was made in order to compare the convergence ability and stability of the proposed hybrid solver with other typical optimal control solvers, such as GPOPS (e.g., orthogonal collocation method) and ICLOCS (direct collocation or multiple shooting method). By setting the maximum number of Newton iterations as 3000, the convergence results are tabulated in Table II. The table summarizes the times of optimal solution found, infeasible point detected and maximum iterations exceeded for different solvers. It is worth noting that the term "infeasible point detected" in Table II means that the algorithm is stuck at a locally infeasible point.

As can be seen from Table II based on the same dispersion model, the proposed hybrid solver can perform a higher probability in terms of finding optimal solutions compared with other typical strategies. A comparison between the algorithm with and without the layer 1 was also performed. The results show that the use of the layer 1 can effectively improve the convergence ability of the proposed hybrid algorithm. Moreover, the number of Newton iterations required for the proposed method is also smaller than its counterparts. The results not only confirm that the design philosophy of the proposed solver can have positive influences in terms of improving the convergence ability and convergence speed but also indicate that the method designed in this paper can have a better performance over other typical optimal control methods.

Remark 6: It is worth noting that the stochastic optimization approach used in this paper can be a feasible way to generate skip entry trajectories. This can be seen from the reference trajectories shown in Figs. 2–4. Clearly, all the obtained solutions can be accepted as feasible solutions. If there is no accuracy requirement, then the initial guess can be accept as a near-optimal solution. In addition, when the nonlinearity of the cost functions or path constraints become higher, which means it is difficult to calculate the gradient

TABLE II
CONVERGENCE RESULTS FOR DIFFERENT OPTIMAL CONTROL SOLVERS

Different methods	Optimal solution found	Infeasible point detected	Maximum iterations exceeded	Successful rate (%)
Proposed hybrid strategy (With layer 1)	907	24	69	90.7
Proposed hybrid strategy (Without layer 1)	811	33	56	81.1
GPOPS (Orthogonal collocation)	798	53	149	79.8
ICLOCS (Direct collocation)	663	69	268	66.3
ICLOCS (Multiple shooting)	691	77	232	69.1

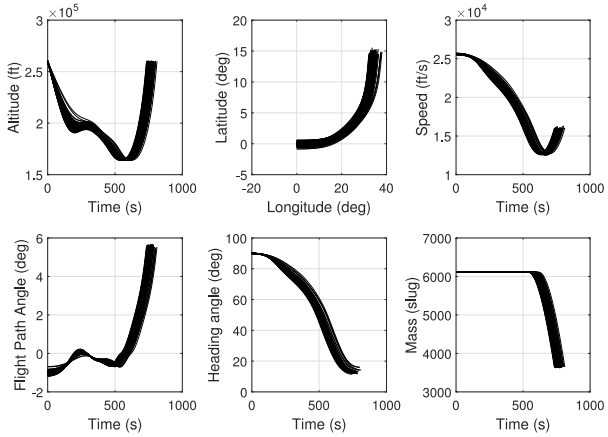

Fig. 11. One hundred dispersed trajectories for the state variables.

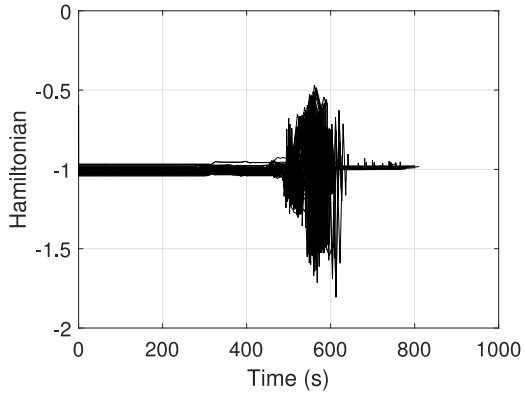

Fig. 12. One hundred dispersed trajectories for the Hamiltonian function.

information for gradient techniques, the stochastic method can also be an efficient way to generate a high quality reference trajectory.

Remark 7: It should be mentioned that the hybrid solver developed in this paper can be easily extended and implemented in other control optimization problems. For instance, recent research [34] has demonstrated the capability to utilize optimal control-based solver in dealing with agent/robot obstacle-avoidance path planning problems [35]. If the proposed solver is applied to solve such a problem, some adjustments should be made in the optimization process. First, different obstacles should be reformulated as a series of path constraints entailed in the optimization model. Besides, since the nonlinearity of the dynamics and constraints in these problems is usually high, the problem tends to be more sensitive with respect to the optimization parameters. Therefore, the proposed solver should start with a relatively small temporal set.

