Received October 3, 2021, accepted October 13, 2021, date of publication October 15, 2021, date of current version October 25, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3120840

Successive Convexification for Online Ascent Trajectory Optimization

CHENG HU, XIBIN BAI, SHIFENG ZHANG, AND HUABO YANG
College of Aerospace Science and Engineering, National University of Defense Technology, Changsha 410073, China
Corresponding author: Shifeng Zhang (zhang_shifeng@hotmail.com)

ABSTRACT In this paper, a method based on the successive convexification is proposed to solve the ascent trajectory optimization problem, the algorithm converges to the optimal solution quickly even if the initial guess is coarse. A three-dimensional motion is formulated with complex aerodynamics and terminal constraints. Based on the modified aerodynamic coefficients, the new auxiliary control variables are designed to deal with the complex aerodynamics and non-smooth of control variables in the discrete optimization problem. The inner nonconvex constraints between the new control are relaxed to be convex without loss. The artificial infeasibility and unboundedness caused by linearization are tackled by the virtual controls and soft constraint for trust region in the successive convexification. The good convergence of the proposed method is illustrated by the iterative solutions of the ascent trajectory optimization problem for a small guided rocket, the accuracy is verified by the comparison with the optimal solution given by the typical optimal control solvers, and the feasibility and stability are demonstrated by optimal solutions of the ascent trajectory optimization problems under different missions and dispersed conditions. These excellent performances validated by the adequate simulations indicate that the proposed algorithm can be implemented online.

INDEX TERMS Convex optimization, ascent trajectory optimization, successive convexification, online trajectory optimization, complex nonlinear aerodynamic force.

I. INTRODUCTION

The ascent trajectory optimization problem has been developed over decades and attracting wide interests and research attention, it is of great significance for the rockets or vehicles to reconstruct the trajectory adaptively when the mission changes or the non-fatal fault occurs during the flight [1]. Generally, the methods for ascent trajectory optimization problems could be categorized as direct method and indirect method, based on whether Hamiltonian first-order necessary condition is satisfied [2], [3]. The former transforms the continuous-time optimization problem into a nonlinear programming (NLP) problem by means of the discretization, in which the discrete state and control are both regarded as decision variables [4]; while the latter makes full use of the first-order necessary conditions to convert the optimization problem into a two-point-boundary-value-problem (TPBVP) of the state and costate variables [5].

The shooting technique and multiple shooting technique [6], [7] were firstly used to solve the TPBVPs, and especially, they were effective in the vacuum ascent trajectory optimization problem, of which the optimal solution was the explicit expression about the initial variable [8]. The convergence of the shooting technique depended on the initial guesses heavily, while the initial value of the costate was difficult to be selected due to its inexplicit physical meaning. Besides, the TPBVPs were converted into the root-finding problems for the system with a certain number of nonlinear algebraic equations by the means of the finite difference approach [9]–[11], and the Newton iteration method was utilized to work out the roots. Based on the previous works, Huang *et al.* [12] presented a mixed variable variational algorithm to solve the ascent trajectory optimization problem under the multiple path and terminal constraints. However, the large-scale Jacobian matrices were inevitable in these indirect algorithms, which would cost much computational time and memory when replaced by the finite difference approximation, or need complicated mathematical derivations when solved analytically.

With the development of the on-board computer, the direct methods showed enormous potential for real-time optimal guidance [13]. The different pseudospectral methods were

The associate editor coordinating the review of this manuscript and approving it for publication was Shanying Zhu.

introduced to solve the trajectory optimization problems successfully, including Chebyshev pseudospectral [14], [15] and Radau pseudospectral [16]. Particularly, Gauss pseudospectral method [13], [17] was developed to be the sophisticated one with the application of a general pseudospectral optimal control software (GPOPS). Furthermore, the direct methods were combined with the intelligent algorithms to solve the trajectory optimization problems, Jiang *et al*. [18] proposed a hybrid optimization strategy by taking the advantages of particle swarm optimization (PSO) and Gauss pseudospectral method, Chai *et al*. [19], [20] utilized the "discretization + optimization" strategies to solve the reentry trajectory planning, such as violation learning deferential evolution-based hp-adaptive pseudo-spectral method and multiple-shooting discretization technique with the newest NSGA-III optimization algorithm, besides, a violation learning deferential evolution method was designed to generate the appropriate initial guess. These works illustrated that pseudospectral methods were effective to discrete the trajectory optimization problems. However, the computational time for the above direct methods fluctuated dramatically without known boundaries; and what's more, the convergence of the optimal or feasible solution could not be guaranteed after a certain number of iterations. These uncertainties hindered the online application of the above direct methods.

As another direct method, the convex programming method has been introduced to solve the ascent trajectory optimization problem in recent decades because of the global optimal solution and good convergence [21]. Sun *et al*. [22] realized the optimization of two-dimensional ascent trajectory via the sequential convex programming (SCP) with iterative narrowing trust region. To deal with the nonconvexity of the aerodynamics, the auxiliary control variables were introduced, and then the relaxation technique was used to relax the nonconvex constraints on the new control [23]–[25]. Açıkmeşe *et al*. [26] and Carson *et al*. [27] proposed the lossless convexification methods to handle the nonconvex thrust boundary and pointing constraints for the pinpoint and precision landing problem. Besides, the SCP method was also employed to solve the reentry trajectory optimization problem [28]–[32]. When all of the above convex programming methods were implemented, the sequential convex subproblems were solved iteratively until the optimal solutions were obtained. However, the convergence of the sequential iterations was not proved fully, thus they might suffer the risk of non-convergence. A successive convexification (SCvx) was proposed to solve non-convex optimal control problems by Mao *et al*. [33], [34], and proof of the convergence properties was presented in details.

In this study, the SCvx is employed to solve the ascent trajectory optimization problem. The new auxiliary control variables based on the fitting aerodynamic coefficients are designed to deal with the nonconvexity of the complex aerodynamics with the aid of an exact relaxation. The artificial infeasibility and unboundedness which may be caused by the linearization in the SCP methods are tackled by the virtual controls and soft constraint for trust region in the successive convexification, respectively. The performance of simulation illustrates that the proposed method solves the ascent trajectory optimization problem rapidly and shows potential for real-time optimal guidance. Compared with the previous works on the SCP method used in the trajectory optimization, the primary contributions of the paper are drawn as three points: 1) the 3 degree of freedom (3-Dof) formulation with accurate and tractable aerodynamics is feasible to different ascent trajectory optimization problems; 2) the proposed new control facilitates the convergence to the optimal solution; 3) the soft constraint of the trust region avoids the non-convergence of the sequential iterations.

The outline of the rest of this article is structured as follows: In the upcoming section, the problem of ascent trajectory optimization is established with the new control. Then, the SCvx algorithm is expanded detailly to solve the ascent trajectory optimization problem in section III. Next, in section IV, adequate simulations and comparisons are performed and discussed to illustrate the accuracy, feasibility and stability of the proposed method. At last, the conclusions are drawn in section V.

II. PROBLEM FORMULATION

The ascent trajectory optimization problem with the new control will be formulated in this section. Firstly, a 3-Dof formulation is presented in subsection II.A for the ascent trajectory optimization problem. Then, based on the analysis of the original problem, the new auxiliary control variables combined with relaxation technology are proposed to deal with the complex and nonconvex aerodynamics in subsection II.B. Finally, the new optimization problem for ascent trajectory is expressed in subsection II.C.

A. ASCENT TRAJECTORY OPTIMIZATION PROBLEM

Under the assumption of a small attack angle and the flat Earth, the dimensionless three-dimensional (3-D) motion of the ascent trajectory for the bank to turn (BTT) vehicle can be expressed as follows.

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u}, t): \\ = \begin{cases} \dfrac{\mathrm{d}x}{\mathrm{d}t} = v\cos\theta\cos\sigma \\ \dfrac{\mathrm{d}y}{\mathrm{d}t} = v\sin\theta \\ \dfrac{\mathrm{d}z}{\mathrm{d}t} = v\cos\theta\sin\sigma \\ \dfrac{\mathrm{d}v}{\mathrm{d}t} = \dfrac{P}{mg_0} - \dfrac{r_E\rho v^2 S_{ref}}{2m}C_D\left(Ma,\alpha\right) - \sin\theta \\ \dfrac{\mathrm{d}\theta}{\mathrm{d}t} = \dfrac{P\alpha}{mg_0 v} + \dfrac{r_E\rho\tilde{v}S_{ref}}{2m}C_L\left(Ma,\alpha\right)\cos\upsilon - \dfrac{\cos\theta}{v} \\ \dfrac{\mathrm{d}\sigma}{\mathrm{d}t} = \dfrac{r_E\rho v S_{ref}C_L}{2m\cos\theta}C_L\left(Ma,\alpha\right)\sin\upsilon \end{cases} \tag{1}$$

where x, y and z are the dimensionless positions which are normalized by the reference length r_E, v is the dimensionless velocity normalized by the reference velocity $v_E = \sqrt{r_E g_0}$,

t is the dimension-less time normalized by the reference time $t_E = \sqrt{r_E/g_0}$, g_0 is the average gravitational acceleration of Earth, θ and σ are the flight path angle and heading angle, respectively. The state of the system is denoted as $\boldsymbol{x} = [x, y, z, v, \theta, \sigma]^{\mathrm{T}}$.

P and m are the thrust and mass of the vehicle, respectively, which can be obtained from the test-run experiment. ρ is the density of the atmosphere, of which the approximate value is given as Eq.(2), where $h_s = 7110$m, $\rho_{\mathrm{SL}} = 1.225\mathrm{kg/m}^3$.

$$\rho = \rho_{SL} \exp\left(-\frac{yr_E}{h_s}\right) \tag{2}$$

S_{ref} is the reference area of the vehicle; C_D and C_L denote the coefficients of the aerodynamic drag and lift. In general, they are assumed as the functions of Mach number Ma and attack angle α. The attack angle α and bank angle υ are the control variables in this ascent trajectory optimization problem, i.e. $\boldsymbol{u} = [\alpha, \upsilon]^{\mathrm{T}}$

Notice that θ and σ in Eq.(1) are dimensionless, their dimensionless derivatives only depend on the reference time. According to the above definitions, we can know that all of the reference parameters are determined by the reference length r_E. So, it is possible to set all the magnitudes of the dimensionless states equal or near by choosing the appropriate reference length, which could facilitate the numerical calculation.

Generally, the minimum-time problem with free terminal time and the maximum-energy problem with fixed terminal time are the two typical problems of ascent trajectory optimization. The free terminal-time minimum-time problem can be transformed into the sequential fixed terminal-time maximum-energy problem, and the latter is easier to converge [35]. Thus, only the maximum-energy problem will be studied in this paper. When the terminal height is fixed, the maximum-energy will be equivalent to maximum-velocity, eventually, the performance index of the optimization problem in our study can be expressed as Eq.(3)

$$J = \phi(\boldsymbol{x}_f) = -v_f \tag{3}$$

The terminal state of the ascent phase should be specially designed when the whole trajectory is taken into consideration, because it effects greatly on the following trajectory. Without loss of generality, the boundary constraints of the fixed interval $[t_0, t_f]$ ascent trajectory optimization problem are given by Eq.(4).

$$\begin{cases} x(t_0) = x_0 \\ y(t_0) = y_0 \\ z(t_0) = z_0 \\ v(t_0) = v_0 \\ \theta(t_0) = \theta_0 \\ \sigma(t_0) = \sigma_0 \end{cases} \quad \begin{cases} x(t_f) = x_f \\ y(t_f) = y_f \\ z(t_f) = z_f \\ \theta(t_f) = \theta_f \\ \sigma(t_f) = \sigma_f \end{cases} \tag{4}$$

Under the assumption of a small angle, the attack angle is constrained by Eq.(5). According to the maneuver of the vehicle, the bank angle is constrained by Eq.(6).

$$0 \le |\alpha| \le \alpha_{\max} \tag{5}$$

$$-\pi \le \upsilon \le \pi \tag{6}$$

In summary, the original nonlinear continuous-time ascent trajectory optimization problem can be formulated as

$$\begin{aligned} \mathbf{P1}: \quad & \min J_1 = \varphi(\boldsymbol{x}_f) \\ & \text{Subject to Eqs.(1), (4)} \sim \text{(6).} \end{aligned}$$

B. NEW CONTROL

It can be known that the high nonlinearity of problem **P1** is mainly generated from the complex aerodynamics. In order to handle this, an ideal and simplified aerodynamic coefficient was proposed under the assumption of a symmetrical shape [24], [25], in which the lift coefficient C_L was proportional to the attack angle α, and the drag coefficient depended on the zero lift drag C_{D0} and the square of the attack angle α^2. Based on the simplified aerodynamic coefficients, the control variables were decoupled, and the system was transformed into an affine one. Using the SCP algorithm to solve the nonlinear affine system could avoid the nonlinearity of controls and achieve a less iterative step of the successive subproblems. However, the optimal solutions worked out by the simplified aerodynamics would be less accurate when the deviation between the actual and simplified aerodynamics could not be ignored. Besides, the derivative of bank angle was chosen as the control variable [31], [32] to obtain the smooth control in the discrete optimization problem, of which system the bank angle was transformed into the extended state. Even so, the bank angle would still suffer the small but unnecessary chatters.

In this paper, the aerodynamic coefficients are modified to be more accurate and tractable. Based on the modified aerodynamics, the nonlinear terms of the control are regarded as the new auxiliary control variables, which are designed to reduce the influence of nonlinear aerodynamics on the system. The inner nonconvex constraints among the new controls are relaxed to be convex ones, and the active sets of the convex constraints after the relaxation are proved theoretically to be the same as the feasible sets of the original problem. Moreover, the new auxiliary controls are discovered to solve the optimization problem with smooth discrete controls effectively through the results of simulation, and this is also applied to overcome the high-frequency chatter of the discrete controls in the reentry trajectory optimization problem [29], [30].

