The Journal of the Astronautical Sciences (2020) 67:1553–1574
https://doi.org/10.1007/s40295-020-00235-y

ORIGINAL ARTICLE

An Iterative Convex Programming Method for Rocket Landing Trajectory Optimization

Jinbo Wang[1] · Huixu Li[1] · Hongbo Chen[1]

Accepted: 11 September 2020 Published online: 30 September 2020

Abstract

A rapid trajectory optimization method is proposed to solve the fuel-optimal Earth-landing problem of reusable rockets, in which the nonlinear aerodynamic drag force is non-negligible. To enable the online and autonomous operation ability, the method is designed based on convex optimization, which features rapid and deterministic convergence properties, and a homotopic-iterative strategy is proposed to convexify the nonlinear system dynamics of the rocket. In the proposed iterative algorithm, the problem is first solved based on the lossless convexification method while the drag force is considered to be zero. Then, during subsequent iterations, the drag profile is approximated by the last solution and homotopically added to the problem. Thus, the nonlinear drag is gradually included while the problem remains convex. Because the convexification of the nonlinear terms is not based on linearization, no reference trajectory or initial guess is needed, which greatly enhances the autonomy of the algorithm. Numerical experiments are provided to demonstrate the effectiveness, rapidness, and accuracy of the proposed algorithm.

Keywords Reusable rocket landing · Online trajectory optimization · Convex optimization · Lossless convexification · Homotopy method

✉ Jinbo Wang
wangjinbo@mail.sysu.edu.cn

Huixu Li
lihx43@mail2.sysu.edu.cn

Hongbo Chen
chenhongbo@mail.sysu.edu.cn

[1] School of Systems Science and Engineering, Sun Yat-Sen University, Guangzhou 510275, People's Republic of China

Introduction

Reusable launch vehicles (RLVs) are effective tools for fulfilling the demands of low cost and fast access to space. Since the era of the space shuttle, many different types of RLVs have been developed and studied. In recent years, with the rapid progress of commercial aerospace companies, such as SpaceX and Blue Origin, vertical takeoff vertical landing (VTVL) RLVs that have evolved from conventional launch vehicles have received increased attention. Vertical landing is a significant feature that characterizes a VTVL rocket. This paper focuses on the online landing trajectory optimization of VTVL rockets, which would enable advanced and autonomous landing guidance.

With the development of computing technology and advanced theories and algorithms, computational methods have become popular for solving aerospace guidance and control problems [21]. In particular, real-time trajectory optimization is a promising and essential approach for achieving optimal and autonomous guidance under complex environments and constraints [7]. In general, numerical methods for trajectory optimization can be categorized into indirect and direct methods [5], where direct methods can be more easily applied to problems with complex vehicle dynamics and constraints. Compared with other direct methods, methods based on pseudospectral discretization can achieve the same accuracy with fewer discretization points and converge faster for smooth problems [15]. Thus, such methods can potentially be used online [25, 26]. However, similar to other direct methods, the problem solving of classic pseudospectral methods mostly relies on nonlinear programming (NLP) solvers; thus, the issues inherent to NLP, such as initial guess sensitivity and indeterminable solution times, which are undesirable for online and real-time applications, are inherited by classic pseudospectral methods.

In recent years, a special type of direct method, namely, the convex optimization method, has been successfully applied to aerospace trajectory optimization and guidance [17, 27, 38]. Unlike traditional NLP algorithms, convex programming algorithms have deterministic convergence properties in addition to their rapidness, making them computationally tractable and propitious for online applications. Because the dynamics and constraints of aerospace problems are mostly nonconvex, the premise of utilizing convex programming algorithms is to transform the optimization problem into an equivalent or approximate convex formulation. Lossless convexification and successive convexification are two powerful approaches for convexifying a problem. The lossless convexification method was proposed by Açıkmeşe et al. [1, 2, 6] to solve the Mars powered landing problem. In addition, with a customized real-time interior-point method (IPM) [10], a guidance algorithm based on lossless convexification was verified through successful flight experiments [32]. The greatest advantage of this method is that little approximation error is introduced by the convexification procedure. For nonlinear dynamics and constraints that cannot be convexified losslessly, successive convexification methods have been developed [16, 22, 35]. The basic methodology of successive convexification is to linearize or approximate the nonlinear constraints to obtain a corresponding convex subproblem, which is then solved iteratively until the solution converges.