VI. CONCLUSION

In this paper, a three-layer-hybrid optimal control solver was constructed and applied to solve the SMV trajectory optimization problem. In order to effectively evaluate a reference trajectory, an initial guess generator using V-based adaptive differential evolution algorithm was applied. Since it is hard to use a large temporal set for stochastic-based algorithms, a specific discretization scheme is implemented to tackle this problem. In addition, a new gradient-based algorithm is used as the inner solver, thereby allowing the designer more flexibility to control the optimization process. Comparative simulations show that the proposed method can have better performance in terms of convergence ability and stability than other typical optimal control solvers. Moreover, it was verified that the obtained solution can satisfy all the optimality conditions, which illustrates the effectiveness of the proposed solver.

Our follow-up research will focus on extending the proposed design such that it can be applied in handling stochastic trajectory planning problems. In addition, it would be also worthwhile to use the proposed method for other path planning applications, especially in the multiagent path planning with collision avoidance.

REFERENCES

[1] C. Lian, X. Xu, H. Chen, and H. He, "Near-optimal tracking control of mobile robots via receding-horizon dual heuristic programming," *IEEE Trans. Cybern.*, vol. 46, no. 11, pp. 2484–2496, Nov. 2016.

[2] R. Ruiz-Cruz, E. N. Sanchez, F. Ornelas-Tellez, A. G. Loukianov, and R. G. Harley, "Particle swarm optimization for discrete-time inverse optimal control of a doubly fed induction generator," *IEEE Trans. Cybern.*, vol. 43, no. 6, pp. 1698–1709, Dec. 2013.

[3] M. F. Corapsiz and K. Erenturk, "Trajectory tracking control and contouring performance of three-dimensional CNC," *IEEE Trans. Ind. Electron.*, vol. 63, no. 4, pp. 2212–2220, Apr. 2016.

[4] Q. Wei, D. Liu, and H. Lin, "Value iteration adaptive dynamic programming for optimal control of discrete-time nonlinear systems," *IEEE Trans. Cybern.*, vol. 46, no. 3, pp. 840–853, Mar. 2016.

[5] J. T. Betts, *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*. Cambridge, U.K.: Cambridge Univ. Press, 2009.

[6] B. A. Conway, "A survey of methods available for the numerical optimization of continuous dynamic systems," *J. Optim. Theory Appl.*, vol. 152, no. 2, pp. 271–306, 2012.

[7] D. A. Benson, G. T. Huntington, T. P. Thorvaldsen, and A. V. Rao, "Direct trajectory optimization and costate estimation via an orthogonal collocation method," *J. Guid. Control Dyn.*, vol. 29, no. 6, pp. 1435–1440, 2006.

[8] W. W. Hager, "Runge–Kutta methods in optimal control and the transformed adjoint system," *Numerische Mathematik*, vol. 87, no. 2, pp. 247–282, 2000.

[9] P. Williams, "Jacobi pseudospectral method for solving optimal control problems," *J. Guid. Control Dyn.*, vol. 27, no. 2, pp. 293–297, 2004.

[10] O. A. Yakimenko, "Direct method for rapid prototyping of near-optimal aircraft trajectories," *J. Guid. Control Dyn.*, vol. 23, no. 5, pp. 865–875, 2000.

[11] T. Albin, D. Ritter, N. Liberda, R. Quirynen, and M. Diehl, "In-vehicle realization of nonlinear MPC for gasoline two-stage turbocharging airpath control," *IEEE Trans. Control Syst. Technol.*, to be published.

[12] R. Genest and J. V. Ringwood, "Receding horizon pseudospectral control for energy maximization with application to wave energy devices," *IEEE Trans. Control Syst. Technol.*, vol. 25, no. 1, pp. 29–38, Jan. 2017.

[13] D. Christopher, H. William, and R. Anil, "An improved adaptive hp algorithm using pseudospectral methods for optimal control," in *Proc. Guid. Navig. Control Co Located Conf.*, 2010, pp. 1–14.

[14] M. Heinkenschloss and D. Ridzal, "A matrix-free trust-region SQP method for equality constrained optimization," *SIAM J. Optim.*, vol. 24, no. 3, pp. 1507–1541, 2014.

[15] J. Laurent-Varin, F. Bonnans, N. Bérend, M. Haddou, and C. Talbot, "Interior-point approach to trajectory optimization," *J. Guid. Control Dyn.*, vol. 30, no. 5, pp. 1228–1238, 2007.

[16] B. Tian, W. Fan, R. Su, and Q. Zong, "Real-time trajectory and attitude coordination control for reusable launch vehicle in reentry phase," *IEEE Trans. Ind. Electron.*, vol. 62, no. 3, pp. 1639–1650, Mar. 2015.