1) AERODYNAMICS COEFFICIENTS MODIFICATION

The aerodynamic coefficients are modified as the quadratic functions of attack angle and Mach number, which are expressed as Eq.(7), thus they can fit the actual aerodynamic data more accurately and be applied to almost all the vehicles

with different shapes.

$$\begin{cases} C_L(Ma,\alpha) = C_L^0 + C_L^\alpha\alpha + C_L^{Ma}Ma + C_L^{\alpha Ma}\alpha Ma \\ \qquad\qquad + C_L^{Ma^2}Ma^2 + C_L^{\alpha^2}\alpha^2 \\ C_D(Ma,\alpha) = C_D^0 + C_D^\alpha\alpha + C_D^{Ma}Ma + C_D^{\alpha Ma}\alpha Ma \\ \qquad\qquad + C_D^{Ma^2}Ma^2 + C_D^{\alpha^2}\alpha^2 \end{cases} \tag{7}$$

The coefficients $C_L^0, C_L^\alpha, \ldots, C_D^{\alpha^2}$ are obtained through fitting aerodynamic data from the CFD simulations or the wind tunnel experiments. Additionally, the aerodynamic function in Eq.(7) is more efficient in the numerical calculation than the interpolation of the aerodynamic data.

Furthermore, the aerodynamic coefficients (7) can be collected as the function of the attack angle α and its square α^2 without loss, which is expressed as Eq.(8).

$$\begin{cases} C_L(Ma,\alpha) = \hat{C}_L^0(Ma) + \hat{C}_L^\alpha(Ma)\alpha + \hat{C}_L^{\alpha^2}\alpha^2 \\ C_D(Ma,\alpha) = \hat{C}_D^0(Ma) + \hat{C}_D^\alpha(Ma)\alpha + \hat{C}_D^{\alpha^2}\alpha^2 \end{cases} \tag{8}$$

where

$$\begin{cases} \hat{C}_L^0(Ma) = C_L^0 + C_L^{Ma}Ma + C_L^{Ma^2}Ma^2 \\ \hat{C}_L^\alpha(Ma) = C_L^\alpha + C_L^{\alpha Ma}Ma \\ \hat{C}_L^{\alpha^2} = C_L^{\alpha^2} \end{cases} \tag{9}$$

$$\begin{cases} \hat{C}_D^0(Ma) = C_D^0 + C_D^{Ma}Ma + C_D^{Ma^2}Ma^2 \\ \hat{C}_D^\alpha(Ma) = C_D^\alpha + C_D^{\alpha Ma}Ma \\ \hat{C}_D^{\alpha^2} = C_D^{\alpha^2} \end{cases} \tag{10}$$

2) CHOICE OF NEW CONTROL

According to the above modified aerodynamics coefficients, we define the new control vector as $\boldsymbol{u} = [u_1, u_2, u_3, u_4]^{\mathrm{T}}$, where the components are

$$u_1 = \alpha, \quad u_2 = \alpha^2, \quad u_3 = \cos\upsilon, \quad u_4 = \sin\upsilon \tag{11}$$

Substituting the modified aerodynamics coefficients and new control $\boldsymbol{u}$ into Eq.(1), the dynamics can be rewritten as

$$\dot{\boldsymbol{x}} = F(\boldsymbol{x},\boldsymbol{u},t): = \begin{cases} \dfrac{\mathrm{d}x}{\mathrm{d}t} = v\cos\theta\cos\sigma \\ \dfrac{\mathrm{d}y}{\mathrm{d}t} = v\sin\theta \\ \dfrac{\mathrm{d}z}{\mathrm{d}t} = v\cos\theta\sin\sigma \\ \dfrac{\mathrm{d}v}{\mathrm{d}t} = \dfrac{P}{mg_0} - \dfrac{r_E\rho v^2 S_{ref}}{2m} \\ \qquad \times\left(\hat{C}_D^0 + \hat{C}_D^\alpha u_1 + \hat{C}_D^{\alpha^2}u_2\right) - \sin\theta \\ \dfrac{\mathrm{d}\theta}{\mathrm{d}t} = \dfrac{Pu_1}{mg_0 v} + \dfrac{r_E\rho v S_{ref}u_3}{2m} \\ \qquad \times\left(\hat{C}_L^0 + \hat{C}_L^\alpha u_1 + \hat{C}_L^{\alpha^2}u_2\right) - \dfrac{\cos\theta}{v} \\ \dfrac{\mathrm{d}\sigma}{\mathrm{d}t} = \dfrac{r_E\rho v S_{ref}u_4}{2m\cos\theta}\left(\hat{C}_L^0 + \hat{C}_L^\alpha u_1 + \hat{C}_L^{\alpha^2}u_2\right) \end{cases} \tag{12}$$

The range of the angle-of-attack is $0 \le |\alpha| \le \alpha_{\max}$, it could be expressed as Eq.(13), where $u_2^- = 0$ and $u_2^+ = \alpha_{\max}^2$. And the range of the bank angle is $-\pi \le \upsilon \le \pi$, which is equal to the range of the anti-trigonometric function on the interval $[-1, 1]$, so there is no need for extra condition to limit the range of new control variables u_3 and u_4 when they are constrained by Eq.(14).

$$\begin{cases} g_1 = u_2^- - u_2 \le 0 \\ g_2 = u_2 - u_2^+ \le 0 \end{cases} \tag{13}$$

$$h_1 = u_3^2 + u_4^2 - 1 = 0 \tag{14}$$

As u_1 and u_2 are not independent variables, their inner relationship can be restricted by Eq.(15).

$$h_2 = u_1^2 - u_2 = 0 \tag{15}$$

Constrained by Eqs.(13)∼(15), the feasible set of the new control is the same as that of the original control, however, the constraints (14) and (15) are nonconvex. Although they can be transformed into convex constraints by the approximate Taylor expansion in the traditional SCP methods, the nonconvex constraints will result in more successive iterations. Thus, a relaxation technique is employed to convexify these nonconvex constraints, it relaxes the feasible sets of constraints (14) and (15) which locate only at the curved boundaries as Eq.(16).

$$\begin{cases} g_3 = u_1^2 - u_2 \le 0 \\ g_4 = u_3^2 + u_4^2 - 1 \le 0 \end{cases} \tag{16}$$

C. NEW ASCENT TRAJECTORY OPTIMIZATION PROBLEM

Modified by the new control and its relaxed constraints, the new ascent trajectory optimization problem **P2** is summarized as

$$\begin{aligned} \mathbf{P2}: \quad & \min J_2 = \phi(\boldsymbol{x}_f) \\ & \text{Subject to Eqs. (12), (4), (13) and (16)} \end{aligned}$$

As a result of the relaxation, the feasible set of the original problem **P1** is a subset of the new problem **P2**, thus it can be drawn that the minimum objective of problem **P2** would be even lower than that of problem **P1**, i.e. $J_2^* \le J_1^*$. Now, it is critical to prove that $J_2^* = J_1^*$ if we want to make full use of the relaxation.

Let the optimal solution of problem **P2** be denoted as $\{\boldsymbol{x}^*(t), \boldsymbol{u}^*(t)\}$, then $\{\boldsymbol{x}^*(t), \boldsymbol{u}^*(t)\}$ will also be the optimal solution of problem **P1** if the conditions $u_1^{*2} = u_2^*$ and $u_3^{*2} + u_4^{*2} = 1$ are satisfied, which indicates that the optimal solution of problem **P2** lies only at the boundaries of the relaxed constraints in Eq.(16).

Proposition: If the optimal solution of problem **P2** is given as $\{\boldsymbol{x}^*(t), \boldsymbol{u}^*(t)\}$, the constraints g_3 and g_4 of Eq.(16) will be active during the interval $[t_0, t_f]$, i.e. $u_1^{*2} = u_2^*$ and $u_3^{*2} + u_4^{*2} = 1$.

Proof: See the Appendix

III. SUCCESSIVE CONVEXIFICATION

The SCvx algorithm described in this section consists of linearization, discretization, virtual control and trust

region constraint. The latter two are utilized to address artificial infeasibility and unboundedness introduced by linearization. Eventually, the proposed algorithm solves the sequential subproblems iteratively until the convergence condition is satisfied.

A. LINEARIZATION

The linearization is used to transform the fixed terminal-time nonlinear continuous-time problem **P2** into a linear one. By converting the nonlinear dynamics to its first-order linear items approximately, linearization produces a linear and convex subproblem.

Based on the first-order Taylor expansion, the nonlinear dynamic (12) is approximated as the linear continuous-time one which is denoted by Eq.(17). The reference trajectory is denoted by $\bar{z} = [\bar{\boldsymbol{x}}; \bar{\boldsymbol{u}}; t]$, which could be obtained by the previous solution; the coefficient matrices of Eq.(17) are given in Eqs.(18) ∼(20).

$$\dot{\boldsymbol{x}}(t) \approx A(t)\boldsymbol{x}(t) + B(t)\boldsymbol{u}(t) + w(t) \tag{17}$$

$$A(t) := \left.\frac{\partial f(\boldsymbol{x},\boldsymbol{u},t)}{\partial \boldsymbol{x}}\right|_{\bar{z}} = \begin{bmatrix} 0 & 0 & 0 & a_{14} & a_{15} & a_{16} \\ 0 & 0 & 0 & a_{24} & a_{25} & 0 \\ 0 & 0 & 0 & a_{34} & a_{35} & a_{36} \\ 0 & a_{42} & 0 & a_{44} & a_{45} & 0 \\ 0 & a_{52} & 0 & a_{54} & a_{55} & 0 \\ 0 & a_{62} & 0 & a_{64} & a_{65} & 0 \end{bmatrix} \tag{18}$$

$$B(t) := \left.\frac{\partial f(\boldsymbol{x},\boldsymbol{u},t)}{\partial \boldsymbol{u}}\right|_{\bar{z}} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ b_{41} & b_{42} & 0 & 0 \\ b_{51} & b_{52} & b_{53} & 0 \\ b_{61} & b_{62} & 0 & b_{64} \end{bmatrix} \tag{19}$$

$$\boldsymbol{w}(t) := f(\bar{\boldsymbol{x}},\bar{\boldsymbol{u}},t) - A(t)\bar{\boldsymbol{x}}(t) - B(t)\bar{\boldsymbol{u}}(t) \tag{20}$$

where the non-zero elements of the matrices A and B are

$$\begin{cases} a_{14} = \cos\theta\cos\sigma \\ a_{15} = -v\sin\theta\cos\sigma \\ a_{16} = -v\cos\theta\sin\sigma \end{cases} \tag{21}$$

$$\begin{cases} a_{24} = \sin\theta \\ a_{25} = v\cos\theta \end{cases} \tag{22}$$

$$\begin{cases} a_{34} = \cos\theta\sin\sigma \\ a_{35} = -v\sin\theta\sin\sigma \\ a_{36} = v\cos\theta\cos\sigma \end{cases} \tag{23}$$

$$\begin{cases} a_{42} = \dfrac{r_E^2\rho v^2 S_{ref}}{2mh_s}\left(\hat{C}_D^0 + \hat{C}_D^\alpha u_1 + \hat{C}_D^{\alpha^2} u_2\right) \\ a_{44} = -\dfrac{r_E\rho v S_{ref}}{m}\left(\hat{C}_D^0 + \hat{C}_D^\alpha u_1 + \hat{C}_D^{\alpha^2} u_2\right) \\ \qquad -\dfrac{r_E\rho v Ma S_{ref}}{2m}\left(\dfrac{\hat{C}_D^0}{\partial Ma} + \dfrac{\hat{C}_D^0}{\partial Ma}u_1\right) \\ a_{45} = -\cos\theta \end{cases} \tag{24}$$

$$\begin{cases} a_{52} = -\dfrac{r_E^2\rho v S_{ref} u_3}{2mh_s}\left(\hat{C}_L^0 + \hat{C}_L^\alpha u_1 + \hat{C}_L^{\alpha^2} u_2\right) \\ a_{54} = -\dfrac{P}{mgv^2}u_1 + \dfrac{r_E\rho S_{ref} u_3}{2m}\left(\hat{C}_L^0 + \hat{C}_L^\alpha u_1 + \hat{C}_L^{\alpha^2} u_2\right) \\ \qquad +\dfrac{r_E\rho Ma S_{ref} u_3}{2m}\left(\dfrac{\hat{C}_L^0}{\partial Ma} + \dfrac{\hat{C}_L^0}{\partial Ma}u_1\right) + \dfrac{\cos\theta}{v^2} \\ a_{55} = \dfrac{\sin\theta}{v} \end{cases} \tag{25}$$

$$\begin{cases} a_{62} = -\dfrac{r_E^2\rho v S_{ref} u_4}{2mh_s\cos\theta}\left(\hat{C}_L^0 + \hat{C}_L^\alpha u_1 + \hat{C}_L^{\alpha^2} u_2\right) \\ a_{64} = \dfrac{r_E\rho S_{ref} u_4}{2m\cos\theta}\left(\hat{C}_L^0 + \hat{C}_L^\alpha u_1 + \hat{C}_L^{\alpha^2} u_2\right) \\ \qquad +\dfrac{r_E\rho Ma S_{ref} u_4}{2m\cos\theta}\left(\dfrac{\hat{C}_L^0}{\partial Ma} + \dfrac{\hat{C}_L^0}{\partial Ma}u_1\right) \\ a_{65} = \dfrac{r_E\rho v S_{ref} C_L\sin\theta u_4}{2m\cos^2\theta}\left(\hat{C}_L^0 + \hat{C}_L^\alpha u_1 + \hat{C}_L^{\alpha^2} u_2\right) \end{cases} \tag{26}$$

$$\begin{cases} b_{41} = -\dfrac{r_E\rho v^2 S_{ref}\hat{C}_D^\alpha}{2m} \\ b_{42} = -\dfrac{r_E\rho v^2 S_{ref}\hat{C}_D^{\alpha^2}}{2m} \end{cases} \tag{27}$$

$$\begin{cases} b_{51} = \dfrac{P}{mg_0\tilde{v}} + \dfrac{r_E\rho v S_{ref}\hat{C}_L^\alpha}{2m}u_3 \\ b_{52} = \dfrac{r_E\rho v S_{ref}\hat{C}_L^{\alpha^2}}{2m}u_3 \\ b_{53} = \dfrac{r_E\rho v S_{ref}}{2m}\left(\hat{C}_L^0 + \hat{C}_L^\alpha u_1 + \hat{C}_L^{\alpha^2} u_2\right) \end{cases} \tag{28}$$

$$\begin{cases} b_{61} = \dfrac{r_E\rho v S_{ref}\hat{C}_L^\alpha}{2m\cos\theta}u_4 \\ b_{62} = \dfrac{r_E\rho v S_{ref}\hat{C}_L^{\alpha^2}}{2m\cos\theta}u_4 \\ b_{64} = \dfrac{r_E\rho v S_{ref}}{2m\cos\theta}\left(\hat{C}_L^0 + \hat{C}_L^\alpha u_1 + \hat{C}_L^{\alpha^2} u_2\right) \end{cases} \tag{29}$$

B. DISCRETIZATION

This step transforms the linear continuous-time problem obtained by linearization into a discrete-time one, and then converts the optimization problem to an NLP problem. It's important for this step to guarantee that the converged solution of the discrete-time problem corresponds to the previous continuous-time dynamics precisely.