In the spirit of jointly utilizing the various powerful mathematical tools above in different stages of solving trajectory optimization problems, pseudospectral convex methods have been developed. In the work by Sagliano [27–29], Cheng et al. [9], and

Wang et al. [34–36], the idea of combining the advantages of the pseudospectral method and convex optimization to rapidly obtain accurate optimal solutions was proposed; the rapidness and high accuracy of the proposed algorithms are due to the properties of the convex optimization method and pseudospectral discretization, respectively.

For rocket landing trajectory optimization and guidance problems studied in this paper, researchers exhibit great tendencies of using convex optimization methods to achieve online operation reliability. Liu analyzed the cooperative effects of thrust and aerodynamic forces on the fuel cost of the two-dimensional planar landing flight by using a successive convexification algorithm [18]. Szmuk et al. studied the six-degree-of-freedom powered descent guidance problem using successive convexification method, and a novel state-triggered constraint formulation is proposed, which facilitates the inclusion of complex "if statements conditioned" constraints [33]. Simplício et al. carried out tradeoff analyses to provide insights to the coupled rocket landing flight mechanics and guidance problems, and a two-stage trajectory optimization algorithm based on convex optimization was designed [30]. Wang et al. proposed a multi-phase pseudospectral-improved successive convexification method and embedded it into a parallel model predictive control framework to solve the rocket landing guidance problem [36]. In spite of the existence of great differences, the above literature (and many other publications that are not cited) is mostly constructed based on well established lossless convexification and successive linearization/convexification techniques. Although successful applications have been reported, the existing methods still present some issues in terms of solution accuracy and autonomy.

The lossless convexification method is known for its theoretical elegance and ease of implementation for convexifying the minimum thrust constraint. The non-drag and constant-gravity model is sufficiently accurate for Mars landing missions. However, the nonlinear drag force, which is beyond the scope of lossless convexification, is one of the most critical factors affecting the Earth-landing scenarios studied in this paper. In addition, in Ref. [1], variable substitutions are introduced to address the nonconvexities caused by the time-varying mass, and the transformational thrust constraint is then approximated by Taylor series expansion. The approximation can be very accurate for Mars missions. However, for the reusable rocket discussed in this paper, the vehicle mass and the thrust magnitude are much greater than those of a Mars lander; thus, the approximation errors are no longer acceptable for precision landing missions.

For the successive convexification methods, despite their practicability and satisfactory performance, the algorithms rely on reference trajectories for linearization based on Taylor series expansion. That is, a reference trajectory must be provided by the user or an initial-trajectory generator to enable convexification and to activate the iteration, and additional trust-region constraints must be enforced to ensure the validity of linearization. More importantly, the convergence of the algorithm is related to the quality of the reference trajectory, which is an issue similar to the initial guess sensitivity of NLP, but rarely discussed in the literature. However, this issue is important because one of the major benefits coming from convex optimization, that is, not relying on an initial guess, would be lost.

Besides convex optimization method, another powerful tool for solving complex trajectory optimization problems is the homotopy method (also called numerical continuation method). The idea of the homotopy method is to solve a difficult-handling problem by starting from a related, easy-solving one, and a family of problems parameterized by homotopic parameters are solved successively until the solution to the original problem is obtained [13, 23]. Lu et al. adopted an atmospheric density homotopy method to solve the endoatmospheric ascent guidance problem of X-33 like RLVs [19]. Pan et al. scientifically surveyed the application of homotopy methods in aerospace trajectory optimization and proposed a double-homotopy method to address the issues of critical point and homotopy path wandering off to infinity [23]. Bai et al. utilized the pseudospectral method to generate initial information of the bang-bang type control problem, and a homotopy method is designed to enforce the constraints satisfaction along the homotopy path [4]. Solving the non-smooth bang-bang type fuel-optimal problem homotopically using an auxiliary smooth energy-optimal problem is also a common practice in the research on low thrust [13] and asteroid landing [37] trajectory optimization. Brendl et al. studied the aerodynamic forces homotopy method to solve the optimal toss back guidance problem of the VTVL RLV [8].

It should be noted that, while the convexification and homotopy methods both transform a complicated problem into a series of "easy" ones, most of the existing research endeavors (including those cited above) on the homotopy method mainly focus on resolving the initial co-state sensitivity issue of two point boundary value problems (TPBVP) derived from Pontryagin's maximum principle (i.e., indirect method). In the current research, the homotopy strategy is embedded in a direct convex optimization method.

