AIAA SciTech Forum
8–12 January 2018, Kissimmee, Florida
2018 AIAA Guidance, Navigation, and Control Conference

10.2514/6.2018-1871

A Pseudospectral-Convex Optimization Algorithm for Rocket Landing Guidance

Jinbo Wang[1] and Naigang Cui[2]
Harbin Institute of Technology, Harbin 150001, China

This paper presents an online trajectory optimization algorithm for the rocket landing guidance problem. By combining pseudospectral discretization and an improved successive convexification method, the precision and rapidness of the algorithm are considered simultaneously. To address precision, the pseudospectral discretization method with so-called "spectral accuracy" is adopted, and according to the characteristics of rocket powered landing flight, the unique feature of pseudospectral method can be utilized to build a more accurate optimization model. From the aspect of rapidness, the discrete optimization problem is transformed into a series of convex subproblems via lossless and successive convexification. The successive convexification algorithm is improved by using a dynamic trust-region updating strategy, thereby improving the convergence performance. Convergence analysis is presented to prove that any accumulation point generated by the improved successive convexification algorithm is a stationary point of the original problem. The effectiveness of the proposed algorithm is demonstrated by numerical experiments. With the high-precision optimized trajectory and fast computing speed, the algorithm has the potential to be implemented onboard for real-time applications.

I. Introduction

The reusable launch vehicle (RLV) is a valid tool for reducing the cost of accessing space and enjoys a long history of research and practice. With the successful landing and reuse of SpaceX's Falcon-9 and Blue Origin's New Shepard rocket, vertical takeoff vertical landing reusable launch vehicles (VTVL-RLV) have received increased attention in recent years [1,2]. Landing guidance is a crucial technique for VTVL-RLV. Guiding a rocket booster to return and land on a planet with atmosphere is not a trivial task: a large range of deceleration with limited control ability is required, various path constraints and narrow terminal constraints for precision soft landing should be followed, and the fuel cost should be minimized to enhance the carrying capacity and handle emergencies. Solving this problem with traditional reentry or ascent guidance methods is daunting since optimality must be considered under strong constraints. Motivated by the Computational Guidance and Control (CG&C) philosophy [3,4], in this paper, an online trajectory optimization algorithm for rocket landing guidance is proposed that is built on pseudospectral method and convex optimization theory.

With the rapid development of onboard computing technology and optimization theory/algorithm, online trajectory optimization is becoming an effective approach to solve aerospace guidance problems. Among the many trajectory optimization methods that focus on rapidness and online operation capability [5-9], convex optimization theory-based methods have great advantages in mathematical principles and calculation. Theoretically, given an optimization problem can be formulated in a convex form, it is considered to be solvable in polynomial time [10]. However, many aerospace applications are dominated by non-convex and nonlinear dynamics and constraints. Lossless convexification and successive convexification, which have been developed in recent years, are effective techniques for handling the non-convexity in optimization.

The concept of lossless convexification was proposed for the Mars soft landing guidance problem [11-14] and has been generalized to solve some kinds of general optimal control problem [15]. In lossless convexification, the original non-convex constraints are relaxed to be convex ones, and the equivalence of the two forms is proved by the maximum principle. Lossless convexification based guidance algorithms have been verified by flight experiments [16,17]. Successive convexification was proposed in [8,18-22] to convexify nonlinear terms in system dynamics and constraints, which is an iterative technique that repeatedly solves linearized subproblem until the solution converges.

[1] PhD candidate, Department of Astronautics Engineering, wangjinbo@hit.edu.cn.
[2] Professor, Department of Astronautics Engineering, Cui_Naigang@163.com.

The modeling and linearization strategy as well as the trust-region constraint are crucial design aspects to guarantee the convergence of successive convexification algorithm.

In addition to the solution algorithms, in terms of problem discretization, methods based on pseudospectral discretization can potentially be used for online trajectory optimization. One of the main advantages of pseudospectral method compared with other discretization methods is the ability to obtain a higher solution accuracy under a smaller computational burden [23-27].

In this paper, a three Degrees-of-Freedom (3-DoF) dynamics model of the 1st stage of VTVL-RLV that considers aerodynamic drag and Earth rotation is built. The modeling and discretization as well as convexification methods for online trajectory optimization of rocket landing guidance problem are studied. For the modeling and discretization, the unique feature of pseudospectral method is utilized to provide a more accurate solution, where the trajectory is divided into two phases and the ignition and terminal time of the flight are designed as special control variables. From the convexification aspect, lossless convexification is used to convexify the thrust magnitude constraint with a "shell-like" feasible region (see Sec. II for details), and an improved successive convexification (iSC) algorithm is proposed, which is equipped with a dynamic trust-region updating strategy. Adjusting the trust-region dynamically according to the iteration statu will benefit algorithm convergence. Additionally, a convergence proof of iSC is given to provide a theoretical assurance of the algorithm.

The first main contribution of this paper is the development of a pseudospectral-improved successive convexification (P-iSC) algorithm that solves the fuel-optimal rocket landing problem with improved solution accuracy and better convergence performance. The second main contribution is the convergence proof of the iSC algorithm.

The paper is organized as follows. Sec. II presents the formulation of rocket dynamics and relevant constraints. The pseudospectral-improved successive convexification algorithm for online trajectory optimization is proposed in Sec. III. In Sec. IV, a convergence proof of the improved successive convexification algorithm is performed. Two groups of numerical experiment are given in Sec. V to illustrate the effectiveness and performance of the proposed algorithm. Concluding remarks are made in Sec. VI.

II. Problem Formulation

In this section, the system dynamics and constraints to be followed by the 1st stage of VTVL-RLV (later referred to as "rocket") during the landing flight are formulated. Then, the fuel-optimal landing guidance problem is formulated as a trajectory optimization problem.

A. Dynamics and Constraints

From the perspective of the working states of the rocket engine, the landing flight is divided into "coast phase" and "powered phase" in this paper. Using the landing process of the 1st stage booster of SpaceX's Falcon-9 as an analogy, the flight phase studied here begins from the end of "supersonic retro propulsion burn" (second burn) [28] and ends at final touchdown.

Attitude control of the rocket is assumed to be ideal. The angle of attack is assumed to be zero or negligible, and since the rocket body is axisymmetric, the aerodynamic lift force is omitted. Considering the space scope of flight and the flight time, the spherical rotating Earth model is adopted, and second-order Earth rotation terms are omitted. The isothermal-barotropic atmosphere model is adopted to describe the air density.

In order to facilitate the description of the forces acting on the rocket, the rocket's body frame is re-defined as shown in Figure 1. The 3-DoF dynamics model of the rocket can be given as (derivation is omitted)

$$\begin{cases} \dot{V} = -P_x/m - D/m - \mu \cdot \sin\gamma/r^2 \\ \dot{\gamma} = P_y/(mV) + V\cos\gamma/r - \mu\cdot\cos\gamma/(Vr^2) + 2\Omega\sin\psi\cos\phi \\ \dot{\psi} = P_z/(mV\cos\gamma) + V\tan\phi\cos\gamma\sin\psi/r - 2\Omega(\cos\psi\tan\gamma\cos\phi - \sin\phi) \\ \dot{\phi} = V\cos\gamma\cos\psi/r \\ \dot{\theta} = V\cos\gamma\sin\psi/(r\cos\phi) \\ \dot{r} = V\sin\gamma \\ \dot{m} = -P_{\text{Total}}/(I_{sp}g_0) \end{cases} \tag{1}$$

where V is the velocity, γ the flight-path angle, ψ the heading angle, ϕ and θ the latitude and longitude, r the radial distance from the Earth center to the rocket, m the rocket mass. $P_i(i = x, y, z)$ are the engine thrust components in the body frame, $D = qS_{\text{ref}}C_D$ is the drag force, $q = \rho V^2/2$ is the dynamic pressure, $\rho = \rho_0 e^{-\kappa H}$ is the air density, S_{ref} is the reference area of the rocket, C_D is the drag coefficient, μ is the gravitational parameter, Ω is the Earth self-rotation rate, P_{Total} is the thrust magnitude and $P_{\mathrm{Total}} = \sqrt{P_x^2 + P_y^2 + P_z^2}$, I_{sp} is the specific impulse of the rocket engine, g_0 is the gravitational acceleration at sea level.

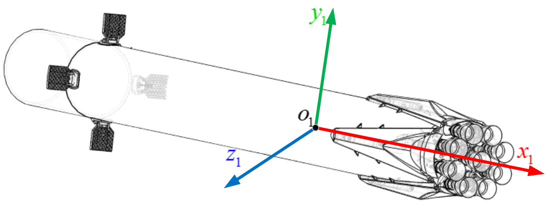

Figure 1 Rocket body frame

The initial time of the coast phase is set equal to 0, the initial time of the powered phase is defined as t_0 and the terminal time of the powered phase is defined as t_f . During the coast phase, the engine thrust is zero, and during the powered phase, the engine cannot be turned off after ignition for reliability, the thrust is thus bounded as $P_{\min} \le P_{\mathrm{Total}} \le P_{\max}$. The change rate of thrust components should also be constrained according to the physical limits of the actuators. The control constraints are

$$\begin{cases} P_x = P_y = P_z = 0 & , t \in [0, t_0] \\ P_{\min} \le \sqrt{P_x^2 + P_y^2 + P_z^2} \le P_{\max} & , t \in [t_0, t_f] \\ -\dot{P}_i^{\max} \le \dot{P}_i \le \dot{P}_i^{\max} , \ i = x, y, z & , t \in [t_0, t_f] \end{cases} \tag{2}$$

This second constraint in (2) leads to a non-convex “shell-like” feasible domain of the engine thrust components, which is illustrated in Figure 2.

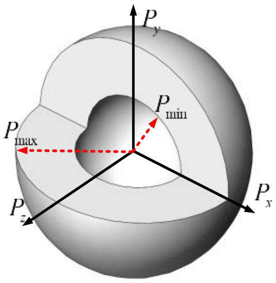

Figure 2 Non-convex thrust constraint

The state inequality constraints include constraints on heating rate $\dot{Q}$ and dynamic pressure q . Since the lift force is omitted in this paper, the normal load constraint is not considered. The state inequality constraints are

$$\begin{cases} \frac{1}{2}\rho V^2 \le q_{\max} \\ k_Q \sqrt{\rho} V^{3.15} \le \dot{Q}_{\max} \end{cases} \tag{3}$$

For precision soft landing, the constraints on the final states include

$$\begin{cases} 0 \le H_f \le \varepsilon_H \\ 0 \le V_f \le \varepsilon_V \\ -\varepsilon_\gamma \le \gamma_f \le \varepsilon_\gamma \\ \phi_d - \varepsilon_\phi \le \phi_f \le \phi_d + \varepsilon_\phi \\ \theta_d - \varepsilon_\theta \le \theta_f \le \theta_d + \varepsilon_\theta \\ m_f \ge m_{\mathrm{dry}} \end{cases} \tag{4}$$

where ε_H , ε_V , ε_γ , ε_ϕ , ε_θ are the tolerable error values of touchdown height, velocity, flight path angle, latitude and longitude, respectively, and m_{dry} is the dry mass of the rocket.