There are different methods for discretization, such as Euler method [36], trapezoid methods [37] and pseudospectral methods [14]. According to the literatures, the first-order-hold (FOH) interpolation [38], [39] is usually utilized to discretize the continuous-time problem. Firstly, the fixed

interval $[t_0, t_f]$ of the ascent phase is divided into $K-1$ subintervals by K evenly spaced temporal nodes.

For each subinterval ($k = 1, 2, \ldots, K-1$), FOH interpolation of the control is expressed as

$$\boldsymbol{u}(t) := \tau_k^- \boldsymbol{u}_k + \tau_k^+ \boldsymbol{u}_{k+1} \quad t \in [t_k, t_{k+1}] \tag{30}$$

where $\boldsymbol{u}_k := \boldsymbol{u}(t_k)$ is the discrete-time control,

$$\tau_k^- := \frac{t_{k+1} - t}{t_{k+1} - t_k} \quad \text{and } \tau_k^+ := \frac{t - t_k}{t_{k+1} - t_k}$$

Substituting the FOH interpolation of the control into Eq.(17), the continuous-time dynamics for each subinterval is expressed as follows

$$\dot{\boldsymbol{x}}(t) = A(t)\boldsymbol{x}(t) + B(t)\tau_k^- \boldsymbol{u}_k + B(t)\tau_k^+ \boldsymbol{u}_{k+1} + \boldsymbol{w}(t) \tag{31}$$

The state transition matrix $\Phi_A(\xi, t_k)$ associated with Eq.(31) is given by

$$\Phi_A(\xi, t_k) := I_{n_x \times n_x} + \int_{t_k}^{\xi} A(\zeta)\Phi_A(\zeta, t_k)d\zeta \tag{32}$$

Let $x_k := x(t_k)$ be the discrete-time state vectors, the discrete-time dynamics for each subinterval is expressed as

$$\begin{aligned}
\boldsymbol{x}_{k+1} &:= A_k \boldsymbol{x}_k + B_k^- \boldsymbol{u}_k + B_k^+ \boldsymbol{u}_{k+1} + \boldsymbol{w}_k \\
A_k &:= \Phi_A(t_{k+1}, t_k) \\
B_k^- &:= A_k \int_{t_k}^{t_{k+1}} \Phi_A^{-1}(\xi, t_k) B(\xi)\tau_k^-(\xi)d\xi \\
B_k^+ &:= A_k \int_{t_k}^{t_{k+1}} \Phi_A^{-1}(\xi, t_k) B(\xi)\tau_k^+(\xi)d\xi \\
B_k^+ &:= A_k \int_{t_k}^{t_{k+1}} \Phi_A^{-1}(\xi, t_k) B(\xi)\tau_k^+(\xi)d\xi
\end{aligned} \tag{33}$$

C. VIRTUAL CONTROL AND TRUST REGION CONSTRAINT

An implementable convex optimization subproblem is generated through the above steps. However, the subproblem may suffer the artificial infeasibility and unboundedness [38], of which the former indicates that the subproblem is infeasible even if there exists the feasible solution of the original optimal problem, and the latter arises when the objective of the linearized optimal subproblem is allowed to be minimized or maximized infinitely. Both the artificial infeasibility and unboundedness lead to the iteration of the successive optimal subproblems stopping at an infeasible solution. In order to tackle these, the virtual control and trust region constraint are proposed to modify the traditional SCP algorithm, respectively.

1) VIRTUAL CONTROL

To address the artificial infeasibility, the virtual control term $\boldsymbol{v}_k \in \mathbb{R}^n (n = 6)$ is added to the dynamics (33)

$$\boldsymbol{x}_{k+1} := A_k \boldsymbol{x}_k + B_k^- \boldsymbol{u}_k + B_k^+ \boldsymbol{u}_{k+1} + \boldsymbol{w}_k + \boldsymbol{v}_k \tag{34}$$

With the unconstrained virtual control $\boldsymbol{v}_k$, the system will be controllable, which indicates that any state in the feasible set can be reachable in the finite time. The virtual control makes it possible for each subproblem to be solved successfully if the feasible set isn't empty, and thus the convergence process of the iterative subproblems is not terminated with an infeasible solution. This is why the virtual control can keep the successive convexification away from the artificial infeasibility.

Despite that we want to recourse to this virtual control as expected, a penalty function in Eq.(35) is employed to encourage sparsity in the vectors $\boldsymbol{v}_k$.

$$J_{vc} = \lambda \sum_{k=1}^{K-1} \|\boldsymbol{v}_k\|_1 \tag{35}$$

where $\lambda \in \mathbb{R}_{++}$ is a large weight and $\|\cdot\|_1$ is the 1-norm of a vector. It can be deduced from the penalty function of virtual control that the virtual controls are expected to be zero until the convergence, then Eq.(34) is equivalent to Eq.(33).

2) TRUST REGION CONSTRAINT

According to the above description, the artificial unboundedness can be solved directly by constraining the decision variables $\boldsymbol{z} = [\boldsymbol{x}_1^{\mathrm{T}}, \boldsymbol{u}_1^{\mathrm{T}}, \boldsymbol{x}_2^{\mathrm{T}}, \boldsymbol{u}_2^{\mathrm{T}}, \ldots, \boldsymbol{x}_K^{\mathrm{T}}, \boldsymbol{u}_K^{\mathrm{T}}]^{\mathrm{T}}$ of the successive subproblem with a trust region, which can be expressed as Eq.(36), where η is the radius of trust region and the inequality sign applies component-wise.

$$\|\boldsymbol{z} - \bar{\boldsymbol{z}}\| \leq \boldsymbol{\eta} \tag{36}$$

However, the constant trust region does not always ensure that the iterations of the successive subproblems converge to the optimal solution. A two-dimensional optimization problem is taken as example, of which the constraint is $f(\boldsymbol{x}) = 0.5\boldsymbol{x}_1^2\text{-}\boldsymbol{x}_2 = 0$,the cost is $\min\{x_2\}$, and the constant trust region is $\|\mathbf{x} - \bar{\mathbf{x}}\|_2 \leq 1$. The iteration process of the SCP algorithm with constant trust region is listed in Table 1. It can be drawn from Table 1 that the iterative solutions of the successive subproblems for the given two-dimensional optimal problem don't converge when the trust region is constant. Eventually, the solutions change between the two points of which one is located at the edge of the other's trust region.

TABLE 1. Iteration of two-dimensional optimization problem using SCP with the constant trust region.

step	x_1	x_2	step	x_1	x_2
0	0.4	0.4	5	-0.372	-0.423
1	-0.376	-0.230	6	0.621	-0.300
2	0.621	-0.304	7	-0.372	-0.423
3	-0.372	-0.424	8	0.621	-0.300
4	0.620	-0.300	9	-0.372	-0.423

To tackle the non-convergence, the narrowing trust region was proposed by Sun *et al.* [22], but the subproblem would be infeasible if the radius $\boldsymbol{\eta}$ of trust region was too small. Besides, an alternative updating rule of trust region radius was described detailly by Mao *et al.* [33], the convergence of the successive convex subproblems is analyzed and proved, but there would be several invalid iterations and computational effort may increase.

The above constraints of the trust region could be named as the hard constraints. Contrarily, a soft constraint of trust region was proposed by Szmuk *et al.* [40]. Compared to the hard ones, the soft constraint of trust region is more feasible and easier implemented, thus it is employed in this paper to guarantee that the solution of the current subproblem doesn't deviate exceedingly away from the reference trajectory obtained in the last iteration step. The soft constraint of the decision variables z is defined as

$$J_{tr} = w_{tr}\boldsymbol{\delta}_z^T\boldsymbol{\delta}_z \tag{37}$$

where $\boldsymbol{\delta}_z = z - \bar{z}$ and $w_{tr} \in \mathbb{R}_{++}$ is a positive weight. The desirable J_{tr} is zero, i.e., $\delta_z = 0$, under which condition the algorithm is converged.

In summary, the convex optimization subproblem of the ascent trajectory is modified as **P3** in this paper, and the SCvx algorithm is presented as follows.

$$\begin{aligned}\textbf{P3}: \quad & \min\ J = \phi(\boldsymbol{x}_f) + J_{vc} + J_{tr} \\ & \text{Subject to Eqs.(34), (4), (13) and (16)}\end{aligned}$$

Algorithm 1 Ascent Trajectory Optimization Problem SCvx Algorithm

Initialize the reference trajectory $\bar{z}$
while the convergence condition isn't satisfied
  solve the optimization subproblem **P3** and get the optimal solution z
  if $\sup_{1\le k\le K} |\boldsymbol{x}_k - \bar{\boldsymbol{x}}_k| \le \varepsilon$
    convergence condition is satisfied
  end if
  $z \rightarrow \bar{z}$
end while
return z

IV. NUMERICAL SOLUTION

In this section, a small guided rocket is taken to evaluate the proposed method for the ascent trajectory optimization problem. Firstly, the parameters and conditions in the numerical calculations are given in subsection IV.A. Next, the convergence of the SCvx with the new control is analyzed through the solutions of the sequential subproblems in subsection IV.B, and in addition, the results obtained by another two traditional controls mentioned in above are used as comparison. Then, the accuracy is verified further by the comparison of the solutions worked out by GPOPS and ICLOCS2 (Imperial College London Optimal Control Software) [41] in subsection IV.C. Furthermore, the feasibility of the proposed method is tested by the optimal results of different missions IV.D. Finally, the stability of the proposed method is illustrated by the Monte-Carlo simulation results under the deviations in trajectory initial state variables and vehicle uncertainties in subsection IV.E.

A. SIMULATION CONDITIONS

The parameters of the small rocket are given in Table 2, the change of the motor thrust is presented in Fig.1, and correspondingly, the mass curve is drawn in Fig.2. The fitting of the aerodynamic coefficients is shown in Fig.3, it is noticed that the aerodynamic coefficients of the phase $t \le 15$s and $t > 15$s are different because of the deployment of the folding wing at $t = 15$s, and this is used to evaluate whether the proposed new control can deal with the step changes of the aerodynamic coefficients. Besides, the reference parameters of the normalization are chosen as reference length $r_E = 20000$m, reference velocity $v_E = 442.72$m/s, and reference time $t_E = 45.18$s.

The MATLAB modeling toolbox YALMIP [42] is used to formulate the subproblem **P3** and the solver MOSEK [43], [44] is employed to obtain the solution iteratively in this paper. All the results are obtained by running the proposed method on desktop with Intel Core i3-4150 3.50GHz, 4G RAM. The dimensionless convergence

TABLE 2. Simulation parameters.

Parameter	Value
Initial mass of rocket	100.93kg
Reference area	0.0255m2
Range of angle-of-attack	[-15, 15]deg
Range of bank angle	[-180, 180]deg
Discretized points	201
Weight of virtual control	1e6
Weight of trust region constraint	0.05

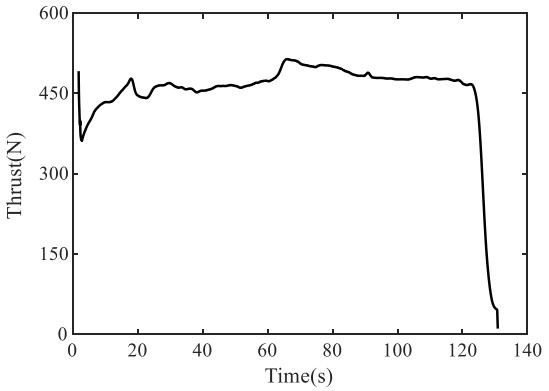


FIGURE 1. Thrust curve.