This paper is motivated by the imperative online trajectory optimization applications of VTVL rocket landing, which requires an efficient algorithm that can rapidly, automatically and accurately produce optimal trajectories. To circumvent the aforementioned issues of existing convexification methods, a homotopic-iterative strategy is designed, and a novel iterative convex programming (ICP) algorithm is proposed. In ICP, a non-drag approximate problem is first solved based on the lossless convexification method. Then, using the obtained solution, the drag force is estimated and homotopically (under the "direct method framework") added to the nominal problem in the form of a "prescribed profile"; that is, during an iteration, the drag force term is approximated by a specific constant value at each discretization point, and thus, the problem remains convex. Furthermore, in the nominal problem, the optimization model is restored to its accurate form from the lossless convexification formulation used in the approximate problem. Since nonlinear terms are convexified using the homotopic-iterative strategy rather than linearization, no reference trajectory is needed, and the initial-guess independence of convex optimization is regained in ICP. Thus, the algorithm is efficient and fully automated, which is favorable for online applications. Numerical experiments are performed to evaluate the proposed algorithm. The effectiveness, rapidness, and accuracy of the algorithm are demonstrated.

The rest of the present paper is organized as follows. First, the rocket landing trajectory optimization problem is formulated in Sec. 2. Then, the proposed ICP method is detailed in Sec. 3. Numerical experiments are performed and analyzed in Sec. 4. Finally, Sec. 5 concludes the paper.

Problem Formulation and Analyses

Problem Formulation

This paper focuses on the final landing flight of a reusable rocket that lands on the Earth or another planet with an atmosphere. Using the landing process of the first stage booster of the SpaceX Falcon-9 rocket as an analogy, the flight phase studied here begins from the "landing burn" ignition and ends at final touchdown. Because the angle of attack (AoA) is usually relatively small during landing, the aerodynamic lift force is insignificant compared with the drag force. Here, the drag force is considered in the dynamics, while the lift force is neglected and can be treated as a disturbance. In a Cartesian coordinate frame with the origin at the landing target, as shown in Fig. 1, the translation dynamics of the rocket are expressed as follows:

$$\dot{\boldsymbol{r}} = \boldsymbol{V} \tag{1}$$

$$\dot{\boldsymbol{V}} = \frac{\boldsymbol{T}}{m} + \frac{\boldsymbol{D}}{m} + \boldsymbol{g}(\boldsymbol{r}) \tag{2}$$

$$\dot{m} = -\alpha \|\boldsymbol{T}\|_2 \tag{3}$$

where $\mathrm{r} = [r_x, r_y, r_z]^T$ and $V = [V_x, V_y, V_z]^T$ are the position and velocity vectors, m is the vehicle mass, and $\mathrm{T} = [T_x, T_y, T_z]^T$ is the thrust vector, which is the control variable of the system. Vector $\boldsymbol{g}(\boldsymbol{r})$ is the gravitational acceleration, which is a nonconstant function of $\boldsymbol{r}$. The constant $\alpha = 1/(I_{sp}g_0)$, where I_{sp} is the specific impulse of the rocket engine, and g_0 is gravitational acceleration at sea level. The drag force is $\boldsymbol{D} = -\rho S_{\mathrm{ref}} C_D \|\boldsymbol{V}\|_2 \boldsymbol{V}/2$, where ρ is the density of air, S_{ref} is the reference area of the rocket, and C_D is the drag coefficient.

Although the constant-gravity model is sufficiently accurate for the final landing flight, in which the scope of flight is relatively small, using a high-fidelity gravity model does not introduce any additional difficulty into the present method. Therefore,

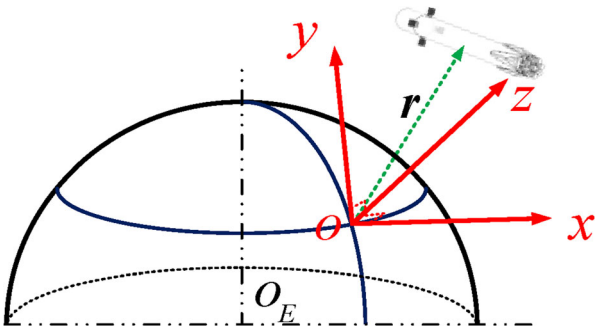

Fig. 1 Coordinate frame for the equation of motion

the algorithm proposed in this paper can be easily extended to accommodate applications in which the fidelity of the gravity model is critical, such as the high-altitude retropropulsion flight of a reusable rocket, e.g., the "entry burn" flight of the SpaceX Falcon 9 rocket.