Remark 1 From the control constraints equation (2), it can be seen that there is no control effect during the coast phase, so it seems that the satisfaction of state constraints during the coast phase cannot be guaranteed. Nevertheless, from the analysis of the changing patterns of flight velocity and atmospheric density (flight height), as well as the trajectory simulation shown in Figure 3 and Figure 4, it can be seen that the heating rate and dynamic pressure are unimodal with respect to the flight time and the peak value occurs at the engine ignition time. So the satisfaction of state constraints for both flight phases can be determined by the engine ignition time, therefore it is essential to design the ignition time as an optimal variable. The optimization of the ignition time and the terminal time are discussed in detail in Sec. III.A.

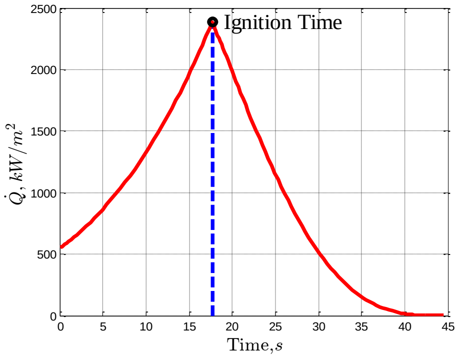

Figure 3 Heating rate vs. Flight time

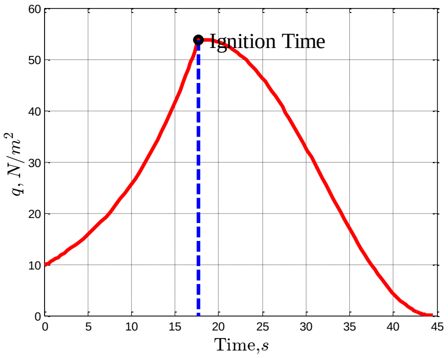

Figure 4 Dynamic pressure vs. Flight time

Remark 2 The dimensional dynamics model is exhibited in the manuscript for clarity, while non-dimensional dynamics model [29] is used in numerical experiments and simulations for the robustness and stability of the algorithms.

B. Trajectory Optimization Problem

The optimization objective is to minimize the fuel cost, so the terminal mass is set as the performance index

$$J = m(t_f)$$

The trajectory optimization problem can be formulated as

$$\begin{aligned} \text{CP0}: \quad & \min \ -m_f \\ & \text{s.t.} \ \ \text{eqs. (1)-(4)} \end{aligned}$$

Here we take the problem CP0 as the original problem, which is non-convex. The sources of non-convexity are the non-zero minimal thrust constraint and the nonlinear dynamics/state constraints. It should be noted that the problem formulated here is a free-time optimization problem.

III. Pseudospectral-Improved Successive Convexification Algorithm

In this section, the continuous nonlinear optimal control problem CP0 is transformed into a series of discrete second-ordered cone programming (SOCP) subproblem that can be solved efficiently by using primal-dual interior point methods (IPMs) [30] in polynomial time. First, CP0 is discretized by the flip-Radau pseudospectral method, and a high-precision discrete problem DP0 is built. Then, the lossless convexification technique is adopted to convexify the thrust constraint. Finally, an improved successive convexification algorithm is proposed to handle the nonlinear terms which are not in the scope of lossless convexification.

A. Pseudospectral Discretization

The guidance problem studied in [11-13] have some similarities to the problem addressed in this paper. However, above researches made several approximations during modeling the optimization problem and a simple zero-order-hold method is adopted in discretization, which cause a relatively large deviation between the discretized model used in optimization and the actual dynamics of the vehicle. Further, in [12,13], to convexify the problem, the terminal flight time is initially fixed and then searched for by an additional linear search procedure, so the free-time optimization problem is transformed into a series of fixed time problem. This approach leads to a relatively complex two-level algorithm structure, and the unimodality of fuel cost respect to the time of flight [12,13] may be unable to be met in increasingly complex problems, such as the one discussed in this paper. Furthermore, for reliability consideration, the rocket engine cannot be shut down once it is ignited, and there is a limit on the minimum thrust, so the coordinated design of ignition time and terminal time is necessary to obtain the fuel optimal trajectory. Using fixed time, the optimality of the ignition-terminal time combination cannot be guaranteed, so the optimality of the trajectory may be lost. Additionally, as suggested in Remark 1, the choice of ignition time is directly related to the satisfaction of the state constraints. In summary, the choice of discretization method and the design of ignition and terminal time are crucial issues.

To circumvent problems stated above and achieve high discrete accuracy with limited discrete points, the flip-Radau pseudospectral method is adopted to discretize CP0. A detailed discussion about the technical details of Radau pseudospectral method can be found in [23,27]. Utilizing the unique time domain mapping method of pseudospectral discretization is an important feature of this paper. With the time domain mapping, the ignition time and the terminal time can be designed as optimal variables, whereas the desired convexity is obtained by lossless and successive convexification.

The detailed discretization and optimization problem modeling are provided in the remainder of this section. A two-phase pseudospectral discretization is constructed to accommodate the coast phase and powered phase of the flight. The discretized equations of coast phase dynamics is as follows

$$\sum_{j=0}^{N_1}\boldsymbol{D}_{ij}\boldsymbol{x}_1(\tau_j)-\frac{t_f^1-t_0^1}{2}\boldsymbol{f}(\boldsymbol{x}_1(\tau_i),\boldsymbol{u}_1(\tau_i))=0,\left(i=1,\ldots,N_1\right) \tag{5}$$

where $\boldsymbol{D}$ is the flip-Radau pseudospectral differentiation matrix [23], $\boldsymbol{f}(\boldsymbol{x},\boldsymbol{u})$ is the right-hand side of the differential dynamics equation, $\tau_i,\left(i=1,\ldots,N_1\right)$ is the collocation points on the domain $(-1,1]$, N_1 is the number of the collocation points in coast phase, τ_0 is the discretization point at -1, $\boldsymbol{x}_1$ and $\boldsymbol{u}_1$ are state and control variables of coast phase, and $\boldsymbol{u}_1=\boldsymbol{0}$ since the engine dose not work in coast phase, t_0^1 and t_f^1 are the initial and terminal time of coast phase respectively, and $t_0^1=0, t_f^1=t_0$ as we defined above.

Similarly, the discretized equations of powered phase dynamics is as follows

$$\sum_{j=0}^{N_2}\boldsymbol{D}_{ij}\boldsymbol{x}_2(\tau_j)-\frac{t_f-t_0}{2}\boldsymbol{f}(\boldsymbol{x}_2(\tau_i),\boldsymbol{u}_2(\tau_i))=0,\left(i=1,\ldots,N_2\right) \tag{6}$$

The phase-connect (i.e., linkage) constraint is as follows

$$\boldsymbol{x}_{1\_f}=\boldsymbol{x}_{2\_0} \tag{7}$$

Since the collocation points of pseudospectral method are the roots of orthogonal polynomials defined on the domain $[-1,1]$, the time domain $[t_0,t_f]$ is first mapped onto $[-1,1]$ in the discretization as

$$\boldsymbol{\tau}=\frac{2\cdot\boldsymbol{t}-t_f-t_0}{t_f-t_0},\boldsymbol{t}\in[t_0,t_f] \tag{8}$$

where $\boldsymbol{\tau}\in[-1,1]$ is the vector of discretization points. After discretization, the ignition time t_0 and terminal time t_f appear in the discretized algebraic constraints (5) and (6), thus, they are now optimal variables together with state and control variables. Hence, the use of pseudospectral method not only improves the accuracy of discretization but also makes the free-time problem easy to deal with, and the optimality of the time values can be guaranteed.

For convenience, the right-hand side of the differential dynamics equations are augmented by the "special control variables" t_0 and t_f as follows

$$\boldsymbol{f}_{\text{aug}}(\boldsymbol{y}_i)=(t_0-t_f)\cdot\boldsymbol{f}(\boldsymbol{x}(\tau_i),\boldsymbol{u}_{\text{aug}}(\tau_i)) \tag{9}$$

where $\boldsymbol{y}_i=\{\boldsymbol{x}(\tau_i),\boldsymbol{u}_{\text{aug}}(\tau_i)\}$ and $\boldsymbol{u}_{\text{aug}}=\left\{t_0,t_f,P_x,P_y,P_z\right\}$. Then, the discretized algebraic constraints can be expressed as

$$2\sum_{j=0}^{N}\boldsymbol{D}_{ij}\boldsymbol{x}(\tau_j)+\boldsymbol{f}_{\text{aug}}(\boldsymbol{y}_i)=0,\left(i=1,\ldots,N,\ N=N_1,N_2\right) \tag{10}$$

As stated in Remark 1, only the state inequality constraints in the powered phase are considered. The discretized form of the state inequality constraints and control constraints are as follows

$$\begin{cases}\dfrac{1}{2}\rho_i V_i^2\le q_{\max}\\ k_Q\sqrt{\rho_i}V_i^{3.15}\le\dot{Q}_{\max}\end{cases},\left(i=1,\ldots,N_2\right) \tag{11}$$

$$\begin{cases}P_{\min}^2\le P_x^2(\tau_i)+P_y^2(\tau_i)+P_z^2(\tau_i)\le P_{\max}^2\ ,\left(i=1,\ldots,N_2\right)\\ -\dot{P}_j^{\max}\le\dot{P}_j(\tau_i)\le\dot{P}_j^{\max}\ ,\left(i=1,\ldots,N_2,\quad j=x,y,z\right)\end{cases} \tag{12}$$

In summary, the discretized nonlinear non-convex fuel-optimal problem DP0 can be rewritten as

$$
\begin{aligned}
\text{DP0}: \ & \min \ -m_f \\
& \text{s.t.} \ \ \text{eqs. (5)-(7),(11)-(12),(4)} \\
& \qquad t_0 < t_f
\end{aligned}
\tag{13}
$$

Two types of non-convexity exist in DP0: one is the thrust constraint with a "shell-like" feasible region, and the other is the nonlinearity in dynamics and state inequality constraints. They are handled by lossless convexification and successive convexification, respectively, in the following of this section.

B. Lossless Convexification of the Thrust Constraint

In lossless convexification, the non-convex problem is relaxed as an equivalent convex problem at the expense of increasing problem dimension, and it can be proved by maximum principle that the optimal solution of relaxed problem is still within the feasible domain of original problem.

In this subsection, using the results in [14], a slack variable Γ is introduced to relax the non-convexity of the minimal thrust constraint. The non-convex control constraint in (12) are transformed into the following convex constraints

$$P_x^2(\tau_i)+P_y^2(\tau_i)+P_z^2(\tau_i) \le \Gamma^2(\tau_i),\ i=1,\ldots,N_2 \tag{14}$$

$$P_{\min} \le \Gamma(\tau_i) \le P_{\max},\ i=1,\ldots,N_2 \tag{15}$$

and the mass dynamics is transformed to

$$\dot{m} = -\Gamma/(I_{sp}g_0) \tag{16}$$

In this transformation, equation (14) is standard second-order cone constraint, and (15) is linear constraint. The optimization problem using the relaxed constraints (14)-(16) instead of the original constraint is referred to as P1. The lossless-ness proof of the above transformation can be found in [14,15].