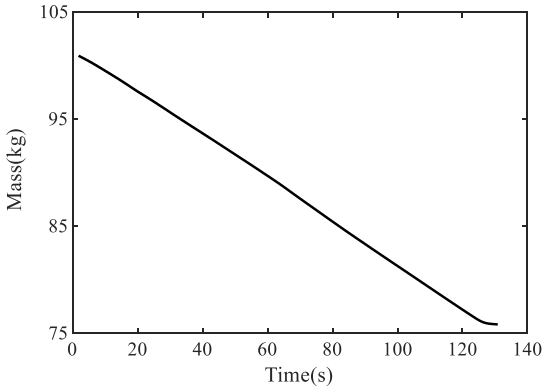
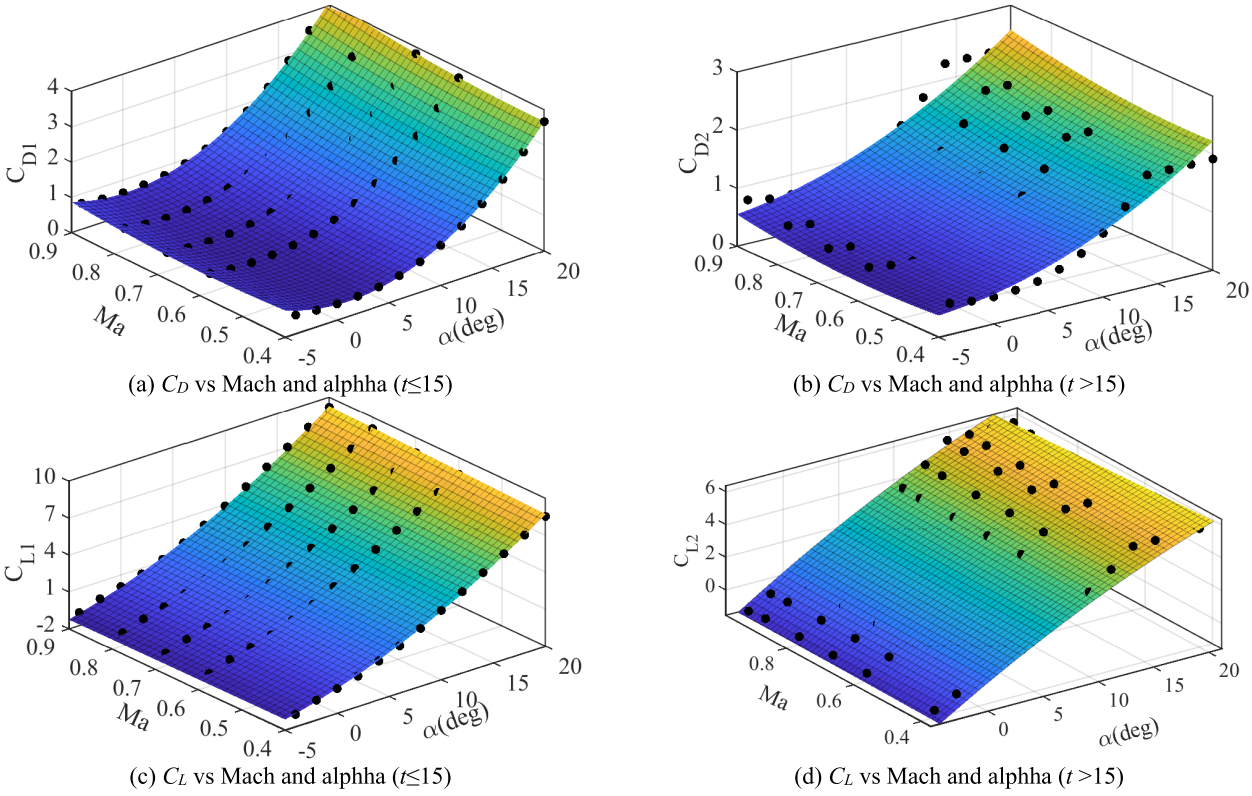


FIGURE 2. Mass curve.



(a) C_D vs Mach and alphha ($t \leq 15$)
(b) C_D vs Mach and alphha ($t > 15$)
(c) C_L vs Mach and alphha ($t \leq 15$)
(d) C_L vs Mach and alphha ($t > 15$)

FIGURE 3. Fitting of aerodynamic coefficients.

condition is selected as

$$\varepsilon = \left[\frac{15}{r_{\mathrm{E}}} \quad \frac{15}{r_E} \quad \frac{15}{r_E} \quad \frac{1}{v_E} \quad \frac{\pi}{180} \quad \frac{\pi}{180} \right]$$

B. OPTIMAL SOLUTION OF THE PROPOSED METHOD

Table 3 lists the initial and terminal conditions of the ascent trajectory optimization problem, where the initial values refer to those after the oblique launching with an impulse thrust. The performance of the proposed method is investigated from the following three aspects in this subsection. Firstly, the convergence of the iteration for the proposed method is presented thoroughly. Next, the advantage of the proposed new control is illustrated through the comparison of the solutions obtained by another two forms of controls, of which one is made up directly by the attack angle and bank angle, and the other consists of the derivatives of attack angle and bank angle. Finally, the integral solutions are calculated by the optimal discrete controls to illustrate the accuracy of the discrete solutions.

1) ITERATION OF THE PROPOSED METHOD

The iteration of the successive subproblems for the ascent trajectory optimization problem is presented in Fig.4.

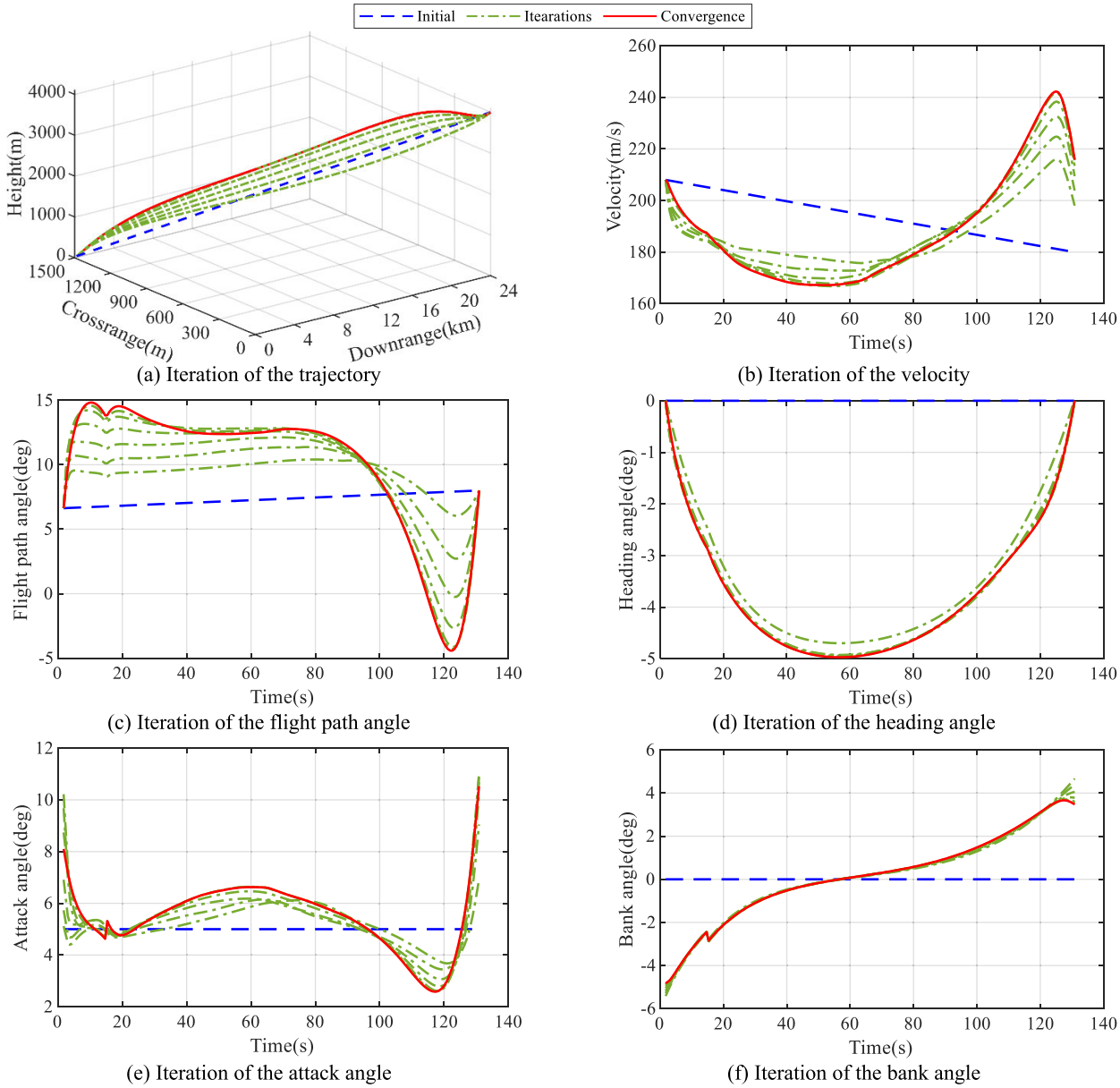
As shown by Fig.4(a)∼(d), we can know that all the states will approach the convergences quickly and stably, even the initial guesses which derive from the points spaced evenly between the initial and terminal conditions are coarse.

TABLE 3. Initial and terminal conditions.

State(unit)	Initial value	Terminal value
x(m)	190	24000
y(m)	29	4000
z(m)	1500	0
v(m/s)	208	max
θ(deg)	6.6	8.0
σ(deg)	0	0
t(s)	0.0	130

The terminal conditions are satisfied well. The velocity decrease in the initial phase because of the large drag, and the maximum terminal velocity is achieved by managing the height in the final phase of flight. The good convergence should not only owe to the linearization of the nonlinear dynamics and the lossless relaxation of nonconvex control constraints, but also benefit from the feasible soft constraint of the trust region. Fig.4(e) and Fig.4(f) are iterations of the actual controls, the convergences are smooth except when the shape of the rocket changes at $t = 15$s, at which time there are small steps.

Furthermore, the difference of the states among the successive solutions is listed in Table 4, which can describe the convergence of the iteration more quantitatively. It can be drawn that the difference between the successive solutions satisfies the convergence condition after 7 iterations and the total CPU time is 1.39s. Besides that, the



(a) Iteration of the trajectory
(b) Iteration of the velocity
(c) Iteration of the flight path angle
(d) Iteration of the heading angle
(e) Iteration of the attack angle
(f) Iteration of the bank angle

FIGURE 4. Iteration of the successive subproblems solved by SCvx with new control.

TABLE 4. Difference of the state among successive solutions obtained by SCvx with new control.

Iteration Step	Δx (m)	Δy (m)	Δz (m)	Δv (m/s)	$\Delta\theta$ (deg)	$\Delta\sigma$ (deg)	Step CPU time(s)	Total CPU time(s)
1	549	102	147.	34.7	2.97	4.70	0.19	0.19
2	304	195	64.8	8.77	3.32	0.42	0.22	0.41
3	165	205	10.4	7.75	2.99	0.10	0.20	0.61
4	102	183	6.42	5.84	2.47	0.07	0.22	0.83
5	58.7	114	3.38	4.14	1.58	0.05	0.19	1.02
6	10.8	15.0	1.47	1.04	0.73	0.07	0.17	1.19
7	11.3	11.5	1.36	0.57	0.43	0.04	0.20	1.39

difference almost decreases with the iterations of the subproblems. These indicate the proposed method converging to the optimal solution quickly and stably.

The subplots of Fig.5 are the relationship between u_1 and u_2 and the relationship between u_3 and u_4, respectively. As we can see from the figure, the optimal solution meets the conditions that $u_1^{*2} = u_2^*$ and $u_3^{*2} + u_4^{*2} = 1$. These illustrate that the constraints g_3 and g_4 in Eq.(16) are active at almost all the discretized points, which means that **Proposition** in subsection II.C is true.

2) COMPARISON OF DIFFERENT CONTROLS

It is discovered in our study that when the SCvx is combined with the other two forms of controls, the iterative steps needed for convergence are near 20. So many successive solutions are difficult to be displayed distinctly in one figure, here only the differences among the successive solutions are listed

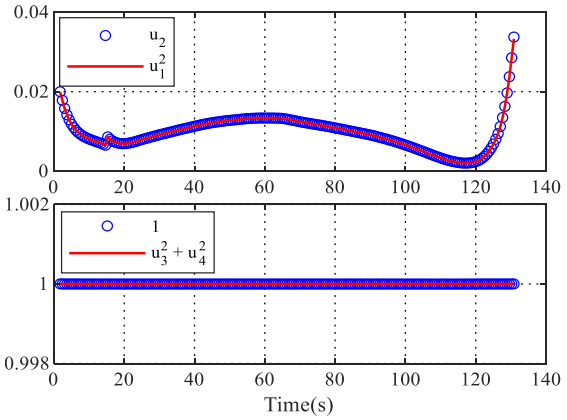


FIGURE 5. Inner constraints of the new control of the optimal solution.

TABLE 5. Difference of the state among successive solutions obtained by SCvx with control 1.