During flight, to ensure reliability, the rocket engine has a nonzero minimum throttle level, and the thrust is bounded as follows:

$$T_{\min} \le \|\mathrm{T}\|_2 \le T_{\max} \tag{4}$$

where $T_{\min}$ and $T_{\max}$ are the minimum and maximum available values of the thrust magnitude. Because $T_{\min} > 0$, the left side of Eq. (4) is nonconvex.

For pinpoint landing, the initial and terminal state constraints are as follows:

$$\begin{cases} \boldsymbol{r}(t_0) = \boldsymbol{r}_0, \boldsymbol{V}(t_0) = \boldsymbol{V}_0, m(t_0) = m_0, \\ \boldsymbol{r}(t_f) = \boldsymbol{r}_f, \boldsymbol{V}(t_f) = \boldsymbol{V}_f, m(t_f) \ge m_{\mathrm{dry}} \end{cases} \tag{5}$$

where t_0 and t_f are the fixed initial and terminal times of flight, respectively, and m_{dry} is the dry mass of the rocket.

For the fuel-optimal problem, the performance index can be simply set as the terminal vehicle mass. Then, the trajectory optimization problem can be formulated as follows:

$$\textit{Problem}\ \ 0\ \ (P0) \qquad \begin{aligned} &\min\ J = -m_f \\ &\text{s.t.}\quad \text{Eqs.}(1)\sim(5) \end{aligned}$$

In addition to the constraints stated above, other common constraints for rocket landing missions, such as the glideslope constraint and the thrust pointing constraint [2], can be easily incorporated into P0. Such constraints do not affect the present method, as they are already convex. Here, we simply omit the constraints for simplicity and clarity. Moreover, the minimum-landing-error [6] and maximum-divert [14] problems can also be solved by the present method. The fuel-optimal problem is chosen here because it may be the most common case for Earth-landing missions.

Problem Analyses

In the present research, the aerodynamic drag force is considered while designing a new convexification strategy for the rocket landing problem. Thus, in this subsection, we first investigate the effect of the drag force on the landing trajectory.

Unlike entry vehicles and other aircraft, aerodynamic forces do not act as the main controlling forces for the reusable rocket during the final landing flight. That is because the velocity of the rocket is relatively small and the magnitude of the aerodynamic drag is much less than that of the thrust. Using the parameters shown in Sec. 4, Fig. 2 illustrates the magnitudes of the forces acting on the rocket during the landing flight, which shows that, for a typical landing mission, the velocity profile is dominated by the thrust magnitude, so does the drag profile. Fig. 3 illustrates the time integrations of the corresponding acceleration norms (i.e., the ΔV contributions), and we find that,

although it is non-negligible, the integral effect of the drag force on the trajectory is much weaker than that of the thrust. Thus, based on the assumption that the value of the drag coefficient is not very large ($C_D \leq 1.22$, as indicated in Sec. 4), and within the scope of the final landing phase, we describe the aerodynamic drag as a "non-negligible but non-dominating force" in the current research. For powered landing flight, estimating the aerodynamic coefficients is challenging since the engine jet plume would exert complex effects on the flow field around the rocket, and available reference database is not substantial. In Ref. [31], it is reported that for a similar landing rocket, the drag coefficient is smaller than 1 during low Mach number and small AoA flight.

Furthermore, based on the formulation of P0, the problem can be efficiently solved by the lossless convexification method when drag is omitted and gravity is considered constant, even though the accuracy of the transformational thrust constraint is not sufficiently high, as discussed in Sec. 1.

Combining the above findings, and inspired by the homotopic strategy extensively studied in the indirect trajectory optimization methods [13, 19, 23, 37], we propose an iterative method. Based on the homotopy method, starting from the solution of a related but simpler auxiliary problem, the algorithm is initialized by a lossless convexification subroutine, in which the drag force is omitted and several approximations are made. Thus, the problem can be easily solved without a reference trajectory. Then, in subsequent iterations, the drag term is estimated at each discretization point along the trajectory using the last solution, and it is homotopically introduced into the discretized system dynamics. The development of this method is discussed in detail in the next section.

Iterative Convex Programming Method

Based on the goal of obtaining improved autonomy and the characteristics of the optimization model, in this section, we propose an ICP method for solving the powered Earth-landing problem. The solution process of ICP proceeds as follows.