[17] N. Yokoyama and S. Suzuki, "Modified genetic algorithm for constrained trajectory optimization," *J. Guid. Control Dyn.*, vol. 28, no. 1, pp. 139–144, 2005.

[18] J. A. Englander and B. A. Conway, "Automated solution of the low-thrust interplanetary trajectory problem," *J. Guid. Control Dyn.*, vol. 40, no. 1, pp. 15–27, 2017.

[19] J. J. Kim and J.-J. Lee, "Trajectory optimization with particle swarm optimization for manipulator motion planning," *IEEE Trans. Ind. Informat.*, vol. 11, no. 3, pp. 620–631, Jun. 2015.

[20] P. C. Roy, M. M. Islam, K. Murase, and X. Yao, "Evolutionary path control strategy for solving many-objective optimization problem," *IEEE Trans. Cybern.*, vol. 45, no. 4, pp. 702–715, Apr. 2015.

[21] R. Chai, A. Savvaris, and A. Tsourdos, "Violation learning differential evolution-based hp-adaptive pseudospectral method for trajectory optimization of space maneuver vehicle," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 53, no. 4, pp. 2031–2044, Aug. 2017.

[22] R. Chai, A. Savvaris, and A. Tsourdos, "Fuzzy physical programming for space manoeuvre vehicles trajectory optimization based on hp-adaptive pseudospectral method," *Acta Astronaut.*, vol. 123, pp. 62–70, Jun./Jul. 2016.

[23] S. M. Elsayed, R. A. Sarker, and D. L. Essam, "An improved self-adaptive differential evolution algorithm for optimization problems," *IEEE Trans. Ind. Informat.*, vol. 9, no. 1, pp. 89–99, Feb. 2013.

[24] D. Liu, X. Yang, D. Wang, and Q. Wei, "Reinforcement-learning-based robust controller design for continuous-time uncertain nonlinear systems subject to input constraints," *IEEE Trans. Cybern.*, vol. 45, no. 7, pp. 1372–1385, Jul. 2015.

[25] Y.-J. Gong *et al.*, "Genetic learning particle swarm optimization," *IEEE Trans. Cybern.*, vol. 46, no. 10, pp. 2277–2290, Oct. 2016.

[26] W. W. Hager, H. Hou, and A. V. Rao, "Convergence rate for a Radau collocation method applied to unconstrained optimal control," *arXiv.org/abs/1508.03783*, pp. 1–24, Sep. 2015.

[27] W. W. Hager, H. Hou, and A. V. Rao, "Convergence rate for a gauss collocation method applied to unconstrained optimal control," *J. Optim. Theory Appl.*, vol. 169, no. 3, pp. 801–824, 2016.

[28] R. Chai, A. Savvaris, A. Tsourdos, S. Chai, and Y. Xia, "Improved gradient-based algorithm for solving aeroassisted vehicle trajectory optimization problems," *J. Guid. Control Dyn.*, vol. 40, no. 8, pp. 2093–2101, 2017.

[29] J. T. Betts and W. P. Huffman, "Mesh refinement in direct transcription methods for optimal control," *Optimal Control Appl. Methods*, vol. 19, no. 1, pp. 1–21, 1998.

[30] F. Liu, W. W. Hager, and A. V. Rao, "Adaptive mesh refinement method for optimal control using nonsmoothness detection and mesh size reduction," *J. Frankl. Inst.*, vol. 352, no. 10, pp. 4081–4106, 2015.

[31] B. Kevin, R. Michael, and D. David, "Optimal nonlinear feedback guidance for reentry vehicles," in *Proc. Guid. Navig. Control Co Located Conf.*, 2006, pp. 1–20, doi: 10.2514/6.2006-6074.

[32] M. Ergezer and D. Simon, "Mathematical and experimental analyses of oppositional algorithms," *IEEE Trans. Cybern.*, vol. 44, no. 11, pp. 2178–2189, Nov. 2014.

[33] Y. Xia, F. Pu, S. Li, and Y. Gao, "Lateral path tracking control of autonomous land vehicle based on ADRC and differential flatness," *IEEE Trans. Ind. Electron.*, vol. 63, no. 5, pp. 3091–3099, May 2016.

[34] A. J. Häusler, A. Saccon, A. P. Aguiar, J. Hauser, and A. M. Pascoal, "Energy-optimal motion planning for multiple robotic vehicles with collision avoidance," *IEEE Trans. Control Syst. Technol.*, vol. 24, no. 3, pp. 867–883, May 2016.