Iteration Step	Δx (m)	Δy (m)	Δz (m)	Δv (m/s)	$\Delta\theta$ (deg)	$\Delta\sigma$ (deg)	Step CPU time(s)	Total CPU time(s)
1	1505	1205.	146	50.0	23.7	4.69	0.14	0.14
2	286	675	62.6	13.0	8.83	1.07	0.14	0.28
3	539	246	39.6	17.8	4.23	0.42	0.12	0.40
4	277	169	15.6	6.05	3.37	0.23	0.13	0.53
5	228	123	12.3	5.12	1.86	0.09	0.16	0.69
6	170	102	10.2	3.65	1.83	0.13	0.13	0.82
7	133	75.3	8.59	3.00	1.12	0.07	0.14	0.96
8	108	63.2	7.45	2.39	1.10	0.08	0.13	1.09
9	83.6	48.5	6.38	1.95	0.77	0.06	0.11	1.20
10	70.1	41.9	5.73	1.60	0.88	0.07	0.13	1.33
11	54.4	32.9	5.13	1.33	0.90	0.07	0.13	1.46
12	47.3	30.4	4.81	1.12	0.95	0.08	0.12	1.58
13	36.6	23.8	4.48	1.13	1.00	0.08	0.11	1.69
14	33.2	24.0	4.40	0.98	1.02	0.09	0.11	1.80
15	25.2	17.9	4.17	1.22	1.07	0.08	0.13	1.93
16	24.2	19.9	4.27	1.03	1.07	0.09	0.11	2.04
17	17.7	13.67	4.04	1.26	1.11	0.09	0.14	2.18
18	18.2	17.0	4.29	1.14	1.10	0.10	0.11	2.29
19	12.5	10.2	4.00	1.27	1.13	0.09	0.09	2.38
20	14.1	14.8	4.40	1.20	1.11	0.10	0.14	2.52

in Table 5 and 6 (where the attack angle and bank angle are regards as control 1 and their derivatives as control 2).

When the above ascent trajectory optimization problem is solved by the SCvx with control 1, the discretized attack angle will change dramatically after 19 iterations, and even divergence in the following iterations, thus only the difference of the state among the first 20 iterations is given in Table 5. Comparing the difference of the state in Table 4 and 5, we can know that the convergence speed of control 1 is slower than that of the new control, especially when the difference is small. This means that the SCvx with control 1 spends much more CPU time to solve the optimization problem with the requirement of high accuracy. In addition, when the step CPU time is taken into consideration, the average step CPU time of the new control is more than that of control 1 because of its more decision variables, but the drawback is compensated well by its quick convergence. Thus, the SCvx with the proposed new control spends near the half time of the control 1, which can be illustrated by the total CPU time.

TABLE 6. Difference of the state among successive solutions obtained by SCvx with control 2.

Iteration Step	Δx (m)	Δy (m)	Δz (m)	Δv (m/s)	$\Delta\theta$ (deg)	$\Delta\sigma$ (deg)	Step CPU time(s)	Total CPU time(s)
1	1694	1377	179	50.0	23.1	5.38	0.16	0.16
2	764	930	73.5	22.2	9.06	1.13	0.14	0.30
3	507	246	17.09	10.90	3.55	0.18	0.16	0.46
4	221	135	10.6	4.74	1.61	0.05	0.14	0.60
5	162	107	7.76	3.39	1.18	0.04	0.16	0.76
6	130	82.5	6.34	2.73	0.93	0.03	0.16	0.92
7	104	63.1	5.14	2.19	0.72	0.02	0.17	1.09
8	83.7	49.0	4.20	1.76	0.56	0.02	0.12	1.21
9	67.6	38.7	3.46	1.43	0.44	0.01	0.14	1.35
10	55.1	31.0	2.88	1.17	0.36	0.01	0.16	1.51
11	45.4	25.3	2.42	0.98	0.30	0.01	0.14	1.65
12	37.6	20.8	2.06	0.83	0.25	0.01	0.16	1.81
13	31.5	17.4	1.76	0.71	0.21	0.01	0.14	1.95
14	26.5	14.7	1.52	0.62	0.18	0.01	0.13	2.08
15	22.4	12.5	1.33	0.54	0.17	0.01	0.14	2.22
16	19.1	10.7	1.17	0.48	0.15	0.00	0.16	2.38
17	16.3	9.29	1.03	0.42	0.14	0.00	0.12	2.50
18	14.0	8.09	0.92	0.38	0.14	0.00	0.14	2.64

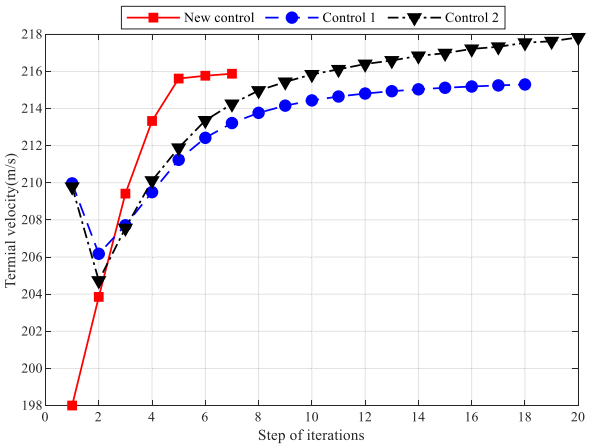
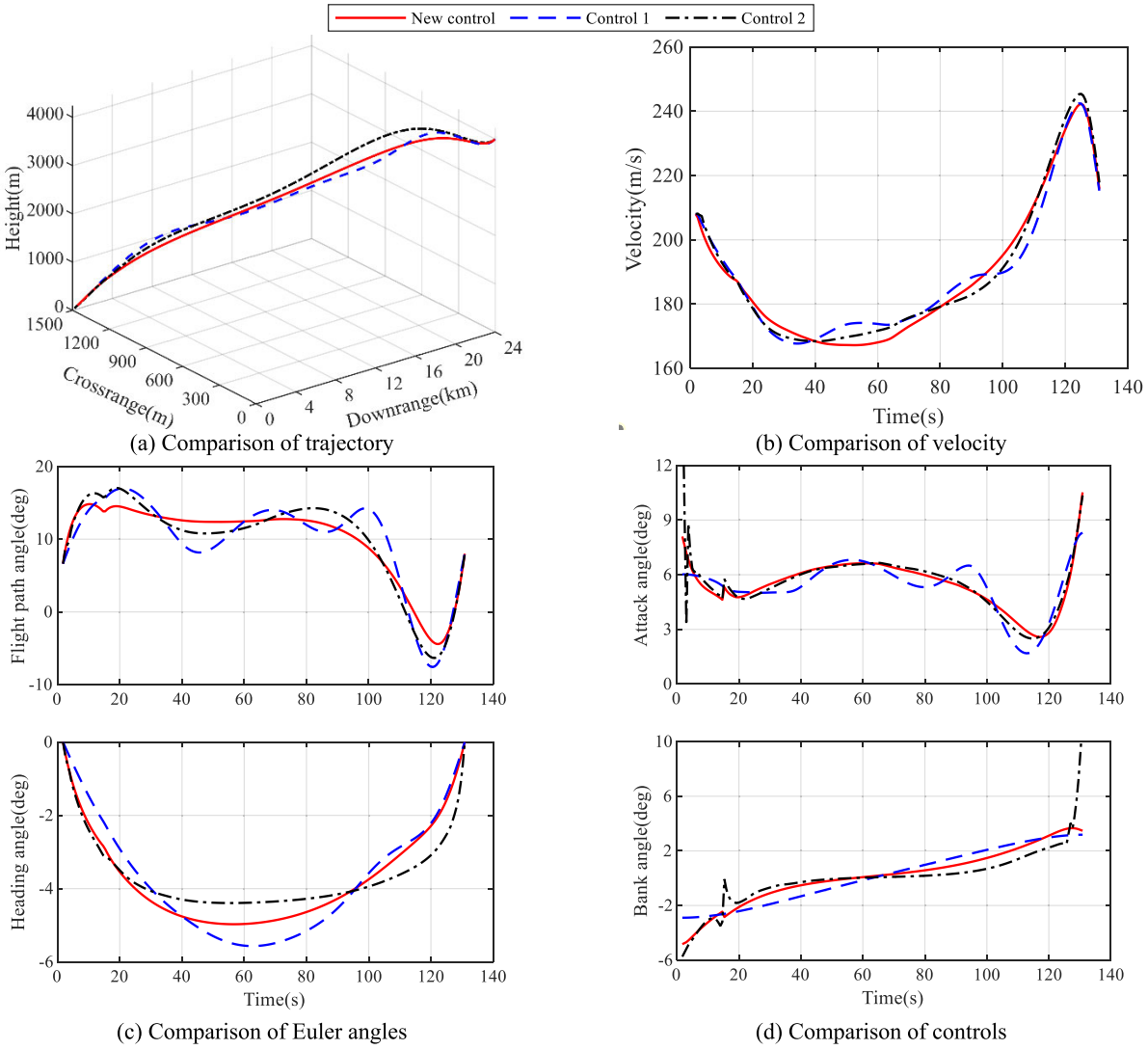


FIGURE 6. Comparison of the terminal velocity under different controls.

It can be known from Table 6 that the SCvx with control 2 takes 18 iterations to approach its solution, which is similar to that of control 1, and the convergence speed is slower than that of the proposed new control too. Additionally, the average step CPU time of control 2 is less than that of the proposed new control even though the numbers of the decision variables for the both are the same. To the best of our knowledge, the obvious distinction of the step CPU time between the new control and the traditional controls (control 1 and 2) is caused by the feasible sets of controls for the successive subproblems, of which the former is always larger than the latter because of the relaxation technology. As a result, the subproblem solved by the SCvx with the proposed new control will spend more time to find the optimal solution.

Fig.6 presents the comparison of the successive terminal velocities obtained by the SCvx with the three controls, which



(a) Comparison of trajectory

(b) Comparison of velocity

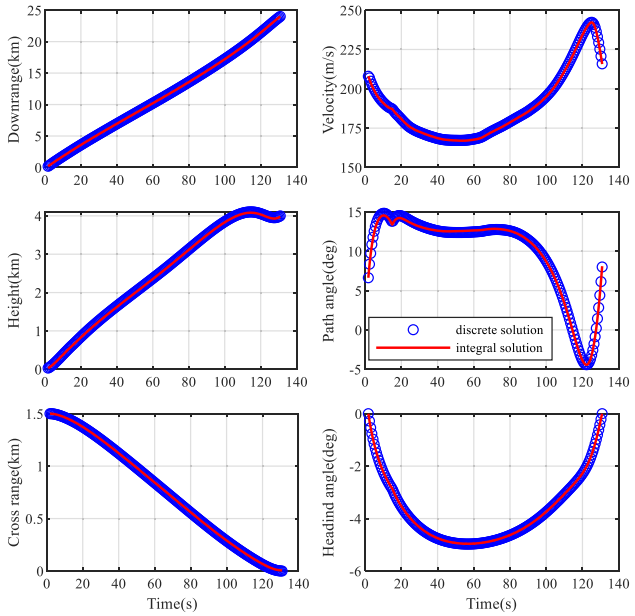
(c) Comparison of Euler angles

(d) Comparison of controls

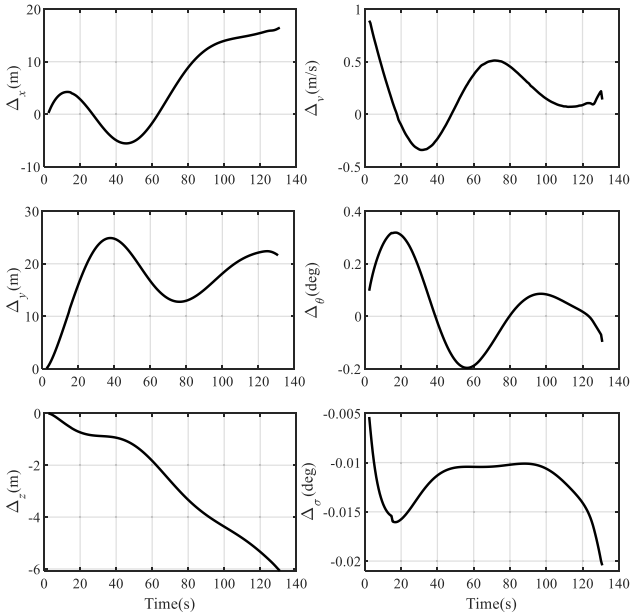
FIGURE 7. **Comparison of the optimal solutions of different controls.**

are the performance indexes of the optimization problem in our study according to the previous description. It can be seen from the figure that the performance index converges more quickly when using proposed the new control. Moreover, the difference between the first terminal velocities of the proposed new control and the traditional controls is conspicuous, but the following terminal velocities change similarly and stably. These mean that the coarse initial guess leads to the different solutions in the first iterations for the different controls, but does not affect the convergence of the subsequent iterations greatly because of the soft constraint of the trust region. Therefore, the SCvx algorithm doesn't need the accurate initial value and has a good convergence.

Fig.7 shows the comparison of the optimal solutions obtained by the three different controls. As we can see from the figure, the optimal solutions of the three controls are similar. The slight differences between the optimal solutions under the different controls should attribute to the cutoff of numerical calculations and convergence conditions. When the attack angle and bank angle are used as the controls directly, they step over 1 and 3 deg respectively at the moment the shape of the rocket changes, which shown by Fig.7(d); what's more, the attack angle changes sharply in the initial phase, and this is difficult for the vehicle to track during the flight. Besides, it can also be seen from the changes of the flight path angle and attack angle in Fig.7 that the derivatives of attack angle and bank angle don't work as expected, the flight path angle and attack angle always fluctuate around the results obtained by the other two controls. Although the derivative of the attack angle could be used to prevent the chatter among the discretized points, the change over the whole interval $[t_0, t_f]$ can't be guaranteed to be smooth. In summary, it can be concluded that the SCvx with the proposed new control can yield smooth discrete control variables when there is no drastic change of aerodynamic coefficients.



(a) Comparison between the histories of integral solutions and discrete solutions



(b) Deviations between integral solutions and discrete solutions

FIGURE 8. Comparison between integral solutions and discrete solutions.

3) COMPARISON WITH THE INTEGRAL SOLUTION

Furthermore, with the purpose to study the accuracy of the optimal solution obtained by the proposed method, the discrete attack angle and bank angle transformed from the optimal new control are interpolated to obtain the integral solution based on the dynamics (1). The integral solution can be regarded as continuous-time if the integration step is considerably small, and it's compared to the optimal discrete solution, which is presented in Fig.8.