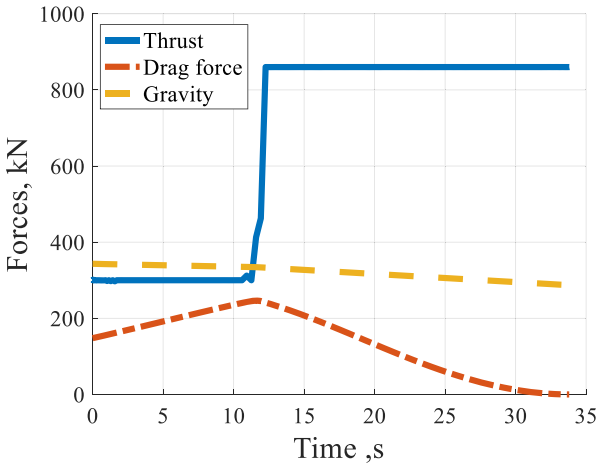

Fig. 2 Magnitudes of the acting forces during flight

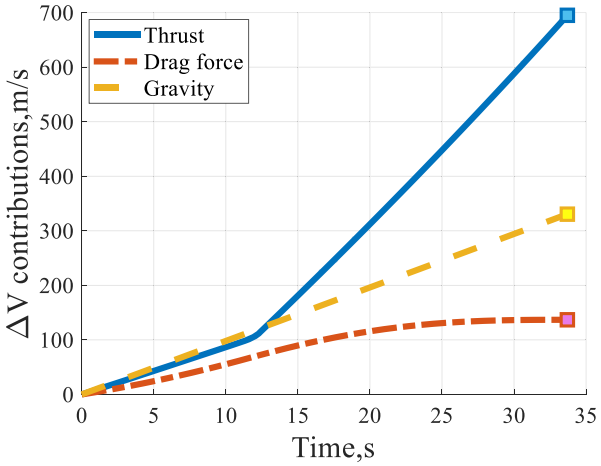

Fig. 3. ΔV contributions during flight.

First, the drag force is omitted, and an average constant gravity is adopted; thus, by using the lossless convexification method proposed in Ref. [1], the problem can be transformed into a second-order cone programming (SOCP) problem and solved efficiently using a matured IPM solver. We call this solution the *starter* of the algorithm.

Then, the drag profile can be approximated by using the velocity and height profiles of the starter solution. In the next iteration, the approximate drag is introduced into the system dynamics at each discretization point as a *specific constant*, such that the problem remains convex. We call the above approximated drag profile a "prescribed profile". However, the incipient approximate drag profile could be rather coarse, and the approximation error may render the problem infeasible. To prevent this situation, the drag term is not fully added at once, but rather is homotopically added into the problem in iterations until the obtained velocity and height trajectories are proper approximations of the solution of P0. For other approximations made in the lossless convexification formulation of the starter problem, such as the gravity model and the transformational thrust constraint, the model errors will be eliminated iteratively by similar procedures.

In each iteration, the solution of the last iteration is used to update the approximate drag, mass, gravity, and constraints. The iterative process stops when the solution converges.

The ICP method can be considered as a proper combination of lossless convexification, successive approximation, and the homotopy method. With such a strategy, the nonlinear drag force can be incorporated into the convex optimization algorithm efficiently, and the algorithm does not rely on any reference trajectory or initial guess.

The rationality of the ICP method is rooted in the fact that the aerodynamic drag force is non-negligible but non-dominating during the final landing flight. The rocket engine should be powerful enough to decelerate the rocket to land softly without the "help" of the drag force. Thus, if the starter solution is feasible, then the problem P0 must also be physically feasible. In addition, with the effective and efficient compensation of the drag force, the converged solution would be highly accurate relative to the nominal optimal solution. The rest of this section provides the details of the ICP method.

Problem Convexification

There are two convexified problems to solve using the ICP algorithm: the starter problem, i.e., the losslessly convexified non-drag problem (approximate problem), and the iteratively solved prescribed-drag-profile problem (nominal problem).

The approximate problem is formulated by omitting the drag force, using a constant-gravity model, and introducing a slack variable and several transformational variables to convexify the problem. Here, we directly express the problem, and the detailed convexification procedure can be found in Ref. [1].