[35] Z. Chen and H.-T. Zhang, "A minimal control multiagent for collision avoidance and velocity alignment," *IEEE Trans. Cybern.*, vol. 47, no. 8, pp. 2185–2192, Aug. 2017.

Runqi Chai (S'15) was born in Beijing, China, in 1993. He is currently pursuing the Ph.D. degree in aerospace engineering with Cranfield University, Cranfield, U.K.

His current research interests include trajectory optimization, guidance, and control.

Al Savvaris (M'08) received the M.Eng. degree in aerospace systems engineering from the University of Hertfordshire, Hertfordshire, U.K., in 1998 and the Ph.D. degree in radiowave propagation and system design from the University of South Wales, Pontypridd, U.K., in 2004.

He is a Reader with the Centre for Cyber-Physical Systems, Cranfield University, Cranfield, U.K. He established the Autonomous Vehicle Dynamics and Control M.Sc. course and the COMAC training programme at Cranfield. His current research interests include systems integration, hybrid energy management, communication systems, embedded systems, guidance, and control. He is currently researching on Innovate U.K. Funded AirStart and USMOOTH Projects. In the past, he researched on the FLAVIIR and ASTRAEA UAS Projects, developing new technologies for unmanned systems, researching on hardware and system integration. He has published over 100 peer-reviewed journal and conference papers.

Dr. Savvaris participated in FP6, in the scope of the FLYSAFE Project, researching on next generation integrated safety systems. He was a member of the Autonomous Systems National Technical Committee, EPSRC College Review Member, and a Reviewer on several international publications, including IMechE and IEEE.

Antonios Tsourdos (M'99) received the M.Eng. degree in electronic, control and systems engineering from the University of Sheffield, Sheffield, U.K., in 1995, the M.Sc. degree in systems engineering from Cardiff University, Cardiff, U.K., in 1996, and the Ph.D. degree in nonlinear robust missile autopilot design and analysis from Cranfield University, Cranfield, U.K. in 1999.

He is currently a Professor of control systems with Cranfield University, where he was the Appointed Head of the Autonomous Systems Group in 2007. He has published over 100 peer-reviewed journal and conference papers.

Prof. Tsourdos was a member of the Team Stellar, the winning team for the U.K. MoD Grand Challenge in 2008 and the IET Innovation Award (Category Team, 2009). He is an editorial board member on several international publications, including IMechE and IEEE. He is a member of the IFAC Technical Committee on Intelligent Autonomous Vehicles, the IET Executive Team on Robotics and Automation, and the ATI Autonomous Systems National Technical Committee. He was also involved in the SEAS DTC on Autonomous Systems Verifications. He has also been engaged in research on guidance and control for single and multiple vehicles as well as verifiable autonomy of autonomous systems and lately dealing with the newly important subjects of integrated system health management and cyber-physical systems.

Senchun Chai was born in Beijing, China. He received the Ph.D. degree with the School of Electronics, University of Glamorgan, Pontypridd, U.K., in 2007.

He was a Post-Doctoral Fellow with the School of Electronics, University of Glamorgan in 2009, where he joined the School of Automation in 2010, and is currently an Associate Professor in control. He was a Researcher with Cranfield University, Cranfield, U.K., from 2009 to 2010, and a Visiting Scholar with the University of Illinois at Urbana–Champaign, Champaign, IL, USA, in 2010 for four months. His current research interests include design of unmanned aerial vehicles, wireless sensor network control, networked control systems, and multiagent control systems.

Yuanqing Xia (M'15–SM'16) was born in Anhui, China, in 1971. He received the B.S. degree from the Department of Mathematics, Chuzhou University, Chuzhou, China, in 1991, the M.S. degree in fundamental mathematics from Anhui University, Wuhu, China, in 1998, and the Ph.D. degree in control theory and control engineering from the Beijing University of Aeronautics and Astronautics, Beijing, China, in 2001.

He has published eight monographs with Springer and Wiley, and over 200 papers in journals. His current research interests include networked control systems, robust control and signal processing, active disturbance rejection control, and flight control.

Prof. Xia was a recipient of the Second Award of the Beijing Municipal Science and Technology (No. 1) in 2010, the Second National Award for Science and Technology (No. 2) in 2011, and the Second Natural Science Award of The Ministry of Education (No. 1) in 2012. He is a Deputy Editor of the *Journal of the Beijing Institute of Technology*, an Associate Editor of *Acta Automatica Sinica*, *Control Theory and Applications*, the *International Journal of Innovative Computing*, *Information and Control*, and the *International Journal of Automation and Computing*.