C. Improved Successive Convexification Algorithm

Lossless convexification can be used to convexify non-convex constraints without losing precision, unfortunately, only a few types of constraint can be convexified by lossless convexification. The nonlinear dynamics and constraints are the primary factors for non-convexity in P1 and they are not in the scope of lossless convexification. In this subsection, an improved successive convexification algorithm is proposed to circumvent the non-convexity from nonlinearity. The idea behind successive convexification is to linearize nonlinear constraints and solve the approximate convex problem iteratively, and measures are taken to ensure the approximate solutions generated by the algorithm converge to the solution of the original problem. After the linearization process, the algorithm solves a convex subproblem in each iteration, and the optimal solution obtained by each iteration is used as the reference trajectory for the next iteration until the solution converges. Successive convexification has been studied extensively in aerospace trajectory optimization and guidance [18-22]. Compared with existing researches, one of the main contributions of this paper is to improve the convergence performance of the successive convexification algorithm by using a penalty form trust-region constraint and a dynamic trust-region updating strategy based on the estimation of linearization error.

The iSC algorithm is described in detail in this subsection. To convexify the problem and provide a basis for the convergence proof of the algorithm in the subsequent section, several equivalent optimization problems are formulated. After pseudospectral discretization, the rocket dynamics constraints can be expressed as general equality constraints. For simplicity, the control variable $\boldsymbol{u}_{\text{aug}}(t)$ and state variable $\boldsymbol{x}(t)$ are denoted together as $\boldsymbol{x}$, thus, the nonlinear problem P1 can be rewritten as

$$
\begin{aligned}
\text{P1:} \ \ & \min \ \ f(\boldsymbol{x}) \\
& \text{s.t.} \ \ \boldsymbol{h}(\boldsymbol{x}) = 0 \\
& \qquad \boldsymbol{g}(\boldsymbol{x}) \le 0 \\
& \qquad \boldsymbol{s_c}(\boldsymbol{x}) \le 0
\end{aligned}
\tag{17}
$$

where $f(\boldsymbol{x}): \mathbb{R}^n \to \mathbb{R}$ is the performance index function whose specific form is not limited. It can be Meyer type, Lagrangian type, or Bolza type, provided that it can be represented by a nonlinear function after discretization (so the proposed algorithm is not restricted to the minimal fuel problem). $\boldsymbol{h}(\boldsymbol{x}): \mathbb{R}^n \to \mathbb{R}^n$ is the equality constraints including the dynamics constraints, $\boldsymbol{g}(\boldsymbol{x}): \mathbb{R}^n \to \mathbb{R}^m$ is the inequality constraints including the state path constraints, $\boldsymbol{s_c}(\boldsymbol{x}): \mathbb{R}^n \to \mathbb{R}^p$ is the second-ordered cone constraints including the losslessly convexified thrust constraint. We assume that the above mentioned functions are differentiable.

The equivalent penalized problem is solved in the algorithm. The penalized problem of P1 is as follows

$$\begin{aligned} &\text{P2: min} \quad p_\alpha(\boldsymbol{x}) := f(\boldsymbol{x}) + \alpha_h \sum_{i=1}^{n} \left| h_i(\boldsymbol{x}) \right| + \alpha_g \sum_{i=1}^{m} \max\left\{0, g_i(\boldsymbol{x})\right\} \\ &\qquad \text{s.t.} \ \ \boldsymbol{s_c}(\boldsymbol{x}) \le 0 \end{aligned} \tag{18}$$

On the basis of P2, relaxation variables are introduced, and the equivalent relaxed penalized problem P3 is acquired

$$\begin{aligned} \text{P3:min} \quad & f(\boldsymbol{x}) + \alpha_h \sum_{i=1}^{n} \xi_{h,i} \cdot \xi_{h,i} + \alpha_g \sum_{i=1}^{m} \xi_{g,i} \\ \text{s.t.} \quad & \boldsymbol{h}(\boldsymbol{x}) = \boldsymbol{\xi}_h \\ & \boldsymbol{g}(\boldsymbol{x}) \le \boldsymbol{\xi}_g, \boldsymbol{\xi}_g \ge 0, \\ & \boldsymbol{s_c}(\boldsymbol{x}) \le 0 \end{aligned} \tag{19}$$

where $\boldsymbol{\xi}_h$ and $\boldsymbol{\xi}_g$ are relaxation variables, α_h and α_g are corresponding penalty parameters. It should be noted that the relaxation variables $\boldsymbol{\xi}_h$ have a certain physical meaning. For the linearized dynamics system, $\boldsymbol{\xi}_h$ can be seen as the “virtual controls” and can alleviate the “artificial infeasibility” at the beginning of iteration [20].

As shown in [31], if the penalty parameters are sufficiently large, P2 and P3 have the same solution as P1. This equivalent condition is easily guaranteed, so it is reasonable to study P2 and P3 instead of P1.

P2 and P3 are two equivalent problems. The performance index of P2 contains more abundant and direct information, which can reflect the influence of variable variation on the problem, whereas the performance index of P3 is differentiable. Thus, $p_\alpha(\boldsymbol{x})$ is used to construct the metric for the quality of linear approximation, and P3 is used to form the subproblem to be solved in each iteration.

Next, P2 and P3 are linearized. The first-order Taylor series expansion of $p_\alpha(\boldsymbol{x})$ about a given state history $\boldsymbol{x}^k$ is as follows

$$\Phi_\alpha(\boldsymbol{x}^k, \Delta\boldsymbol{x}) := f(\boldsymbol{x}^k) + \nabla f(\boldsymbol{x}^k) \cdot \Delta\boldsymbol{x} + \alpha_h \sum_{i=1}^{n} \left| h_i(\boldsymbol{x}^k) + \nabla h_i(\boldsymbol{x}^k) \cdot \Delta\boldsymbol{x} \right| + \alpha_g \sum_{i=1}^{m} \max\left\{0, g_i(\boldsymbol{x}^k) + \nabla g_i(\boldsymbol{x}^k) \cdot \Delta\boldsymbol{x}\right\} \tag{20}$$

Since the first-order Taylor series expansion is a good approximation of the nonlinear function only if the perturbation $\Delta\boldsymbol{x}$ is sufficiently small, a trust-region constraint is added in penalty form to the linearization of P3

$$\begin{aligned} \text{LP3: min} \quad & \frac{1}{2} c_k \Delta\boldsymbol{x}^k \cdot \Delta\boldsymbol{x}^k + f(\boldsymbol{x}^k) + \nabla f(\boldsymbol{x}^k) \cdot \Delta\boldsymbol{x}^k + \alpha_h \sum_{i=1}^{n} \xi_{h,i}^k \cdot \xi_{h,i}^k + \alpha_g \sum_{i=1}^{m} \xi_{g,i}^k \\ \text{s.t.} \quad & \boldsymbol{h}(\boldsymbol{x}^k) + \nabla \boldsymbol{h}(\boldsymbol{x}^k) \cdot \Delta\boldsymbol{x}^k = \boldsymbol{\xi}_h^k \\ & \boldsymbol{g}(\boldsymbol{x}^k) + \nabla \boldsymbol{g}(\boldsymbol{x}^k) \cdot \Delta\boldsymbol{x}^k \le \boldsymbol{\xi}_g^k, \boldsymbol{\xi}_g^k \ge 0, \\ & \boldsymbol{s_c}(\boldsymbol{x}^k + \Delta\boldsymbol{x}^k) \le 0 \end{aligned} \tag{21}$$

where $\Delta\boldsymbol{x}^k$, $\boldsymbol{\xi}_h^k$ and $\boldsymbol{\xi}_g^k$ are optimal variables in the k th iteration, $c_k \Delta\boldsymbol{x}^k \cdot \Delta\boldsymbol{x}^k / 2$ is the trust-region constraint in penalty form, c_k is the penalty paremeter. When LP3 is used as the subproblem of the k th iteration of the algorithm, the reference trajectory of linearization is taken as the optimal trajectory of the $(k-1)$ th iteration.

For the fuel-optimal problem studied in this paper, we can obtain (the subscript indicating the discretization points are omitted for simplicity)

$$\begin{aligned} \text{LP3: min} \quad & \frac{1}{2} c_k^x \Delta\boldsymbol{x}^k \cdot \Delta\boldsymbol{x}^k + \frac{1}{2} c_k^u \Delta\boldsymbol{u}_{\text{aug}}^k \cdot \Delta\boldsymbol{u}_{\text{aug}}^k + m_f + \alpha_h \sum_{i=1}^{n} \xi_{h,i}^k \cdot \xi_{h,i}^k + \alpha_g \sum_{i=1}^{m} \xi_{g,i}^k \\ \text{s.t.} \quad & 2\boldsymbol{D} \cdot \boldsymbol{x}^k + \boldsymbol{f}_{\text{aug}}(\boldsymbol{x}^k, \boldsymbol{u}_{\text{aug}}^k) + \boldsymbol{A}_i(\boldsymbol{x}^k, \boldsymbol{u}_{\text{aug}}^k)\Delta\boldsymbol{x}^k + \boldsymbol{B}_i(\boldsymbol{x}^k, \boldsymbol{u}_{\text{aug}}^k)\Delta\boldsymbol{u}_{\text{aug}}^k + \boldsymbol{h}_i^\omega(\boldsymbol{x}^k) = \boldsymbol{\xi}_h^k \\ & q(\boldsymbol{r}^k, \boldsymbol{V}^k) + \boldsymbol{c}_{11}\Delta\boldsymbol{V}^k + \boldsymbol{c}_{16}\Delta\boldsymbol{r}^k - q_{\max} \le \boldsymbol{\xi}_{g\_q}^k, \quad \boldsymbol{\xi}_{g\_q}^k \ge 0 \\ & \dot{Q}(\boldsymbol{r}^k, \boldsymbol{V}^k) + \boldsymbol{c}_{21}\Delta\boldsymbol{V}^k + \boldsymbol{c}_{26}\Delta\boldsymbol{r}^k - \dot{Q}_{\max} \le \boldsymbol{\xi}_{g\_\dot{Q}}^k, \quad \boldsymbol{\xi}_{g\_\dot{Q}}^k \ge 0 \\ & (\boldsymbol{P}_x^k + \Delta\boldsymbol{P}_x^k)^2 + (\boldsymbol{P}_y^k + \Delta\boldsymbol{P}_y^k)^2 + (\boldsymbol{P}_z^k + \Delta\boldsymbol{P}_z^k)^2 \le (\boldsymbol{\Gamma}^k + \Delta\boldsymbol{\Gamma}^k)^2 \\ & -\dot{P}_i^{\max} \le \dot{\boldsymbol{P}}_i^k + \Delta\dot{\boldsymbol{P}}_i^k \le \dot{P}_i^{\max} \ ,\left(i = x, y, z\right) \\ & P_{\min} \le \boldsymbol{\Gamma}^k + \Delta\boldsymbol{\Gamma}^k \le P_{\max} \\ & t_0^k + \Delta t_0^k < t_f^k + \Delta t_f^k \\ & \boldsymbol{x}_f^k + \Delta\boldsymbol{x}_f^k \in \Psi_f \end{aligned} \tag{22}$$