Losslessly Convexified Non-Drag Problem (Approximate Problem P1):

$$\min J = -z_f \tag{6}$$

s.t.

$$\dot{\boldsymbol{r}} = \boldsymbol{V} \tag{7}$$

$$\dot{\boldsymbol{V}} = \boldsymbol{u} + \overline{\mathbf{g}} \tag{8}$$

$$\dot{z} = -\alpha\sigma \tag{9}$$

$$\|\boldsymbol{u}\|_2 \leq \sigma \tag{10}$$

$$T_{\min}e^{-z_0}\left\{1-[z(t)-z_0(t)] + [z(t)-z_0(t)]^2/2\right\} \leq \sigma(t) \leq T_{\max}e^{-z_0}\{1-[z(t)-z_0(t)]\} \tag{11}$$

$$\ln(m_0-\alpha T_{\max}t) \leq z(t) \leq \ln(m_0-\alpha T_{\min}t) \tag{12}$$

$$\begin{cases} \mathrm{r}(t_0) = \boldsymbol{r}_0,\ \mathrm{V}(t_0) = \boldsymbol{V}_0,\ z(t_0) = \ln(m_0), \\ \mathrm{r}(t_f) = \boldsymbol{r}_f, \mathrm{V}(t_f) = \boldsymbol{V}_f, z(t_f) \geq \ln(m_{\mathrm{dry}}) \end{cases} \tag{13}$$

where $\overline{\mathbf{g}} = \left[\mathbf{g}(\boldsymbol{r}_0) + \boldsymbol{g}(\boldsymbol{r}_f)\right]/2$ is the average constant gravity, $\boldsymbol{u} = \boldsymbol{T}/m$ is the transformational thrust acceleration, $\sigma = \Gamma/m$ is the transformational slack variable,

Γ is the slack variable introduced to convexify the thrust constraint, $z = \ln(m)$ is the transformational mass, and $\|\boldsymbol{u}\|_2 \leq \sigma$ is the transformational thrust constraint transformed and convexified from Eq.(4). The constraints in Eq. (12) are set to ensure that the bounds on the transformational mass will not be violated, and $z_0(t) = \ln(m_0 - \alpha T_{\max} t)$. The constraints in Eq. (11) are the second-order cone (left side) and linear (right side) approximations of the following nonconvex constraints:

$$T_{\min} e^{-z(t)} \leq \sigma(t) \leq T_{\max} e^{-z(t)} \tag{14}$$

which is the transformational form of the constraints on the original slack variable Γ.

Eq. (14) contains variables that represent the vehicle mass and the magnitude of the engine thrust, which are quite large for a VTVL rocket booster compared with those of the Mars lander studied in Ref. [1]. Thus, the first and second order approximations used in Eq.(11) are not sufficient for the current problem, as discussed in Sec. 1. This issue will be ameliorated in the nominal problem.

The cost function of P1 is linear, and all constraints are linear or second-order cones; thus, after discretization, P1 is a SOCP problem.

Next, we formulate the problem to be iteratively solved via ICP as follows:

Iteratively Solved Prescribed-Drag-Profile Problem (Nominal Problem P2):

$$\min J = -z_f \tag{15}$$

s.t.

$$\dot{\boldsymbol{r}} = \boldsymbol{V} \tag{16}$$

$$\dot{\boldsymbol{V}} = \boldsymbol{u} + \delta \cdot \widehat{\boldsymbol{D}}\left(\underline{\boldsymbol{r}}, \underline{\boldsymbol{V}}\right) / \underline{\boldsymbol{m}} + \widehat{\boldsymbol{g}}\left(\underline{\boldsymbol{r}}\right) \tag{17}$$

$$\dot{z} = -\alpha\sigma \tag{18}$$

$$\|\boldsymbol{u}\|_2 \leq \sigma \tag{19}$$

$$T_{\min} \exp(-\underline{z}) \leq \sigma \leq T_{\max} \exp(-\underline{z}) \tag{20}$$

$$\ln(m_0-\alpha T_{\max}t)\leq z(t)\leq \ln(m_0-\alpha T_{\min}t) \tag{21}$$

$$\begin{cases} \mathrm{r}(t_0)=\boldsymbol{r}_0,\ \mathrm{V}(t_0)=\boldsymbol{V}_0,\ \boldsymbol{z}(t_0)=\ln(m_0), \\ \mathrm{r}(t_f)=\boldsymbol{r}_f, \mathrm{V}(t_f)=\boldsymbol{V}_f, \boldsymbol{z}(t_f)\geq \ln(m_{\mathrm{dry}}) \end{cases} \tag{22}$$

where the underlined variables (e.g., $\underline{\boldsymbol{V}}$) are taken from the starter solution or the last iteration, and the variables with a hat (e.g., $\widehat{\boldsymbol{D}}$) are approximated by the underlined variables. It should be noted that $\widehat{\boldsymbol{D}}/\underline{\boldsymbol{m}}$ and $\widehat{\boldsymbol{g}}$ are not constant values, but rather prescribed profiles that are *time-varying* along the trajectory. The variable δ is the homotopy parameter, which varies between 0 and 1. The design of δ is discussed in Sec. 3.3. When δ equals 0, P2 is degraded to the non-drag problem, and when δ equals 1, P2 is an approximation of P0.