It can be seen from the comparison shown by Fig.8(a) that the integral solution has a good agreement with the discrete solution. This high accuracy of the discrete result should attribute to the appropriate interpolation and points in the discretization, as well as the tough convergence condition. Furthermore, to show the difference between the integral solutions and the discrete solutions more quantitively, the deviations are shown in Fig.8(b). When the deviations are studied with the comparison of the convergence condition set in subsection 4.1, it will be interesting to find that the deviations of all states except for the height change within their convergence conditions, but the height deviation still varies in the twice of the convergence condition. This is greatly helpful to select the convergence condition when we employ the SCvx to solve the ascent trajectory optimization problem.

C. COMPARISON WITH SOLUTION OF GPOPS AND ICLOCS2

The typical optimal control solvers ICLOCS2 and GPOPS are utilized to work out the optimal solution of the ascent trajectory problem mentioned above. The solutions are compared to that obtained by the proposed method, which is presented in Fig.9.

TABLE 7. Comparison of the optimal results of Different methods.

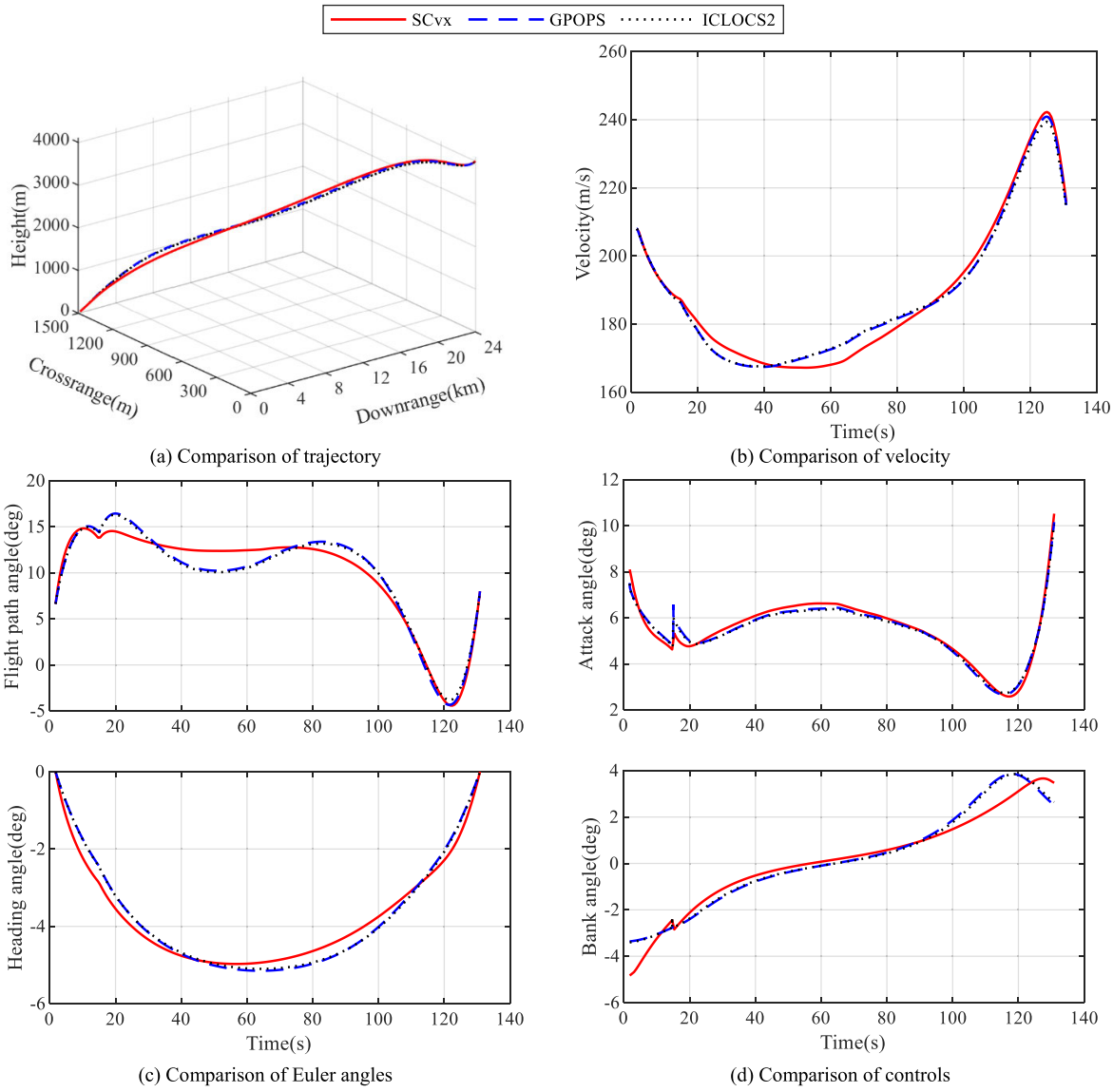
Method	Terminal velocity (m/s)	Total CPU time (s)
SCvx	215.87	1.39
GPOPS	214.93	15.73
ICLOCS2	215.26	10.24

It can be seen from Fig.9(a) that the optimal trajectories obtained by the different methods are almost the same, but the other states and the controls are slightly different. These negligible differences should result from the linearization and discretization strategies of the three methods. Additionally, the terminal velocity and total CPU time of these methods are compared in Table 7. The table indicates that all the methods can solve the above ascent trajectory optimization problem successfully with similar results, but GPOPS and ICLOCS2 expense much more time than the SCvx.

D. OPTIMIZATION RESULTS UNDER DIFFERENT MISSIONS

In this subsection, the proposed method is implemented to solve the ascent trajectory optimization problems under different missions, of which the different initial and terminal positions are listed in Table 8, the other conditions and parameters are the same as those in subsection 4.1.

Fig.11 shows the comparison of the optimal solutions solved by the proposed method under different missions. The three-dimensional trajectory together with the altitude



(a) Comparison of trajectory

(b) Comparison of velocity

(c) Comparison of Euler angles

(d) Comparison of controls

FIGURE 9. Comparison of solutions obtained by proposed SCvx and GPOPS.

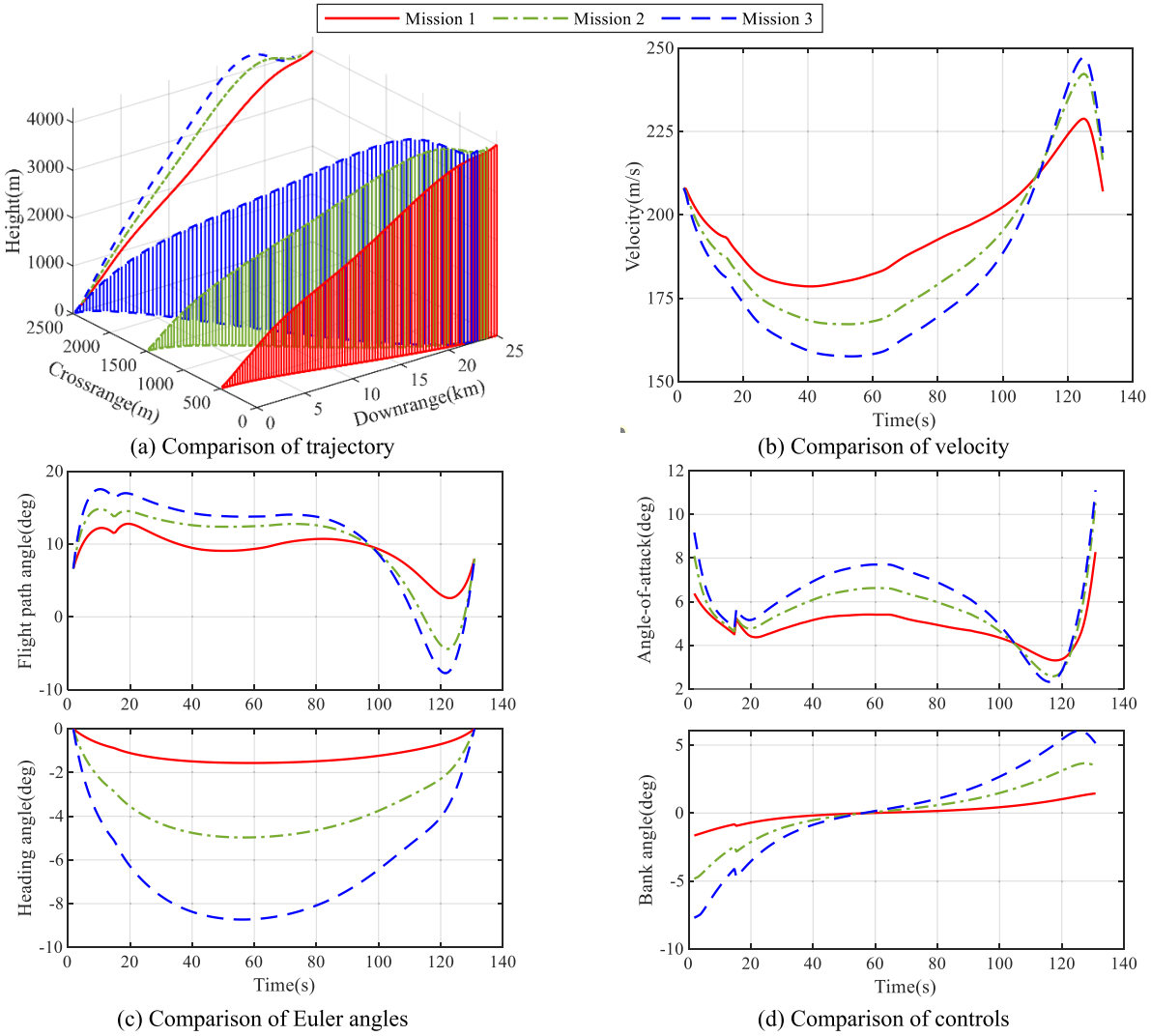
TABLE 8. Initial and terminal positions for different missions.

Mission	Initial position(m)	Terminal position(m)
1	190 29 500	25000 4000 0
2	190 29 1500	24000 4000 0
3	190 29 2500	23000 4000 0

history and the ground track of three different missions is presented in Fig.11(a), and it indicates that the proposed algorithm is feasible and effective to solve the optimization problems with different missions. As a fixed terminal-time maximum-velocity problem, the maximum terminal velocity is achieved by the managing energy in the vertical under the mission 2 and 3, this is also illustrated by the sign of the flight path angle in Fig.11(c), consequently, the terminal velocities of the two missions are larger than that of mission 1. In addition, the optimal controls of different missions in Fig.11(d) are smooth except for t = 15s, at which time small steps are caused by the change of the aerodynamic shape, but still within the allowable range.

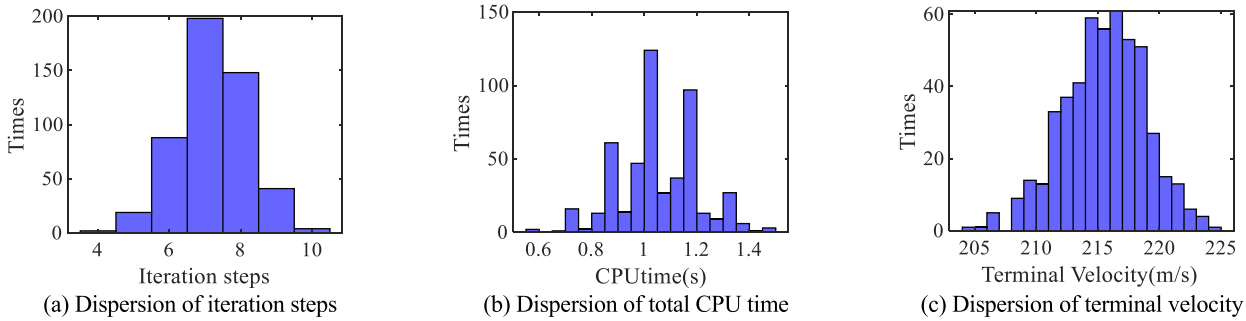
E. OPTIMAL SOLUTIONS UNDER DEVIATIONS AND UNCERTAINTIES

In this subsection, Monte-Carlo simulation is conducted to evaluate the stability of the proposed method under the deviations in initials and the uncertainties of the rocket.



(a) Comparison of trajectory
(b) Comparison of velocity
(c) Comparison of Euler angles
(d) Comparison of controls

FIGURE 10. Comparison of the solutions under different missions.