where $\boldsymbol{h}^\omega(\boldsymbol{x})$ contains terms related to the rotation of the Earth, Ψ_f is the feasible terminal set as described in (4), $\boldsymbol{A} = \partial\boldsymbol{f}_{\text{aug}} / \partial\boldsymbol{x}$ and $\boldsymbol{B} = \partial\boldsymbol{f}_{\text{aug}} / \partial\boldsymbol{u}$ are the linearized state matrix and control matrix, and we have

$$\boldsymbol{A}(\boldsymbol{x}^k,\boldsymbol{u}_{\text{aug}}^k)=\begin{bmatrix} a_{11} & a_{12} & 0 & 0 & 0 & a_{16} & a_{17} \\ a_{21} & a_{22} & 0 & 0 & 0 & a_{26} & a_{27} \\ a_{31} & a_{32} & a_{33} & a_{34} & 0 & a_{36} & a_{37} \\ a_{41} & a_{42} & a_{43} & 0 & 0 & a_{46} & 0 \\ a_{51} & a_{52} & a_{53} & a_{54} & 0 & a_{56} & 0 \\ a_{61} & a_{62} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}_k \quad \boldsymbol{B}(\boldsymbol{x}^k,\boldsymbol{u}_{\text{aug}}^k)=\begin{bmatrix} b_{11} & b_{12} & 0 & b_{14} & 0 & 0 \\ b_{21} & b_{22} & 0 & 0 & b_{25} & 0 \\ b_{31} & b_{32} & 0 & 0 & 0 & b_{36} \\ b_{41} & b_{42} & 0 & 0 & 0 & 0 \\ b_{51} & b_{52} & 0 & 0 & 0 & 0 \\ b_{61} & b_{62} & 0 & 0 & 0 & 0 \\ b_{71} & b_{72} & b_{73} & 0 & 0 & 0 \end{bmatrix}_k \tag{23}$$

The elements of the matrix $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{c}_{ij}$ are shown in the Appendix.

Based on LP3, the improved successive convexification algorithm is as follows

Algorithm 1 Improved Successive Convexification Algorithm

```
Input  Initial variables $x^0$; initial trust-region parameter $c_0$, trust-region parameter bounds
       $c_{\max} \ge c_{\min} > 0$; trust-region updating parameters $0<\rho_1<\rho_2<1, 0<\sigma_1<1<\sigma_2$; penalty
       parameters $\alpha_h, \alpha_g$; termination criterion parameters $\varepsilon_x>0, \varepsilon_h>0, \varepsilon_g>0$; maximal iteration time $k_{\max}$.
Step 1 Solve subproblem LP3 and obtain $\Delta x^k$ and $\xi_h^k$, $\xi_g^k$
    if $|\Delta x^k|<\varepsilon_x, |\xi_h^k|<\varepsilon_h, |\xi_g^k|<\varepsilon_g$   STOP, return success and $x^k$
    else if $k>k_{\max}$                  STOP, return fail
    else                                  go to Step 2
Step 2 Compute the ratio:
        $r_k := \frac{p_\alpha(x^k)-p_\alpha(x^k+\Delta x^k)}{p_\alpha(x^k)-\Phi_\alpha(x^k,\Delta x^k)}$        (24)
    if $r_k \ge \rho_1$      the $k$ th iteration is successful, set $x^{k+1}:=x^k+\Delta x^k$, go to Step 3
    else                     the $k$ th iteration is unacceptable, set $x^{k+1}:=x^k$, go to Step 3
Step 3 Update trust-region parameter:
    if $r_k<\rho_1$                    set $c_{k+1}=\sigma_2 c_k$, $k \leftarrow k+1$, go to Step 1
    else if $r_k \in [\rho_1,\rho_2)$  set $c_{k+1}=\text{mid}(c_{\min},c_k,c_{\max})$, $k \leftarrow k+1$, go to Step 1
    else if $r_k \ge \rho_2$           set $c_{k+1}=\text{mid}(c_{\min},\sigma_1 c_k,c_{\max})$, $k \leftarrow k+1$, go to Step 1
```

Now we give some explanations about Algorithm 1.

Remark 3 *Dynamic trust-region constraint in penalty form*. The trust-region is an important safeguard for convergence of the algorithm. With the classic form $|\boldsymbol{x}^{k+1}-\boldsymbol{x}^k|\le\boldsymbol{\delta}$ [18-20], the irrational design of constraint bounds will affect the convergence of the algorithm. If $\boldsymbol{\delta}$ is too small, a poor initial reference trajectory will lead to infeasibility, and if $\boldsymbol{\delta}$ is too large, the linearization error will hinder the convergence. To circumvent this problem, we use a penalty formed trust-region with a dynamic penalty parameter in LP3 instead of the classic *hard* constraint. The penalty trust-region can be considered as a *soft* constraint, which reduces the sensitivity of the constraint satisfaction with respect to constraint bounds. Additionally, the trust-region is related to the linearization error explicitly in Step 3. If the linearization error is small, a larger trust-region (small penalty) is adopted to expand the solution space, and if the linearization error is large, the penalty parameter is magnified in the next iteration to restrain the linearization error.

Remark 4 *Metric of linearization error*. In Step 2, the ratio r_k is used as a metric for the linearization error of current iteration. As defined in P2, $p_\alpha(\boldsymbol{x})$ includes information about the performance index and the constraints, so $p_\alpha(\boldsymbol{x}^k)-p_\alpha(\boldsymbol{x}^k+\Delta\boldsymbol{x}^k)$ reflects the improvement of the optimality and the constraints satisfaction of the nonlinear problem with the optimal solution $\Delta\boldsymbol{x}^k$. And $p_\alpha(\boldsymbol{x}^k)-\Phi_\alpha(\boldsymbol{x}^k,\Delta\boldsymbol{x}^k)$ reflects the improvement of the linearized problem, and r_k closing to 1 indicates the solution of the linearized problem is a relatively good approximation. In Step 2, if r_k is too small, the current solution is rejected and the penalty parameter increases in the next iteration, and if r_k is close enough to 1, linearization is considered more reliable and the penalty parameter is decreased.

Remark 5 *Reformulation of subproblem*. The penalty trust-region $c_k\Delta\boldsymbol{x}\cdot\Delta\boldsymbol{x}/2$ is a second-order term in the performance index of LP3. For clarity, this formulation is maintained throughout the manuscript, whereas in the algorithm implementation, the quadratic convex programming problem are converted to second-ordered cone programming (SOCP) problem by variable substitution for better numerical properties [10].

The main feature of the iSC algorithm is the use of a dynamic trust-region constraint in penalty form, and numerical experiments show that the convergent performance can be improved with this strategy.

IV. Convergence of the Improved Successive Convexification Algorithm

Although the successive convexification algorithm has been applied to many aerospace and robotics applications, and the algorithm's performance is reported to be satisfactory, convergence proof of the algorithm is still an open challenge. Among studies of the convergence of successive linearization and successive approximation based methods, a proof considering linear dynamics with concave state inequality constraints is given in [19], Ref.[21] provides a proof that considers only the convexification of system dynamics while constraints are assumed to be convex already, a lucid proof of a successive linearization algorithm which focusing on nonlinear semidefinite programming problems is given in [32]. We follow the methodology of [21,32] and formulate the proof to consider the general nonlinear problem with second-ordered cone constraints. We discuss this problem from two perspectives. First, we show that Algorithm 1 is well defined. Then, the proof that states any accumulation point generated by Algorithm 1 is a stationary point of P3 is given. P3 is the penalty form of P1, and in [31], it is proven that P3 and P1 are equivalent if the penalty parameter is sufficiently large. We adopt this conclusion and focus on P3 in the proof.

A. Well-definiteness of Algorithm 1

In Step 2 of Algorithm 1, r_k is in fractional form. To show that Algorithm 1 is well defined, we need to prove that the denominator in r_k is positive as long as the termination condition is not reached, i.e. $\Delta\boldsymbol{x}^k\neq\mathbf{0}$. Furthermore, the termination criterion in Step 1 must be justified.

We begin with a useful lemma as follows

Lemma 1. Let $\boldsymbol{x}^k$ be the reference trajectory of the current iteration and $\Delta\boldsymbol{x}^k$ be the solution of LP3 then,

$$p_\alpha(\boldsymbol{x}^k)-\Phi_\alpha(\boldsymbol{x}^k,\Delta\boldsymbol{x}^k)\geq\frac{1}{2}c_k\Delta\boldsymbol{x}^k\cdot\Delta\boldsymbol{x}^k \tag{25}$$

Proof: From the definition of p_α and Φ_α, $p_\alpha(\boldsymbol{x}^k)=\Phi_\alpha(\boldsymbol{x}^k,\mathbf{0})$. $\Delta\boldsymbol{x}^k=\mathbf{0}$ is a feasible solution to LP3, and $\Delta\boldsymbol{x}^k$ is the optimal solution of the k th iteration, so we have

$$\frac{1}{2}c_k\Delta\boldsymbol{x}^k\cdot\Delta\boldsymbol{x}^k+\Phi_\alpha(\boldsymbol{x}^k,\Delta\boldsymbol{x}^k)\leq\frac{1}{2}c_k\mathbf{0}\cdot\mathbf{0}+\Phi_\alpha(\boldsymbol{x}^k,\mathbf{0})=p_\alpha(\boldsymbol{x}^k)$$

and the statement is proved.

From the lemma above, the denominator in r_k is always nonnegative, then we have the numerator $p_\alpha(\boldsymbol{x}^k)-p_\alpha(\boldsymbol{x}^k+\Delta\boldsymbol{x}^k)=p_\alpha(\boldsymbol{x}^k)-p_\alpha(\boldsymbol{x}^{k+1})\geq 0$ for all successful iterations, thus the sequence $\{p_\alpha(\boldsymbol{x}^k)\}$ is monotonically non-increasing.

Next, we indicate that $p_\alpha(\boldsymbol{x}^k)-\Phi_\alpha(\boldsymbol{x}^k,\Delta\boldsymbol{x}^k)=0$ if and only if $\Delta\boldsymbol{x}^k=\mathbf{0}$, so Algorithm 1 is well defined.

Lemma 2. Let $\boldsymbol{x}^k$ be the reference trajectory of the current iteration and $\Delta\boldsymbol{x}^k$ be the solution of LP3, then $p_\alpha(\boldsymbol{x}^k)-\Phi_\alpha(\boldsymbol{x}^k,\Delta\boldsymbol{x}^k)=\mathbf{0}$ if and only if $\Delta\boldsymbol{x}^k=\mathbf{0}$.

Proof: From the definition of Φ_α, we get $p_\alpha(\boldsymbol{x}^k)-\Phi_\alpha(\boldsymbol{x}^k,\Delta\boldsymbol{x}^k)=0$ if $\Delta\boldsymbol{x}^k=\mathbf{0}$. Conversely, let $p_\alpha(\boldsymbol{x}^k)-\Phi_\alpha(\boldsymbol{x}^k,\Delta\boldsymbol{x}^k)=0$, Lemma 1 implies $c_k\Delta\boldsymbol{x}^k\cdot\Delta\boldsymbol{x}^k=0$ and hence, $\Delta\boldsymbol{x}^k=\mathbf{0}$. The proof is now completed.