Because $\widehat{\boldsymbol{D}}/\underline{\boldsymbol{m}}$ and $\widehat{\boldsymbol{g}}$ are constant terms *at any specific time point t* along the trajectory, the velocity dynamics Eq. (17) is linear and convex. In addition, the constraints on the transformational slack variable σ in Eq. (20) take their original forms as in Eq. (14), where the parameters of the exponential functions are also taken from the starter solution or the last iteration. Thus, similar to $\widehat{\boldsymbol{D}}/\underline{\boldsymbol{m}}$ and $\widehat{\boldsymbol{g}}$, the constraint bounds are prescribed profiles, and the constraints are linear.

The above convexification method is a simple and effective approach for avoiding the accuracy issue discussed in Sec. 1 while keeping the constraints convex. Moreover, in P2, the convexification of nonlinear terms is not based on the traditional linearization method, and when the iterative solution converges, the dynamics and constraints in P2 are equivalent to those in P0.

Problem Discretization

Using the same number of discretization points, the discretization accuracy of pseudospectral methods would significantly exceed that of other transcription methods [15], which is valuable for online algorithms. Thus, in this paper, the flipped Radau pseudospectral (FRP) method is adopted to discretize P1 and P2. The FRP method is chosen from the pseudospectral families because it is straightforward to formulate the initial conditions, and the method has favorable theoretical and numerical characteristics [27]. Comprehensive discussions of the Radau pseudospectral method can be found in Ref. [11].

For the rocket landing problem, Sagliano [27] and Wang et al. [34] have provided detailed discretization processes of the non-drag problem P1 and the original nonlinear problem P0 (with a different reference frame) using FRP, respectively. In addition, because the FRP method is already relatively mature, the discretization processes are quite straightforward. Thus, the formulations of the discretized problems are not listed to save space; interested readers can find these formulations in Ref. [27] and Ref. [34]. Here, we refer to the discretized version of P1 as *DP1* and the discretized version of P2 as *DP2*.

The ICP Algorithm

The basic idea of the ICP method is as stated above. In this subsection, the algorithm is described, and the technical details are discussed. In Algorithm 1, for simplicity, the state and control variables are denoted together as x.

Algorithm 1: The Iterative Convex Programming Algorithm

Initialization. Set *homotopy steps* $0<\omega_1<\omega_2<1$; set the termination criterion parameter $\varepsilon>0$; set the maximum number of iterations $k_{\max}$; set index $k=0$.

1 ***Solve*** DP1, obtain solution $\underline{x}^0$;
2 ***Set*** $\delta=0+\omega_2$, $\Delta x^0=\underline{x}^0$;
3 **while** $|\Delta x^k|>\varepsilon$
4 ***Update*** $\hat{D}$, $\hat{g}$, and the constraint bounds on σ with $\underline{x}^k$;
5 ***Solve*** DP2, obtain solution $\underline{x}^{k+1}$ when feasible;
6 **if** DP2 is *feasible*
7 ***Set*** $\delta=\min\{\delta+\omega_2,1\}$, and $\Delta x^{k+1}=\left|\underline{x}^{k+1}-\underline{x}^k\right|$;
8 ***Set*** $k=k+1$;
9 **if** $k>k_{\max}$ **STOP**;
10 **end if**
11 **else if** DP2 is *infeasible*
12 ***Set*** $\delta=\max\{\delta-\omega_1,0\}$;
13 **end if**
14 **end while**

During initialization, some necessary parameters, including the homotopy steps ω_1 and ω_2, which guarantee the feasibility of the iteration, need to be provided by the user.

In line 1, the losslessly convexified and FRP discretized problem DP1 is solved, and $\underline{x}^0$ is the abovementioned starter solution. Then, in line 2, the homotopy parameter δ is updated with step ω_2; thus, in the second iteration, $100\cdot\omega_2$ percent of the approximate drag force is added to the problem.

Lines 3–14 are the main iteration steps. The while loop stops if the solutions of two successive iterations are sufficiently close, as determined by the termination criterion parameter ε, as indicated in line 3. Alternatively, if the maximum number of iterations is reached, then the loop is exited, as indicated in line 9.

In each iteration, the approximate prescribed profiles and the constraint bounds are first updated using the data of the starter solution or the last iteration. Then, with updated parameters, the iteratively convexified and FRP discretized problem DP2 is solved. In the early iterations, because the drag approximation is not yet sufficiently

accurate, DP2 may be infeasible. Thus, the if-else branch in lines 7–10 pertains to the feasible case, and line 12 pertains to the infeasible case.