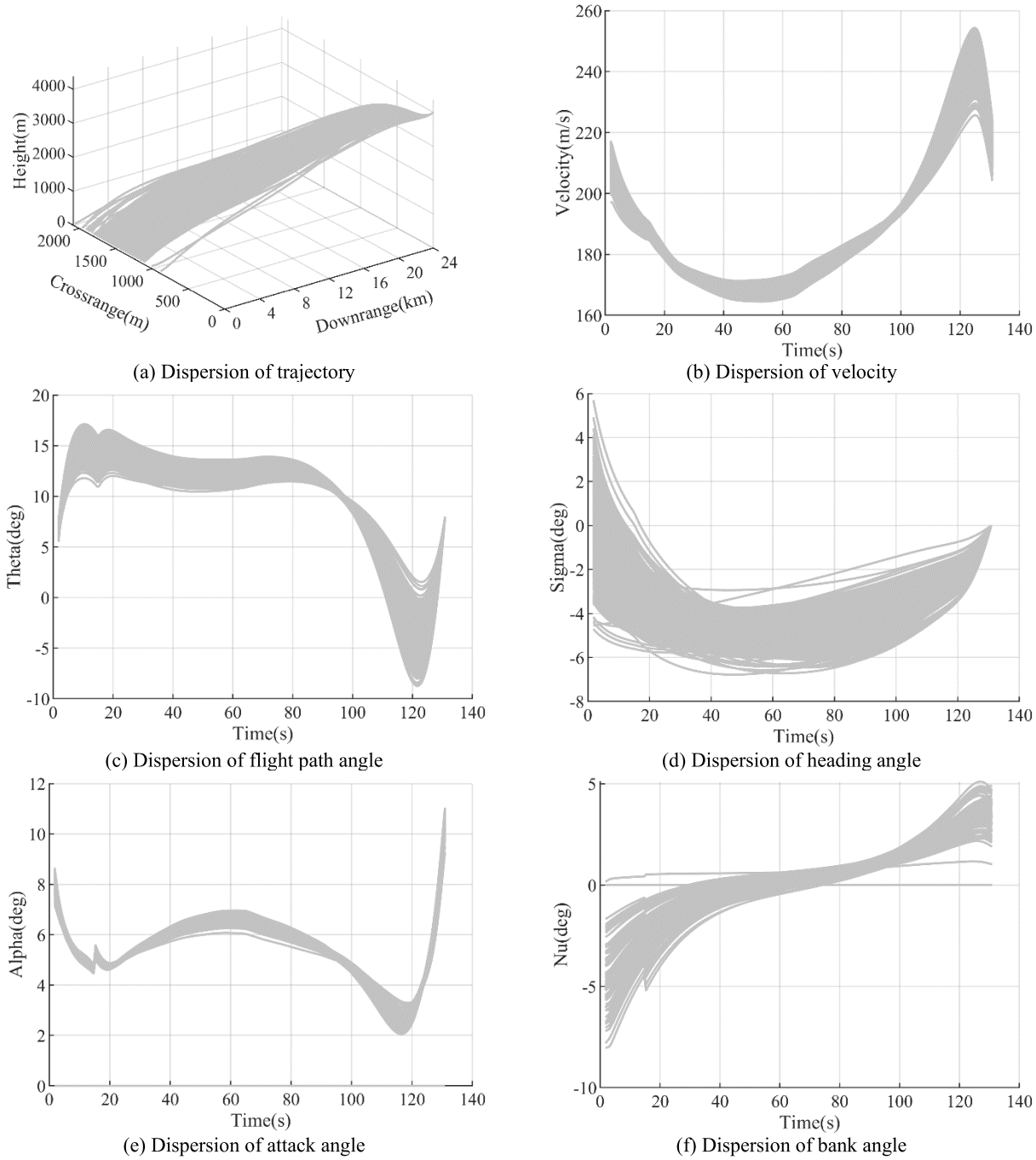


(a) Dispersion of iteration steps
(b) Dispersion of total CPU time
(c) Dispersion of terminal velocity

FIGURE 11. Dispersion of the performance under the deviations and uncertainties.

The dispersion of the initial and the uncertainty are listed in Table.9. 500 Monte-Carlo simulations are implemented under the mission 2.

The dispersion of 500 simulations for the states and controls under the deviations and uncertainties is shown in Fig.12, it illustrates that the terminal conditions are satisfied stably when using the proposed method to solve the trajectory optimization problem under the deviations and uncertainties. Furthermore, the performance of the proposed method can be evaluated from the Fig.11, which



(a) Dispersion of trajectory
(b) Dispersion of velocity
(c) Dispersion of flight path angle
(d) Dispersion of heading angle
(e) Dispersion of attack angle
(f) Dispersion of bank angle

FIGURE 12. Dispersion of the state and control under the deviations and uncertainties.

presents the dispersions of the iteration steps, total CPU time and the terminal velocity, it can be known from the figure that the iteration steps and the CPU time change within the stable range, and the dispersions correspond to the dispersion model of the deviations and uncertainties well; the dispersion range of the terminal velocity is almost the same as the 3σ-range of the distribution for the initial velocity.

The comparative result of the typical optimal control solvers ICLOCS2 and GPOPS is displayed in Table 10, the similar conclusion can be drawn as the that in IV.C.

However, as a fixed final time problem, the system would fail when current velocity could not reach the terminal condition because of the deviations and uncertainties, i.e., the maximum range of the fixed time interval under the deviations and uncertainties is less than the range required. This drawback

TABLE 9. Dispersions of the initial deviation and uncertainty.

State(unit)	Distribution	3σ Range
z_0(m)	Gaussian with zero mean	500
v_0(m/s)	Gaussian with zero mean	10
θ_0(deg)	Gaussian with zero mean	1
σ_0(deg)	Gaussian with zero mean	5
mass(kg)	Uniform	2%
thrust(N)	Uniform	2%

TABLE 10. Statistical result of Monte-Carlo simulation.

Method		SCvx	GPOPS	ICLOCS2
iteration steps	Minimum	4	4	3
	Maximum	10	10	7
	Average	7.22	7.65	5.25
Total CPU time(s)	Minimum	0.55	5.85	4.89
	Maximum	1.49	22.63	18.26
	Average	1.23	14.23	8.65
Terminal velocity (m/s)	Minimum	204.04	203.56	204.35
	Maximum	224.02	223.74	224.42
	Average	215.51	214.62	214.85

can be addressed by replacing the time t with the downrange x as the new independent variable.

V. CONCLUSION

In this paper, the successive convexification is successfully applied to the 3-Dof ascent trajectory optimization problems for the small guided rocket with complex nonlinear aerodynamics, the optimal solutions can be worked out rapidly. The new control and effective relaxation are designed to address the nonconvex of the aerodynamics, which is proved theoretically by Karush-Kuhn-Tucker condition and Pontryagin's minimum principle. A soft constraint of the trust region is put forward to avoid the artificial unboundedness and nonconvergence of the algorithm. Based on linearization and discretization, the proposed method can solve the nonlinear continuous-time optimal control problem accurately. The results of simulation illustrate that 1) the SCvx with proposed new control approaches more rapidly to the convergence than the traditional controls even if the initial guess is coarse; 2) the optimal solution obtained by utilizing the proposed method is considerably accurate; and 3) the proposed algorithm is feasible and stable to solve the ascent trajectory optimization problem. Therefore, the method is potential to be implemented online.

The future work of the paper will be transplanting the proposed method to the real-platform, including improving the model structural parameters, enhancing the feasibility and stability further. In addition, the proposed method will also be extended to solve other trajectory optimization problems, such as the reentry trajectory optimization and trajectory optimiza-tion of multi-vehicles.

APPENDIX

This section provides a detailed proof of the **Proposition** by Karush-Kuhn-Tucker condition and Pontryagin's minimum principle. Generally, the Hamiltonian and Lagrange functions of the ascent trajectory optimization problem **P2** are Eq.(A1) and (A2) respectively.

$$\begin{aligned} H &= \lambda_v\left[\frac{P}{mg_0}-\frac{r_E\rho v^2 S_{ref}}{2m}\left(\hat{C}_D^0+\hat{C}_D^{\alpha}u_1+\hat{C}_D^{\alpha^2}u_2\right)-\sin\theta\right] \\ &\quad+\lambda_\theta\left[\frac{P}{mg_0 v}u_1+\frac{r_E\rho v S_{ref}u_3}{2m}\left(\hat{C}_L^0+\hat{C}_L^{\alpha}u_1+\hat{C}_L^{\alpha^2}u_2\right)\right. \\ &\quad\left.-\frac{\cos\theta}{v}\right] \\ &\quad+\lambda_\sigma\frac{r_E\rho v S_{ref}u_4}{2m\cos\theta}\left(\hat{C}_L^0+\hat{C}_L^{\alpha}u_1+\hat{C}_L^{\alpha^2}u_2\right) \\ &\quad+\lambda_x v\cos\theta\cos\sigma \\ &\quad+\lambda_y v\sin\theta+\lambda_z v\cos\theta\sin\sigma \end{aligned} \tag{A1}$$

$$\begin{aligned} L &= H+\lambda_1\left(u_2^- - u_2\right)+\lambda_2\left(u_2-u_2^+\right)+\lambda_3\left(u_1^2-u_2\right) \\ &\quad+\lambda_4\left(u_3^2+u_4^2-1\right) \end{aligned} \tag{A2}$$

where λ_x, λ_y, λ_z, λ_v, λ_θ, λ_σ are the costate variables for the states, and λ_1, λ_2, λ_3, λ_4 are the Lagrange multipliers for the inequality constraints in Eq.(13) and(16).

$$\lambda_{xf}=0,\lambda_{yf}=0,\lambda_{zf}=0,\lambda_{vf}=-1,\lambda_{\theta f}=0,\lambda_{\sigma f}=0 \tag{A3}$$

The stationary conditions (or KKT conditions) are as follows.

$$\begin{aligned} \frac{\partial L}{\partial u_1} &= \frac{\partial H}{\partial u_1}+2\lambda_1 u_1 \\ &= -\lambda_v\frac{r_E\rho v^2 S_{ref}\hat{C}_D^0}{2m}+\lambda_\theta\left(\frac{P}{mg_0 v}+\frac{r_E\rho v S_{ref}\hat{C}_L^{\alpha}}{2m}u_3\right) \\ &\quad+\lambda_\sigma\frac{r_E\rho v S_{ref}\hat{C}_L^{\alpha}}{2m\cos\theta}u_4+2\lambda_3 u_1=0 \end{aligned} \tag{A4}$$

$$\begin{aligned} \frac{\partial L}{\partial u_2} &= \frac{\partial H}{\partial u_2}-\lambda_1+\lambda_2-\lambda_3 \\ &= -\lambda_v\frac{r_E\rho v^2 S_{ref}\hat{C}_D^{\alpha^2}}{2m}+\lambda_\theta\frac{r_E\rho v S_{ref}\hat{C}_L^{\alpha^2}}{2m}u_3 \\ &\quad+\lambda_\sigma\frac{r_E\rho v S_{ref}\hat{C}_L^{\alpha}}{2m\cos\theta}u_4-\lambda_1+\lambda_2-\lambda_3=0 \end{aligned} \tag{A5}$$

$$\begin{aligned} \frac{\partial L}{\partial u_3} &= \frac{\partial H}{\partial u_3}+2\lambda_4 u_3=\lambda_\theta\frac{r_E\rho v S_{ref}}{2m}\left(\hat{C}_L^0+\hat{C}_L^{\alpha}u_1+\hat{C}_L^{\alpha^2}u_2\right) \\ &\quad+2\lambda_4 u_3=0 \end{aligned} \tag{A6}$$

$$\begin{aligned} \frac{\partial L}{\partial u_4} &= \frac{\partial H}{\partial u_4}+2\lambda_4 u_4=\lambda_\sigma\frac{r_E\rho v S_{ref}}{2m\cos\theta}\left(\hat{C}_L^0+\hat{C}_L^{\alpha}u_1+\hat{C}_L^{\alpha^2}u_2\right) \\ &\quad+2\lambda_4 u_4=0 \end{aligned} \tag{A7}$$

The complementary slack conditions are denoted as Eqs. (A8)∼(A11).

$$\lambda_1\geq 0,\quad \lambda_1\left(u_2^- - u_2\right)=0 \tag{A8}$$
$$\lambda_2\geq 0,\quad \lambda_2\left(u_2-u_2^+\right)=0 \tag{A9}$$
$$\lambda_3\geq 0,\quad \lambda_3\left(u_1^2-u_2\right)=0 \tag{A10}$$
$$\lambda_4\geq 0,\quad \lambda_4\left(u_3^2+u_4^2-1\right)=0 \tag{A11}$$

The proof is performed through contradiction. Firstly, it will be proven that $u_1^2 \leq u_2$ is an active constraint, after that, $u_3^2 + u_4^2 \leq 1$ is proven active during $[t_0, t_f]$ too.

A. PROOF OF ACTIVE CONSTRAINT $\mathbf{u_1^2 \leq u_2}$

Assume that the condition $u_1^2 \leq u_2$ is not an active constraint, i.e., $u_1^2 < u_2$. According to the complementary slack condition Eq.(A10), we have $\lambda_3 = 0$. Substituting $\lambda_3 = 0$ into Eq.(A4), Eq.(A12) can be obtained during $[t_0, t_f]$.

$$-\lambda_v \frac{r_E \rho v^2 S_{ref} \hat{C}_D^0}{2m} + \lambda_\theta \left(\frac{P}{m g_0 v} + \frac{r_E \rho v S_{ref} \hat{C}_L^\alpha}{2m} u_3 \right) + \lambda_\sigma \frac{r_E \rho v S_{ref} \hat{C}_L^\alpha}{2m \cos\theta} u_4 = 0 \quad \text{(A12)}$$

Substituting Eq.(A3) into Eq.(A12), it is easy to know that $\lambda_{vf} = 0$ when $t = t_f$, which obviously contradicts the transversality condition $\lambda_{vf} = -1$. Therefore, the assumption $u_1^2 < u_2$ is invalid, and on the contrary, $u_1^2 \leq u_2$ is an active constraint.