Next, a justification is provided for the termination criterion in Step 1, which states that the current iterate $\boldsymbol{x}^k$ is a stationary point of P3 if and only if $\Delta\boldsymbol{x}^k=\mathbf{0}$, and $\boldsymbol{\xi}_h=\boldsymbol{\xi}_g=\mathbf{0}$ indicates that the solution of P3 is identical the solution of P1. To this end, we first study the KKT conditions of P3 and LP3.

Since LP3 is convex, the problem is equivalent to its KKT conditions, thus $(\Delta\boldsymbol{x}^k,\boldsymbol{\xi}_h,\boldsymbol{\xi}_g)$ is a solution of LP3 if and only if there exist Lagrange multipliers $(\boldsymbol{\rho}^k,\boldsymbol{\mu}^k,\boldsymbol{\lambda}^k,\boldsymbol{\upsilon}^k)\in\mathbb{R}^n\times\mathbb{R}^m\times\mathbb{R}^m\times\mathbb{R}^p$ such that the following KKT conditions hold

$$\begin{aligned}
&c_k\Delta\boldsymbol{x}^k+\nabla f(\boldsymbol{x}^k)+\sum_{i=1}^{n}\rho_i^k\nabla h_i(\boldsymbol{x}^k)+\sum_{i=1}^{m}\lambda_i^k\nabla g_i(\boldsymbol{x}^k)+\sum_{i=1}^{p}\upsilon_i^k\nabla s_{c,i}(\boldsymbol{x}^k+\Delta\boldsymbol{x}^k)=0\\
&2\alpha_h\xi_{h,i}^k-\rho_i^k=0,\quad i=1,\ldots,n\\
&\alpha_g-\mu_i^k-\lambda_i^k=0,\quad i=1,\ldots,m\\
&h_i(\boldsymbol{x}^k)+\nabla h_i(\boldsymbol{x}^k)\cdot\Delta\boldsymbol{x}^k=\xi_{h,i}^k,\qquad \rho_i^k(h_i(\boldsymbol{x}^k)+\nabla h_i(\boldsymbol{x}^k)\cdot\Delta\boldsymbol{x}^k-\xi_{h,i}^k)=0,\quad i=1,\ldots,n\\
&\xi_{g,i}^k\geq 0,\quad \mu_i^k\geq 0,\quad \xi_{g,i}^k\mu_i^k=0,\quad i=1,\ldots.,m\\
&\lambda_i^k\geq 0,\quad g_i(\boldsymbol{x})+\nabla g_i(\boldsymbol{x}^k)\cdot\Delta\boldsymbol{x}^k-\xi_{g,i}^k\leq 0,\quad \lambda_i^k(g_i(\boldsymbol{x})+\nabla g_i(\boldsymbol{x}^k)\cdot\Delta\boldsymbol{x}^k-\xi_{g,i}^k)=0,\quad i=1,\ldots.,m\\
&\upsilon_i^k\geq 0,\quad s_{c,i}(\boldsymbol{x}^k+\Delta\boldsymbol{x}^k)\leq 0,\quad \upsilon_i^k\cdot s_{c,i}(\boldsymbol{x}^k+\Delta\boldsymbol{x}^k)=0,\quad i=1,\ldots.,p
\end{aligned}\tag{26}$$

While the KKT condition of P3 is as follows

$$\begin{aligned}
&\nabla f(\boldsymbol{x}^*)+\sum_{i=1}^{n}\rho_i^*\nabla h_i(\boldsymbol{x}^*)+\sum_{i=1}^{m}\lambda_i^*\nabla g_i(\boldsymbol{x}^*)+\sum_{i=1}^{p}\upsilon_i^*\nabla s_{c,i}(\boldsymbol{x}^*)=0\\
&2\alpha_h\xi_{h,i}^*-\rho_i^*=0,\quad i=1,\ldots,n\\
&\alpha_g-\mu_i^*-\lambda_i^*=0,\quad i=1,\ldots,m\\
&h_i(\boldsymbol{x}^*)=\xi_{h,i}^*,\qquad \rho_i^*(h_i(\boldsymbol{x}^*)-\xi_{h,i}^*)=0,\quad i=1,\ldots,n\\
&\xi_{g,i}^*\geq 0,\quad \mu_i^*\geq 0,\quad \xi_{g,i}^*\mu_i^*=0,\quad i=1,\ldots.,m\\
&\lambda_i^*\geq 0,\quad g_i(\boldsymbol{x}^*)-\xi_{g,i}^*\leq 0,\quad \lambda_i^*(g_i(\boldsymbol{x}^*)-\xi_{g,i}^*)=0,\quad i=1,\ldots.,m\\
&\upsilon_i^*\geq 0,\quad s_{c,i}(\boldsymbol{x}^*)\leq 0,\quad \upsilon_i^*\cdot s_{c,i}(\boldsymbol{x}^*)=0,\quad i=1,\ldots.,p
\end{aligned}\tag{27}$$

where $(\boldsymbol{\rho}^*,\boldsymbol{\mu}^*,\boldsymbol{\lambda}^*,\boldsymbol{\upsilon}^*)\in\mathbb{R}^n\times\mathbb{R}^m\times\mathbb{R}^m\times\mathbb{R}^p$ are Lagrange multipliers and $\boldsymbol{x}^*$ is a stationary point of P3.

Comparing (26) and (27), it is obvious that they are identical when $\Delta\boldsymbol{x}^k=\boldsymbol{0}$. This observation can be summarized as follows

Proposition 1. The current iteration $\boldsymbol{x}^k$ is a stationary point of P3 if and only if $\Delta\boldsymbol{x}^k=\boldsymbol{0}$ is the solution of the subproblem LP3.

Proof: The statement follows from the preceding arguments.

Furthermore, it is evident that if $\boldsymbol{\xi}_h=\boldsymbol{\xi}_g=\boldsymbol{0}$, the solution of the penalized problem P3 and the original problem P1 are identical, together with Proposition 1, the termination criterion of Algorithm 1 is justified.

Remark 6 In general numerical conditions, Algorithm 1 will produce an infinite iterative sequence, so in implementation, the termination criterion of Algorithm 1 is set to be $|\Delta\boldsymbol{x}|<\varepsilon_x,|\boldsymbol{\xi}_h|<\varepsilon_h,|\boldsymbol{\xi}_g|<\varepsilon_g$.

B. Convergence of Algorithm 1

We assume that Algorithm 1 generates an infinite sequence $\{\boldsymbol{x}^k\}$ in the sequel and will prove that any accumulation point of $\{\boldsymbol{x}^k\}$ is a stationary point of P3.

First, since the feasible domain of LP3 in Algorithm 1 is convex and compact, for the infinite sequence $\{\boldsymbol{x}^k\}$, it is reasonable to assume that there is at least one subsequence $\{\boldsymbol{x}^k\}_{k\in K}\to\boldsymbol{x}^*$, i.e. that $\{\boldsymbol{x}^k\}$ has at least one accumulation point.

Assumption 1. Let $\{\boldsymbol{x}^k\}$ be a sequence generated by Algorithm 1, then $\{\boldsymbol{x}^k\}$ has accumulation points.

Next, we give three important lemmas that will be used in the proof of convergence theorem.

Lemma 3. Let $\{\boldsymbol{x}^k\}$ be a sequence generated by Algorithm 1 and $\{\boldsymbol{x}^k\}_{k\in K}$ be a subsequence converging to some $\boldsymbol{x}^*$ in such a way that $\{c_k\left\|\Delta\boldsymbol{x}^k\right\|\}_{k\in K}\to 0$. Then, $\boldsymbol{x}^*$ is a stationary point of P3.

Proof: The condition $\{c_k\left\|\Delta\boldsymbol{x}^k\right\|\}_{k\in K}\to 0$ and $c_k>c_{\min}$ implies that $\{\left\|\Delta\boldsymbol{x}^k\right\|\}_{k\in K}\to 0$. Then, similar to the process of the proof of Proposition 1, taking the limit $k\to\infty$ on the subsequence K in the KKT condition of LP3, we can obtain the KKT condition of P3. The proof is now completed.

The reason we set $\{c_k\left\|\Delta\boldsymbol{x}^k\right\|\}_{k\in K}\to 0$ instead of setting $\{\left\|\Delta\boldsymbol{x}^k\right\|\}_{k\in K}\to 0$ directly in Lemma 4 is that we can't assure $\{c_k\}_{k\in K}$ is bounded, and there is a term $c_k\Delta\boldsymbol{x}^k$ in the KKT condition of LP3, so we need $\{c_k\left\|\Delta\boldsymbol{x}^k\right\|\}_{k\in K}\to 0$ if we want $c_k\Delta\boldsymbol{x}^k$ to be vanished as $k\to\infty$.

Lemma 4. Let $\{\boldsymbol{x}^k\}$ be a sequence generated by Algorithm 1, and $\{\boldsymbol{x}^k\}_{k\in K}$ be a subsequence converging to some $\boldsymbol{x}^*$. If $\boldsymbol{x}^*$ is not a stationary point of P3, then

$$\limsup_{k\to\infty,k\in K} c_k < \infty$$

Proof: The proof is by contradiction. Assume

$$\lim_{k\to\infty,k\in K} c_k = \infty \tag{28}$$

then, according to the updating rule of c_k in Step 3, with a sufficiently large k, none of the iterations $k \in K$ are successful because otherwise, there will be $c_{k+1} \le c_{\max}$. Hence, we have $r_k < \rho_1$ and $\boldsymbol{x}^k = \boldsymbol{x}^{k+1}$ for all $k \in K$ that are sufficiently large.

Now, we show that $r_k \to 1$ as $k \to \infty, k \in K$, which leads to a contradiction of $r_k < \rho_1$.

First, we note that there is a constant $\vartheta > 0$ such that $c_k \|\Delta\boldsymbol{x}^k\| > \vartheta, k \in K$, otherwise, if $c_k \|\Delta\boldsymbol{x}^k\| \to 0$, then $\boldsymbol{x}^*$ is a stationary point of P3 according to Lemma 4, which is contradicts the assumption. Then, by Lemma 1, we have

$$p_\alpha(\boldsymbol{x}^k) - \Phi_\alpha(\boldsymbol{x}^k, \Delta\boldsymbol{x}^k) \ge \frac{1}{2} c_k \Delta\boldsymbol{x}^k \cdot \Delta\boldsymbol{x}^k = \frac{1}{2} c_k \|\Delta\boldsymbol{x}^k\|^2 \ge \frac{1}{2}\vartheta \|\Delta\boldsymbol{x}^k\|$$

It is further noted that $\{\|\Delta\boldsymbol{x}^k\|\}_{k\in K} \to 0$, otherwise, we will have $c_k \Delta\boldsymbol{x}^k \cdot \Delta\boldsymbol{x}^k \to \infty$ according to (28), which implies that the performance index of LP3 tends to infinity as $k \in K$. However, this cannot be true since the feasible $\Delta\boldsymbol{x}$=$\boldsymbol{0}$ would provide a better objective value. Considering this and according to the definition of p_α and Φ_α, Φ_α is the first order approximation of p_α, we have $|\Phi_\alpha(\boldsymbol{x}^k, \Delta\boldsymbol{x}^k) - p_\alpha(\boldsymbol{x}^k + \Delta\boldsymbol{x}^k)| = o(\|\Delta\boldsymbol{x}^k\|)$ as $\{\|\Delta\boldsymbol{x}^k\|\}_{k\in K} \to 0$.

Then, we have

$$|r_k - 1| = \left| \frac{p_\alpha(\boldsymbol{x}^k) - p_\alpha(\boldsymbol{x}^k + \Delta\boldsymbol{x}^k)}{p_\alpha(\boldsymbol{x}^k) - \Phi_\alpha(\boldsymbol{x}^k, \Delta\boldsymbol{x}^k)} - 1 \right| = \left| \frac{\Phi_\alpha(\boldsymbol{x}^k, \Delta\boldsymbol{x}^k) - p_\alpha(\boldsymbol{x}^k + \Delta\boldsymbol{x}^k)}{p_\alpha(\boldsymbol{x}^k) - \Phi_\alpha(\boldsymbol{x}^k, \Delta\boldsymbol{x}^k)} \right| \le \frac{o(\|\Delta\boldsymbol{x}^k\|)}{\frac{1}{2}\vartheta\|\Delta\boldsymbol{x}^k\|} \to 0$$

as $k \to \infty, k \in K$. It is in contradiction to the assumption. The proof is now completed.

Lemma 5. Let $\{\boldsymbol{x}^k\}$ be a sequence generated by Algorithm 1, then, there are infinite successful iterations.

Proof: Assume that there is a k_0 such that the iterations are unsuccessful for all $k > k_0$ then, we have $c_k \to \infty$ and $\boldsymbol{x}^k$=$\boldsymbol{x}^{k_0}$ as $k > k_0$. Since $\boldsymbol{x}^{k_0}$ is not a stationary point of P3, or the algorithm would have stopped, then $c_k \to \infty$ is in contradiction to Lemma 4. The proof is now completed.

Now, we provide the main convergence result of Algorithm 1.

Proposition 2. Let $\{\boldsymbol{x}^k\}$ be a sequence generated by Algorithm 1. Then, any accumulation point of it is a stationary point of P3.

Proof: Let $\boldsymbol{x}^*$ be an accumulation point of $\{\boldsymbol{x}^k\}_{k\in K}$. According to Lemma 5, we assume without loss of generality that all iterations $k \in K$ are successful.

Assume $\boldsymbol{x}^*$ is not a stationary point of P3, then Lemma 4 implies

$$\limsup_{k\to\infty,k\in K} c_k < \infty$$

Therefore, there is a constant $\chi > 0$ such that $c_k < \chi$ as $k \in K$. We have $r_k \ge \rho_1$ for successful iterations, so we can obtain from the definition of r_k and Lemma 1

$$p_\alpha(\boldsymbol{x}^k) - p_\alpha(\boldsymbol{x}^{k+1}) \ge \rho_1(p_\alpha(\boldsymbol{x}^k) - \Phi_\alpha(\boldsymbol{x}^k, \Delta\boldsymbol{x}^k)) \ge \frac{1}{2}\rho_1 c_k \Delta\boldsymbol{x}^k \cdot \Delta\boldsymbol{x}^k \ge \frac{1}{2}\rho_1 c_{\min} \Delta\boldsymbol{x}^k \cdot \Delta\boldsymbol{x}^k \tag{29}$$

where $\{p_\alpha(\boldsymbol{x}^k)\}$ is monotonically non-increasing for successful iterations, so we have $p_\alpha(\boldsymbol{x}^k) - p_\alpha(\boldsymbol{x}^{k+1}) \to 0$ as $k \to \infty$,then, we obtain $\{\|\Delta\boldsymbol{x}^k\|\}_{k\in K} \to 0$ from (29), and since $c_k < \chi$, we have $\{c_k \|\Delta\boldsymbol{x}^k\|\}_{k\in K} \to 0$, which implies $\boldsymbol{x}^*$ is a stationary point of P3 according to Lemma 3, which contradicts the assumption. Hence, we state that any accumulation point $\boldsymbol{x}^*$ is a stationary point of P3. The proof is now completed.

V. Numerical Demonstration

To demonstrate the performance of the Pseudospectral-Improved Successive Convexification algorithm described in this paper, two groups of numerical experiment are performed. The purpose of the first group of experiment is to verify the effectiveness, precision, and computational efficiency of the P-iSC algorithm. The numerical results are compared with the solutions obtained by GPOPS-II [33], which is a general-purpose MATLAB software program. The second group verifies the improvement of convergence performance compared with the classic successive convexification algorithm using fixed trust-region constraint. All programs are coded using MATLAB 2012a, and the MOSEK [34] optimizer API for C is invoked to solve the SOCP subproblems. All numerical optimizations and simulations in this paper are performed on a laptop with Intel Core i7-7700HQ 2.80GHz CPU.

The rocket parameters used in the numerical experiments are shown in Table 1.

Table 1 Rocket parameters

Variable	Value
Total mass (kg)	35000
Dry mass (kg)	25000
Thrust range (kN)	[400, 850]
Specific impulse (s)	250
Drag coefficient	0.25
Reference area (m^2)	8.0

The initial values and terminal constraints of the trajectory are shown in Table 2. For clarity, the longitude and latitude values are transformed into local coordinates.

Table 2 Initial values and terminal constraints of the trajectory

Variable	Initial value	Terminal constraint
$V\ (m/s)$	400	[0, 0.2]
$\gamma\ (^\circ)$	-70	[-90.5, -89.5]
$\psi\ (^\circ)$	50	[-100, 100]
$X\ (m)$	-2400	[-0.5, 0.5]
$Y\ (m)$	-2800	[-0.5, 0.5]
$H\ (m)$	15000	[0, 0.2]

A. Effectiveness of the P-iSC algorithm

In this subsection, a comparison between P-iSC and GPOPS-II is performed. GPOPS-II is an optimization software package that uses the hp-adaptive Radau pseudospectral method and the SNOPT/IPOPT solver. GPOPS-II has been tested in a wide range of problems and it can be viewed as a typical representative of optimization softwares based on nonlinear programming (NLP). GPOPS-II is used to examine the effectiveness of the P-iSC algorithm.

The trajectory profiles shown in Figures 5-13 and the optimal solutions/CPU time summarized in Table 3 are the results comparison between P-iSC and GPOPS-II with the following settings. The collocation points in coast phase and powered phase are set to be 50 and 100, respectively, in P-iSC. The hp-adaptive mesh refinement is carried out in GPOPS-II. The termination criterion in Algorithm 1 are set to be $\varepsilon_x = 10^{-5}, \varepsilon_h = 10^{-5}, \varepsilon_g = 10^{-5}$ (non-dimensional), and the convergence tolerance of GPOPS-II is set to 10^{-5}. The SNOPT solver is used in GPOPS-II. Same initial trajectory values are used in P-iSC and GPOPS-II.

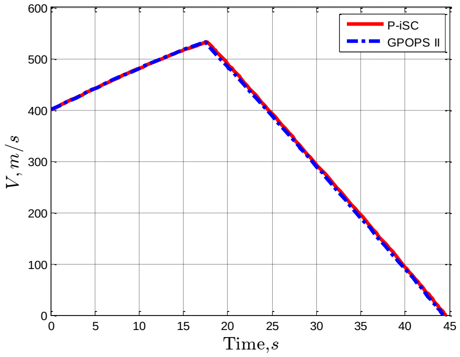

Figure 5 Comparison of velocity profiles

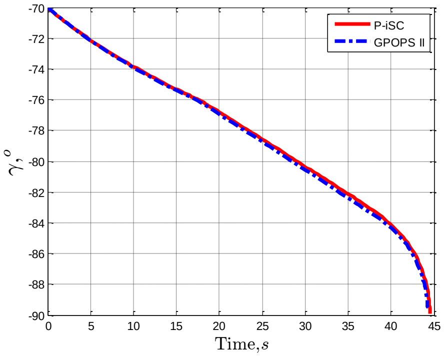

Figure 6 Comparison of flight path angle profiles

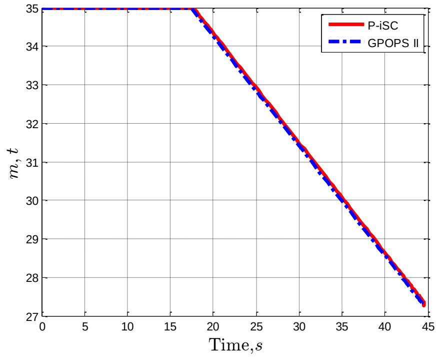

Figure 7 Comparison of mass profiles

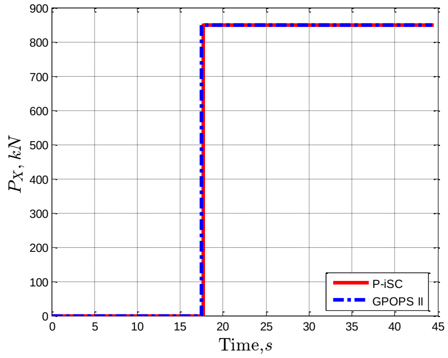

Figure 8 Comparison of P_x profiles

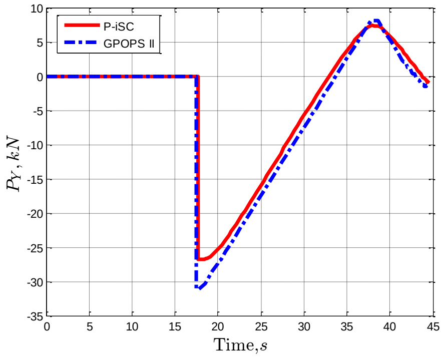

Figure 9 Comparison of P_y profiles

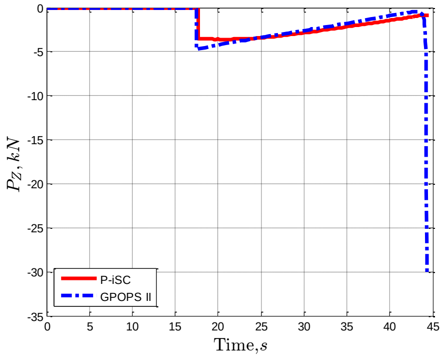

Figure 10 Comparison of P_z profiles

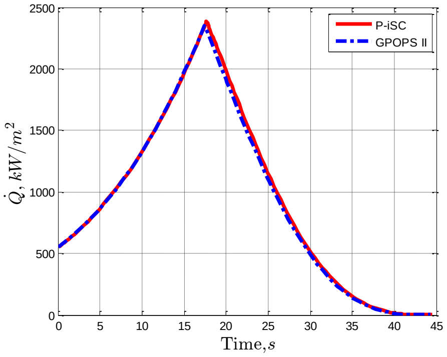

Figure 11 Comparison of heating rate profiles

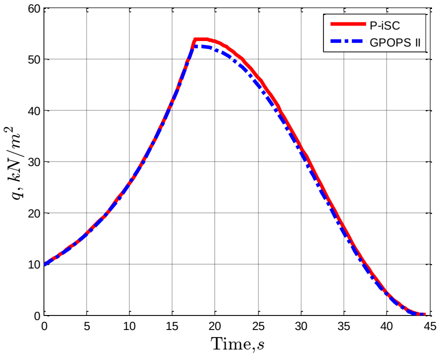

Figure 12 Comparison of dynamic pressure pressure profiles

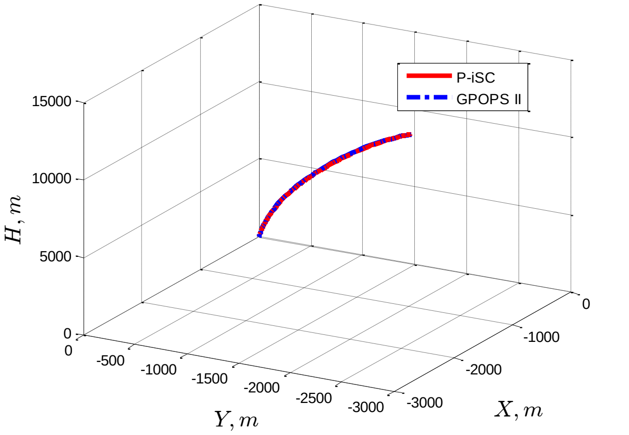

Figure 13 Comparison of three-dimensional trajectory

Table 3 Comparison of optimization results

	P-iSC	**GPOPS-II**
Ignition time (s)	17.70	17.43
Terminal time (s)	44.54	44.28
Final mass (kg)	27272.9	27342.8
CPU time consumption (s)	**4.7**	**107.5**