B. PROOF OF ACTIVE CONSTRAINT $\mathbf{u_3^2 + u_4^2 \leq 1}$

Assume that $u_3^2 + u_4^2 \leq 1$ is not an active constraint, i.e., $u_3^2 + u_4^2 < 1$. According to the complementary slack condition equation Eq.(A11), we have $\lambda_4 = 0$. Substituting $\lambda_4 = 0$ into Eq.(A6) and Eq.(A7), we can get Eq.(A13).

$$\begin{cases} \lambda_\theta \dfrac{r_E \rho v S_{ref}}{2m} \left(\hat{C}_L^0 + \hat{C}_L^\alpha u_1 + \hat{C}_L^{\alpha^2} u_2 \right) = 0 \\ \lambda_\sigma \dfrac{r_E \rho v S_{ref}}{2m \cos\theta} \left(\hat{C}_L^0 + \hat{C}_L^\alpha u_1 + \hat{C}_L^{\alpha^2} u_2 \right) = 0 \end{cases} \quad \text{(A13)}$$

It can be deduced easily from the Eq.(A13) that the only terms can be zeros are λ_θ and λ_σ. Combined with the transversality condition $\lambda_{\theta f} = 0$ and $\lambda_{\sigma f} = 0$ in Eq.(A3), we can know that $\lambda_\theta \equiv 0$ and $\lambda_\sigma \equiv 0$. The derivatives of the costate λ_θ and λ_σ which are obtained from the minimum principle can be expressed as

$$\begin{aligned} \dot{\lambda}_\theta &= -\frac{\partial H}{\partial \theta} \\ &= \lambda_v \cos\theta - \lambda_\theta \frac{\sin\theta}{v} - \lambda_\sigma \frac{r_E \rho v S_{ref} C_L \sin\theta u_4}{2m \cos^2\theta} \\ &\quad + \lambda_x v \sin\theta \cos\sigma - \lambda_y v \cos\theta + \lambda_z v \sin\theta \sin\sigma = 0 \end{aligned} \quad \text{(A14)}$$

$$\dot{\lambda}_\sigma = -\frac{\partial H}{\partial \sigma} = \lambda_x v \cos\theta \sin\sigma - \lambda_z v \cos\theta \cos\sigma = 0 \quad \text{(A15)}$$

Substituting the transversality condition (A3) into Eq.(A14), the possible condition that Eq.(A14) is satisfied at $t = t_f$ is $\lambda_{vf} = 0$ or $\cos\theta_f = 0$. Because the condition $\cos\theta_f = 0$ corresponds to a special trajectory, the only condition for Eq.(A14) is $\lambda_{vf} = 0$, which obviously contradicts the transversality condition $\lambda_{vf} = -1$. So, the assumption $u_3^2 + u_4^2 < 1$ is invalid, and on the contrary, $u_3^2 + u_4^2 \leq 1$ is an active constraint.

In conclusion, $u_1^2 \leq u_2$ and $u_3^2 + u_4^2 \leq 1$ are active constraints, therefore the relaxation technique is without loss.

REFERENCES

[1] K. Lu, Q. Gong, J. Hou, J. Liu, Z. Qi, and Z. Song, "Intelligent autonomous system and its application in flight vehicles," in *Proc. 35th Chin. Control Conf. (CCC)*, Jul. 2016, pp. 6–12.
[2] O. von Stryk and R. Bulirsch, "Direct and indirect methods for trajectory optimization," *Ann. Oper. Res.*, vol. 37, no. 1, pp. 357–373, Dec. 1992.
[3] J. T. Betts, "Survey of numerical methods for trajectory optimization," *J. Guid., Control, Dyn.*, vol. 21, no. 2, pp. 193–207, Mar. 1998.
[4] C. R. Hargraves and S. W. Paris, "Direct trajectory optimization using nonlinear programming and collocation," *J. Guid., Control, Dyn.*, vol. 10, no. 4, pp. 338–342, Jul. 1987.
[5] 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, Nov. 2006.
[6] G. Dukeman, "Atmospheric ascent guidance for rocket-powered launch vehicles," in *Proc. AIAA Guid., Navigat., Control Conf. Exhib.*, Aug. 2002, p. 4559.
[7] G. Dukeman and A. Calise, "Enhancements to an atmospheric ascent guidance algorithm," in *Proc. AIAA Guid., Navigat., Control Conf. Exhib.*, Aug. 2003, p. 5638.
[8] A. J. Calise, N. Melamed, and S. Lee, "Design and evaluation of a three-dimensional optimal ascent guidance algorithm," *J. Guid., Control, Dyn.*, vol. 21, no. 6, pp. 867–875, Nov. 1998.
[9] P. Lu, H. Sun, and B. Tsai, "Closed-loop endoatmospheric ascent guidance," *J. Guid., Control, Dyn.*, vol. 26, no. 2, pp. 283–294, Mar. 2003.
[10] P. Lu and B. Pan, "Highly constrained optimal launch ascent guidance," *J. Guid., Control, Dyn.*, vol. 33, no. 2, pp. 404–414, Mar. 2010.
[11] B. Lu, N. Cui, Y. Fu, W. Shan, and X. Chang, "Closed-loop atmospheric ascent guidance based on finite element method," *Aircr. Eng. Aerosp. Technol.*, vol. 87, no. 5, pp. 393–401, Sep. 2015.
[12] P. Huang, Y. Guan, and R. Huang, "A mixed variable variational method for optimal endo-atmospheric ascent guidance," in *Proc. IEEE Aerosp. Conf.*, Mar. 2015, pp. 1–8.
[13] W. Shi, Z. Jing, and Y. Yang, "Ascent trajectory optimisation for hypersonic vehicles via Gauss pseudospectral method," *Int. J. Space Sci. Eng.*, vol. 1, no. 1, p. 64, 2013.
[14] F. Fahroo and I. M. Ross, "Direct trajectory optimization by a Chebyshev pseudospectral method," *J. Guid., Control, Dyn.*, vol. 25, no. 1, pp. 160–166, Jan. 2002.
[15] X. Guo and M. Zhu, "Direct trajectory optimization based on a mapped Chebyshev pseudospectral method," *Chin. J. Aeronaut.*, vol. 26, no. 2, pp. 401–412, Apr. 2013.
[16] P. Han, J. Shan, and X. Meng, "Re-entry trajectory optimization using an hp-adaptive Radau pseudospectral method," *Proc. Inst. Mech. Eng., G, J. Aerosp. Eng.*, vol. 227, no. 10, pp. 1623–1636, Oct. 2013.
[17] D. Zhang, X. Lu, L. Liu, and Y. Wang, "An online ascent phase trajectory reconstruction algorithm using Gauss pseudospectral method," *Proc. Int. Conf. Modeling, Identificat. Control*, Wuhan, China, Jun. 2012, pp. 1184–1189.
[18] X. Jiang and S. Li, "Mars atmospheric entry trajectory optimization via particle swarm optimization and Gauss pseudo-spectral method," *Proc. Inst. Mech. Eng., G, J. Aerosp. Eng.*, vol. 230, no. 12, pp. 2320–2329, Oct. 2016.
[19] 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.
[20] R. Chai, A. Savvaris, A. Tsourdos, Y. Xia, and S. Chai, "Solving multi-objective constrained trajectory optimization problem by an extended evolutionary algorithm," *IEEE Trans. Cybern.*, vol. 50, no. 4, pp. 1630–1643, Apr. 2020.
[21] D. Q. Tran and M. Diehl, "An application of sequential convex programming to time optimal trajectory planning for a car motion," in *Proc. 48h IEEE Conf. Decis. Control (CDC) Held Jointly With 28th Chin. Control Conf.*, Dec. 2009, pp. 4366–4371.
[22] S. Zhengxiang, C. Tao, W. Songyan, and Y. Ming, "Convex method for ascent trajectory optimization using iterative narrowing trust region," in *Proc. IEEE CSAA Guid., Navigat. Control Conf. (CGNCC)*, Aug. 2018, pp. 1–6.
[23] X. Cheng, H. Li, and R. Zhang, "Efficient ascent trajectory optimization using convex models based on the Newton–Kantorovich/Pseudospectral approach," *Aerosp. Sci. Technol.*, vol. 66, pp. 140–151, Jul. 2017.

[24] K. Zhang, S. Yang, and F. Xiong, "Rapid ascent trajectory optimization for guided rockets via sequential convex programming," *Proc. Inst. Mech. Eng., G, J. Aerosp. Eng.*, vol. 233, no. 13, pp. 4800–4809, Oct. 2019.
[25] X. Yan and L. He, "Unpowered approach and landing trajectory planning using second-order cone programming," *Aerosp. Sci. Technol.*, vol. 101, Jun. 2020, Art. no. 105841.
[26] B. Acikmese, J. M. Carson, and L. Blackmore, "Lossless convexification of nonconvex control bound and pointing constraints of the soft landing optimal control problem," *IEEE Trans. Control Syst. Technol.*, vol. 21, no. 6, pp. 2104–2113, Nov. 2013.
[27] J. M. Carson, B. Acikmese, and L. Blackmore, "Lossless convexification of powered-descent guidance with non-convex thrust bound and pointing constraints," in *Proc. Amer. Control Conf.*, Jun. 2011, pp. 2651–2656.
[28] Z. Wang and M. J. Grant, "Constrained trajectory optimization for planetary entry via sequential convex programming," *J. Guid., Control, Dyn.*, vol. 40, no. 10, pp. 2603–2615, Oct. 2017.
[29] D.-J. Zhao and Z.-Y. Song, "Reentry trajectory optimization with waypoint and no-fly zone constraints using multiphase convex programming," *Acta Astronautica*, vol. 137, pp. 60–69, Aug. 2017.
[30] X. Liu, Z. Shen, and P. Lu, "Entry trajectory optimization by second-order cone programming," *J. Guid., Control, Dyn.*, vol. 39, no. 2, pp. 227–241, Feb. 2016.
[31] P. Pei, S. Fan, W. Wang, and D. Lin, "Online reentry trajectory optimization using modified sequential convex programming for hypersonic vehicle," *IEEE Access*, vol. 9, pp. 23511–23525, 2021.
[32] X. Zhou, R.-Z. He, H.-B. Zhang, G.-J. Tang, and W.-M. Bao, "Sequential convex programming method using adaptive mesh refinement for entry trajectory planning problem," *Aerosp. Sci. Technol.*, vol. 109, Feb. 2021, Art. no. 106374.
[33] Y. Mao, M. Szmuk, and B. Acikmese, "Successive convexification of non-convex optimal control problems and its convergence properties," in *Proc. IEEE 55th Conf. Decis. Control (CDC)*, Dec. 2016, pp. 3636–3641.
[34] Y. Mao, D. Dueri, M. Szmuk, and B. Açıkmeşe, "Successive convexification of non-convex optimal control problems with state constraints," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 4063–4069, Jul. 2017.
[35] P. F. Gath and A. J. Calise, "Optimization of launch vehicle ascent trajectories with path constraints and coast arcs," *J. Guid., Control, Dyn.*, vol. 24, no. 2, pp. 296–304, Mar. 2001.
[36] G. Farkas, "Unstable manifolds for RFDEs under discretization: The Euler method," *Comput. Math. Appl.*, vol. 42, nos. 8–9, pp. 1069–1081, Oct. 2001.
[37] T. Öner and O. R. Isik, "Accelerating convergence for Backward–Euler method by adding time relaxation term," *J. Modern Methods Numer. Math.*, vol. 5, no. 2, p. 1, Jan. 2014.
[38] M. Szmuk, T. P. Reynolds, and B. Açıkmeşe, "Successive convexification for real-time six-degree-of-freedom powered descent guidance with state-triggered constraints," *J. Guid., Control, Dyn.*, vol. 43, no. 8, pp. 1399–1413, Aug. 2020.
[39] A. Boyali, S. Thompson, and D. Wong, "Applications of successive convexification in autonomous vehicle planning and control," in *Proc. 4th Int. Conf. Autom., Control Robots (ICACR)*, Oct. 2020, pp. 88–95.
[40] M. Szmuk and B. Acikmese, "Successive convexification for 6-DoF Mars rocket powered landing with free-final-time," in *Proc. AIAA Guid., Navigat., Control Conf.*, Jan. 2018, p. 617.
[41] Y. Nie, O. Faqir, and E. C. Kerrigan, "ICLOCS2: Try this optimal control problem solver before you try the rest," in *Proc. UKACC 12th Int. Conf. Control (CONTROL)*, Sep. 2018, p. 336.
[42] J. Lofberg, "YALMIP: A toolbox for modeling and optimization in MATLAB," in *Proc. IEEE Int. Conf. Robot. Autom.*, Sep. 2005, pp. 284–289.
[43] J. Dahl and E. D. Andersen, "A primal-dual interior-point algorithm for nonsymmetric exponential-cone optimization," *Math. Program.*, pp. 1–22, Mar. 2021.
[44] R. Badenbroek and J. Dahl, "An algorithm for nonsymmetric conic optimization inspired by MOSEK," *Optim. Methods Softw.*, pp. 1–38, Feb. 2021.

CHENG HU received the B.E. and M.E. degrees from the College of Aerospace Science and Engineering, National University of Defense Technology, where he is currently pursuing the Ph.D. degree. His research interests include trajectory optimization and guidance onboard.

XIBIN BAI received the B.S. degree in space engineering and the M.S. and Ph.D. degrees in aeronautical and astronautical science and technology from the National University of Defense Technology, Changsha, China, in 2011, 2013, and 2018, respectively. Since 2019, he has been a Lecturer with the National University of Defense Technology. His research interests include inertial navigation and aircraft dynamics and control.

SHIFENG ZHANG received the Ph.D. degree in control theory and engineering from the National University of Defense Technology, Changsha, China, in 2000. Since 2010, he has been a Professor and a Ph.D. Supervisor with the College of Aeronautics and Astronautics, National University of Defense Technology. His research interests include aircraft overall design, fight dynamics, guidance and control, inertial navigation, and measurement and precision analysis.

HUABO YANG received the Ph.D. degree in aircraft design from the National University of Defense Technology, Changsha, China, in 2008. In 2013, he was appointed as an Associate Professor with the College of Aeronautics and Astronautics, National University of Defense Technology. His research interests include test and calibration of inertial measurement unit, fight dynamics, and guidance and control.