As shown in the figures and Table 3, the solutions using P-iSC and GPOPS-II are nearly identical. There are slight differences in the thrust profiles, ignition time, and terminal time. All the constraints are satisfied using both approaches. Similar results can be obtained using different rocket parameters, initial values, number of collocation points and termination criterion. That is to say, for the rocket landing guidance problem, the solution accuracy and general effectiveness of the P-iSC algorithm are comparable with those of GPOPS-II. Additionally, the statement in Theorem 1 that if the algorithm reaches convergence, the solution of the P-iSC algorithm is identical to the solution of the original nonlinear optimization problem (with tolerable error) is verified.

As stated previously, a feature of the P-iSC algorithm is the guarantee of the optimality of ignition time and terminal time, as we formulate the problem as a free-time optimization problem and take advantage of the unique time mapping of pseudospectral discretization. This statement can be verified by the numerical results, with multiple experiment settings, the ignition time and terminal time results using P-iSC are similar to the results of GPOPS-II with a deviation of subseconds.

It should be noted that P-iSC's CPU time consumption is only 4.37% of GPOPS-II's, so we can claim that under the same accuracy requirement, well-tuned P-iSC is much more efficient than general NLP based methods. This is due to the advantages of convex optimization theory.

B. Convergence performance of the P-iSC algorithm

In this subsection, the convergence performance of the P-iSC algorithm is analyzed, and a comparison is made between P-iSC and the classic successive convexification algorithm with fixed trust-region constraint (FT-SC). Same as in Section V.A, we set the termination criterion to be $\varepsilon_x = 10^{-5}, \varepsilon_h = 10^{-5}, \varepsilon_g = 10^{-5}$ (non-dimensional) in both algorithms, and the trust-region bounds in FT-SC are tuned for the best convergence performance. Table 4 shows the optimization results of different number of collocation points used in each phase. The average $|\Delta \boldsymbol{x}|$ values in each iteration of P-iSC and FT-SC with 50 collocation points in coast phase and 100 points in powered phase are shown in Figure 14-15.

Table 4 Comparison between P-iSC and FT-SC

Collo. Points phase I	Collo. Points phase II	Average CPU time for each iteration (s) P-iSC	Iteration times P-iSC	Average CPU time for each iteration (s) FT-SC	Iteration times FT-SC
20	20	0.05	12	0.04	19
20	50	0.09	11	0.9	16
50	50	0.15	11	0.14	24
80	80	0.31	10	0.28	35
50	100	0.47	10	0.44	44
80	100	0.51	10	0.52	46
100	100	0.62	13	0.76	45
50	150	0.66	9	0.71	41
50	200	1.01	11	0.95	47
100	200	1.34	10	1.47	55
150	200	2.02	13	2.07	62

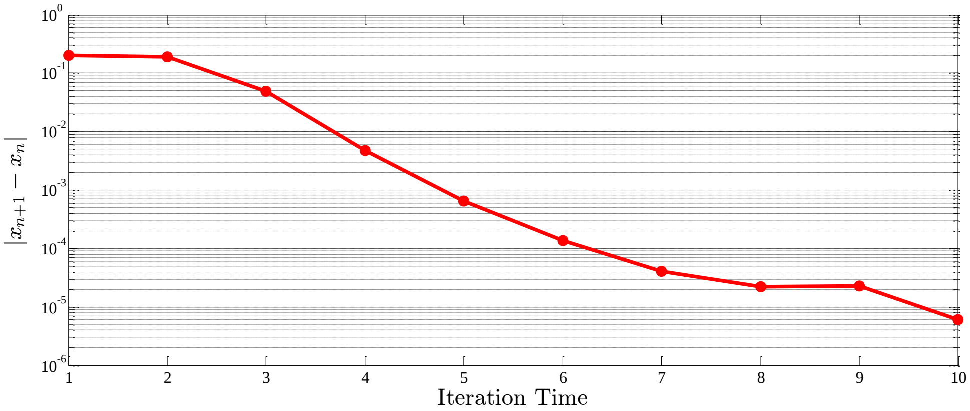

Figure 14 Average $|\Delta \boldsymbol{x}|$ in P-iSC with (50, 100) collocation points

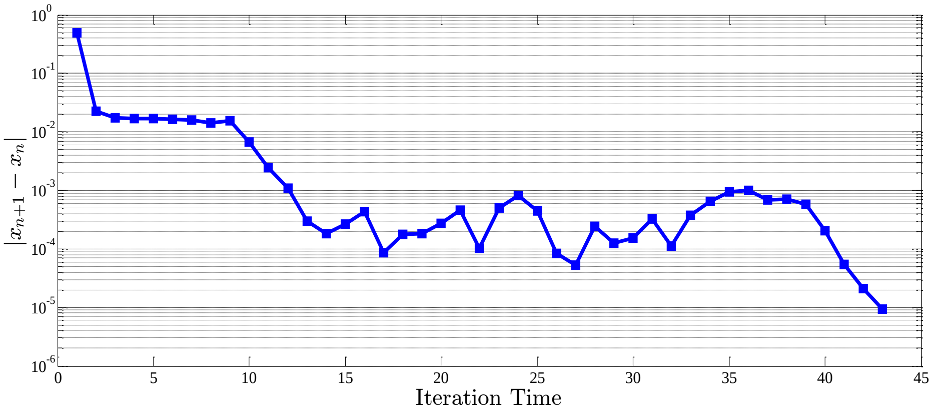

Figure 15 Average $|\Delta \boldsymbol{x}|$ in FT-SC with (50, 100) collocation points

From the figures and Table 4, the dynamic trust-region updating strategy brings apparent improvement to the convergence performance of successive convexification algorithm, and convergence is not sensitive to the number of collocation points while the number of collocation points significantly affects the iteration times of FT-SC algorithm.

For small collocation point number, the average CPU time for each iteration in P-iSC is slightly longer than in FT-SC. This is caused by the additional computation burden due to the evaluation of linearization error. As the collocation points number increases, the additional computation burden is overwhelmed by the optimization computing time.

VI. Conclusion

In this paper, we present an online trajectory optimization algorithm for rocket landing guidance problem. With the proper combination of pseudospectral discretization and convex optimization method, the precision and rapidness of the algorithm can be considered simultaneously. For the fuel optimal problem, the design of the engine ignition time and terminal flight time is a critical factor for optimizing the trajectory. Taking the advantage of unique time domain mapping of pseudospectral method, we formulate the problem as a free-time optimization problem. Ignition time and terminal time are considered as special control variables and are optimized together with other state and control variables. Along with the “spectral accuracy” of pseudospectral discretization, the optimality and precision of the solution are improved. To convexify the discrete nonlinear problem, an improved successive convexification algorithm is proposed, where a dynamic trust-region updating strategy is adopted to improve the convergence performance. According to the linearization error evaluated online during iterating, the trust-region is adjusted to reduce linearization error and speed up the convergence. Numerical experiments show that this strategy brings apparent improvements comparing with fixed trust-region based method, and the convergence is not sensitive to the number of discrete points. A convergence proof of iSC is given to provide a theoretical assurance for the algorithm. The high solution accuracy and the fast convergence rate are verified through numerical demonstrations.

In future work, sparse structure and code optimization of the algorithm will be studied to enhance the algorithm capability for onboard real-time applications, and hardware-in-the-loop simulation will be performed with actual onboard computer.

Appendix

The elements of matrix A, B and c_{ij} of the powered phase are given as follows. The elements of the coast phase can be gotten by setting $t_0 = 0$, $t_f = t_0$ and $\varGamma = P_x = P_y = P_z = 0$.

$$a_{11} = -(t_0 - t_f)\frac{\rho_0 e^{-\kappa\cdot H} V S_{ref} C_D}{m},\ a_{12} = -(t_0 - t_f)\frac{\mu\cdot\cos\gamma}{r^2},\ a_{16} = (t_0 - t_f)(\frac{\kappa\rho_0 e^{-\kappa\cdot H} V^2 S_{ref} C_D}{2m} + \frac{2\mu\cdot\sin\gamma}{r^3}),$$

$$a_{17} = (t_0 - t_f)\frac{P_x + D}{m^2},\ a_{21} = -(t_0 - t_f)\frac{P_y}{mV^2} + (t_0 - t_f)\frac{\cos\gamma}{r} + (t_0 - t_f)\frac{\mu\cdot\cos\gamma}{V^2 r^2},$$

$$a_{22} = -(t_0 - t_f)\frac{V\sin\gamma}{r} + (t_0 - t_f)\frac{\mu\cdot\sin\gamma}{Vr^2},\ a_{26} = -(t_0 - t_f)\frac{V\cos\gamma}{r^2} + 2(t_0 - t_f)\frac{\mu\cdot\cos\gamma}{Vr^3},\ a_{27} = -(t_0 - t_f)\frac{P_y}{m^2 V},$$

$$a_{31} = -(t_0 - t_f)\frac{P_z}{mV^2\cos\gamma} + (t_0 - t_f)\frac{\tan\phi\cos\gamma\sin\psi}{r},\ a_{32} = (t_0 - t_f)\frac{P_z}{mV\cos^2\gamma}\sin\gamma - (t_0 - t_f)\frac{V\tan\phi\sin\gamma\sin\psi}{r},$$

$$a_{33} = (t_0 - t_f)\frac{V\tan\phi\cos\gamma\cos\psi}{r},\ a_{34} = (t_0 - t_f)\frac{V\cos\gamma\sin\psi}{\cos^2\phi\cdot r},\ a_{36} = -(t_0 - t_f)\frac{V\tan\phi\cos\gamma\sin\psi}{r^2},$$

$$a_{37} = -(t_0 - t_f)\frac{P_z}{m^2 V\cos\gamma},\ a_{41} = (t_0 - t_f)\frac{\cos\gamma\cos\psi}{r},\ a_{42} = -(t_0 - t_f)\frac{V\sin\gamma\cos\psi}{r},\ a_{43} = -(t_0 - t_f)\frac{V\cos\gamma\sin\psi}{r},$$

$$a_{46} = -(t_0 - t_f)\frac{V\cos\gamma\cos\psi}{r^2},\ a_{51} = (t_0 - t_f)\frac{\cos\gamma\sin\psi}{r\cos\phi},\ a_{52} = -(t_0 - t_f)\frac{V\sin\gamma\sin\psi}{r\cos\phi},\ a_{53} = (t_0 - t_f)\frac{V\cos\gamma\cos\psi}{r\cos\phi},$$

$$a_{54} = (t_0 - t_f)\frac{V\cos\gamma\sin\psi}{r\cos^2\phi}\sin\phi,\ a_{56} = -(t_0 - t_f)\frac{V\cos\gamma\sin\psi}{r^2\cos\phi},\ a_{61} = (t_0 - t_f)\sin\gamma,\ a_{62} = (t_0 - t_f)V\cos\gamma.$$

$$b_{11} = -\frac{P_x}{m} - \frac{\rho_0 e^{-\kappa\cdot H} V^2 S_{ref} C_D}{2m} - \frac{\mu\cdot\sin\gamma}{r^2}, b_{12} == \frac{P_x}{m} + \frac{\rho_0 e^{-\kappa\cdot H} V^2 S_{ref} C_D}{2m} + \frac{\mu\cdot\sin\gamma}{r^2},\ b_{14} = -(t_0 - t_f)\frac{1}{m},$$

$$b_{21} = \frac{P_y}{mV} + \frac{V\cos\gamma}{r} - \frac{fM\cdot\cos\gamma}{Vr^2},\ b_{22} = -\frac{P_y}{mV} - \frac{V\cos\gamma}{r} + \frac{fM\cdot\cos\gamma}{Vr^2},\ b_{25} = (t_0 - t_f)\frac{1}{mV},$$

$$b_{31} = \frac{P_z}{mV\cos\gamma} + \frac{V\tan\phi\cos\gamma\sin\psi}{r},\ b_{32} = -\frac{P_z}{mV\cos\gamma} - \frac{V\tan\phi\cos\gamma\sin\psi}{r},\ b_{36} = (t_0 - t_f)\frac{1}{mV\cos\gamma},$$

$$b_{41} = \frac{V\cos\gamma\cos\psi}{r},\ b_{42} = -\frac{V\cos\gamma\cos\psi}{r},\ b_{51} = \frac{V\cos\gamma\sin\psi}{r\cos\phi},\ b_{52} = -\frac{V\cos\gamma\sin\psi}{r\cos\phi},\ b_{61} = V\sin\gamma,\ b_{62} = -V\sin\gamma,$$

$$b_{71} = -C_m\varGamma,\ b_{72} = C_m\varGamma,\ b_{73} = -(t_0 - t_f)C_m.$$

$$c_{11} = \rho_0 e^{-\kappa\cdot H} V,\ c_{16} = -\frac{1}{2}\kappa\rho_0 e^{-\kappa\cdot H} V^2,\ c_{21} = 3.15\cdot k_Q(\rho_0 e^{-\kappa\cdot H})^{0.5} V^{2.15},\ c_{26} = -\frac{1}{2}k_Q\kappa(\rho_0 e^{-\kappa\cdot H})^{0.5} V^{3.15}.$$

References

[1] Blackmore, L., “Autonomous Precision Landing of Space Rockets,” *Winter Bridge on Frontiers of Engineering*, Vol. 4, No. 46, 2016, pp. 15-20.

[2] Liu, X., “Fuel-Optimal Rocket Landing with Aerodynamic Controls,” *AIAA Guidance, Navigation, and Control Conference*, Grapevine, TX, 2017.

[3] Lu, P., “Introducing Computational Guidance and Control,” *Journal of Guidance, Control, and Dynamics*, Vol. 40, Special Issue on Computational Guidance and Control, 2017, pp. 193-193.

[4] Tsiotras, P., and Mesbahi, M., “Toward an Algorithmic Control Theory,” *Journal of Guidance, Control, and Dynamics*, Vol. 40, Special Issue on Computational Guidance and Control, 2017, pp. 194-196.

[5] Bollino, K. and Ross, I.M., “A pseudospectral feedback method for real-time optimal guidance of reentry vehicles,” *Proc. of IEEE American Control Conference*, New York, NY, July 2007.

[6] Bollino, K., “High-Fidelity Real-Time Trajectory Optimization for Reusable Launch Vehicles,” Ph.D. Dissertation, Naval Postgraduate School, Monterey, CA, Dec 2006.

[7] Ross, I.M., and Fariba, F. “Issues in the real-time computation of optimal control,” *Mathematical and computer modelling*, Vol. 43, No. 9, 2006, pp. 1172-1188.

[8] Morgan, D., Soon, J., and Fred, H. “Model predictive control of swarms of spacecraft using sequential convex programming,” *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 6, 2014, pp. 1725-1740.

[9] Wang, Z., and Michael, G. “Constrained Trajectory Optimization for Planetary Entry via Sequential Convex Programming,” *AIAA Atmospheric Flight Mechanics Conference*, 2016.

[10] Boyd, S. and Vandenberghe, L., *Convex Optimization*, Cambridge University Press, 2004.

[11] Açıkmeşe, B., and Ploen, S. R., “Convex Programming Approach to Powered Descent Guidance for Mars Landing”, *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 5, 2007, pp. 1353-1366.

[12] Blackmore, L., Açıkmeşe, B., and Scharf , D. P., “Minimum-Landing-Error Powered-Descent Guidance for Mars Landing Using Convex Optimization”, *Journal of Guidance, Control, and Dynamics*, Vol. 33, No. 4, 2010, pp. 1161-1171.

[13] Açıkmeşe, B., Scharf , D. P., Blackmore, L., and Wolf, A., “Enhancements on the Convex Programming Based Powered Descent Guidance Algorithm for Mars Landing,” *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, Honolulu, Hawaii, 2008.

[14] Açıkmeşe, B., and Blackmore, L., “Lossless convexification of a class of optimal control problems with non-convex control constraints,” *Automatica*, Vol. 47, No.2, 2011, pp. 341-347.

[15] Harris, M. W., “Lossless Convexification of Optimal Control Problems,” Ph.D. Dissertation, Aerospace Engineering Dept., Texas Univ., Austin, TX, 2014.

[16] Scharf, D. P., Regehr, M. W., Dueri, D., Açıkmeşe, B., Vaughan, G. M., Benito, J., Ansari, H., Aung, M., Johnson, A., Masten, D., Nietfeld, S., Casoliva, J., and Mohan, S., “ADAPT Demonstrations of Onboard Large-Divert Guidance with a VTVL Rocket,” *IEEE Aerospace Conference*, Inst. of Electrical and Electronics Engineers Paper AERO.2014.6836462, 2014.

[17] Scharf, D., Açikmeşe, B., Dueri, D., Casoliva, J., and Benito, J., “Implementation and Experimental Demonstration of Onboard Powered-Descent Guidance,” *Journal of Guidance, Control, and Dynamics*, Vol. 40, Special Issue on Computational Guidance and Control, 2017, pp. 213-229.

[18] Liu, X., Shen, Z., and Lu, P., “Entry Trajectory Optimization by Second-Order Cone Programming,” *Journal of Guidance, Control, and Dynamics*, Vol. 39, No.2, 2016, pp. 227-241.

[19] Lu, P., and Liu, X., “Autonomous Trajectory Planning for Rendezvous and Proximity Operations by Conic Optimization,” *Journal of Guidance, Control, and Dynamics*, Vol. 36, No. 2, 20173, pp. 375-398.

[20] Szmuk, M., Açıkmeşe, B., and Berning, A. W., “Successive Convexification for Fuel-Optimal Powered Landing with Aerodynamic Drag and Non-Convex Constraints,” *AIAA Guidance, Navigation, and Control Conference*, San Diego, CA, 2016.

[21] Mao, Y., Szmuk, M., and Açıkmeşe, B., “Successive convexification of non-convex optimal control problems and its convergence properties,” *2016 IEEE 55th Conference on Decision and Control (CDC)*, IEEE, 2016.

[22] Liu, X., Lu, P., and Pan, B., “Survey of Convex Optimization for Aerospace Applications,” *Astrodynamics*, Vol. 1, No. 1, 2017.

[23] Garg, D., “Advances in global pseudospectral methods for optimal control,” Ph.D. Dissertation, Diss. Florida Univ., Gainesville, FL, 2011.

[24] Fariba, F., and Ross, I.M. “Advances in pseudospectral methods for optimal control,” *AIAA guidance, navigation and control conference and exhibit*. 2008.

[25] Gong, Q., , Ross, I.M., et al. “Connections between the covector mapping theorem and convergence of pseudospectral methods for optimal control,” *Computational Optimization and Applications* Vol.41, No.3, 2008, pp. 307-335.

[26] Gong, Q., Ross, I.M., and Fariba, F. "Spectral and Pseudospectral Optimal Control Over Arbitrary Grids." Journal of Optimization Theory and Applications 169.3 (2016): 759-783.

[27] Sagliano, M., Theil, S., et al. “On the Radau pseudospectral method: theoretical and implementation advances,” *CEAS Space Journal*, Vol.9, No.3, 2017, pp. 313-331.

[28] Musk, E., et al., “X marks the spot: Falcon 9 attempts ocean platform landing,” SpaceX [website], URL: http://www.spacex.com/news/2014/12/16/x-marks-spot-falcon-9-attempts-ocean-platform-landing [cited 15 May 2017]

[29] Lu, P., “Entry Guidance: A Unified Method,” *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 3, 2014, pp. 713-728.

[30] Nesterov, Y., and Nemirovsky, A., *Interior-Point Polynomial Methods in Convex Programming*, SIAM, Philadelphia, PA, 1994.

[31] Fletcher, R., *Practical Methods of Optimization: Vol. 2: Constrained Optimization.* John Wiley & Sons, 1981.

[32] Kanzow, C., Nagel, C., Kato, H., and Fukushima, M., “Successive linearization methods for nonlinear semidefinite programs,” *Computational Optimization and Applications*, Vol. 31, No. 3, 2005, pp. 251-273.

[33] Patterson, M. A., and Rao, A. V., “GPOPS-II: A MATLAB software for solving multiple-phase optimal control problems using hp-adaptive Gaussian quadrature collocation methods and sparse nonlinear programming,” *ACM Transactions on Mathematical Software (TOMS)*, Vol.41, No. 1, 2014, pp. 1:1-1:37.

[34] Andersen, E. D., Roos, C., and Terlaky, T., “On Implementing a Primal-Dual Interior-Point Method for Conic Quadratic Optimization,” Mathematical Programming, Vol. 95, No. 2, 2003, pp. 249–277.