If DP2 is feasible, then the current solution is used to update the parameters of the next iteration, and the homotopy parameter δ is increased by step ω_2 until it reaches one. If DP2 is infeasible, then the homotopy parameter δ is decreased by step ω_1 until it reaches zero, and the current iteration is repeated until DP2 becomes feasible. Although numerical experiments show that DP2 would hardly be infeasible, it is still necessary to perform this security step to avoid unexpected errors, and it is a common practice in the homotopy method [37]. In addition, by setting $\omega_1 < \omega_2$, when the drag force is added "too fast" and infeasibility occurs, the maximum allowable δ value is sought to continue the iteration.

It can be observed that the implementation of the ICP algorithm is quite simple and straightforward, and only two parameters (ω_1 and ω_2) must be tuned by the user. Moreover, no prior or empirical information about the landing trajectory is required.

Discussion

In this subsection, some features of the proposed algorithm are further discussed.

Regarding the algorithm's ability and implementation, as desired based on the motivation for this research, the ICP algorithm is fully automated. That is, unlike in successive convexification algorithms based on linearization, no initial-trajectory information is needed to start the algorithm. Furthermore, no trust-region constraints [34] must be designed, and the design and tuning of the two homotopy steps are very simple.

Regarding the performance of the algorithm, first, because the nonlinear drag and gravity are approximated as prescribed profiles using the solution of the last iteration and do not depend on the current optimal variables, the discretization matrix of ICP is sparser than that of successive convexification algorithms. In every iteration, the CPU time required to solve DP2 is comparable to that required to solve the losslessly convexified non-drag problem. Second, with the design of the homotopic-iterative strategy, ICP converges rapidly.

Regarding the flexibility of the algorithm, various nonlinear terms in the system dynamics of rockets, such as the aerodynamic forces, gravity, and the terms related to the Earth's rotation, can be addressed by the proposed homotopic-iterative strategy. Thus, the method can be easily extended (with proper parameter adjustment) to accommodate different rocket planetary landing and ascent problems.

There are two disadvantages of the ICP algorithm. First, the algorithm requires a specified time of flight, which is inherited from the lossless convexification method, and free-time problems are difficult to convexify using the proposed method. The line search method used in Ref. [6] can be adopted to find the optimal terminal time, at the expense of adding another iteration loop, which reduces the algorithm's efficiency. Second, nonconvex state path constraints are difficult to enforce by ICP. Because the dependencies between the nonconvex constraint functions and the optimal variables will be eliminated under the proposed convexification strategy, which causes both sides of the inequality constraint to be constants.

A practicable approach for circumventing the abovementioned issues is to complement ICP with a classic successive convexification method, such as the pseudospectral-improved successive convexification (PiSC) algorithm developed in Ref. [34], which can address free-time problems and various path constraints. For online and closed-loop guidance applications [36], the PiSC algorithm can be used when the path constraints are about to be active, and the optimal time of flight can also be provided by PiSC, while in other flight phases, the ICP algorithm can be used to enhance computational efficiency and autonomy.

Numerical Experiments

In this section, numerical experiments are performed to demonstrate the performance of the proposed algorithm. The convergence, computational efficiency, and solution accuracy of the ICP algorithm are analyzed. In addition, to verify the performance of the ICP algorithm, the numerical results are compared with those obtained by PiSC and GPOPS-II [24]. As stated in Ref. [34], PiSC is a linearization-based successive convexification algorithm. GPOPS-II is an optimization software package for the Radau pseudospectral method and NLP methods.

All programs in this paper are coded using MATLAB 2016a and executed on a laptop with an Intel Core i7-8550U 1.80 GHz CPU. To enhance numerical stability and robustness, the nondimensionalization method described in Ref. [20] is used in the three algorithms. The SOCP problems produced by ICP and PiSC are solved using the MOSEK solver [3]. The numbers of collocation points used in ICP and PiSC are both set to 150, which is more than sufficient for the landing problem studied in this paper, and it should be noted that using fewer collocation points will significantly alleviate the computational burden. The hp-adaptive mesh refinement is performed, and the SNOPT solver [12] is used in GPOPS-II. The termination criteria of the three algorithms are all set to 10^{-5}.

The rocket parameters used in this paper are reported in Table 1. The initial position and velocity vectors are set to $\boldsymbol{r}_0 = [-2768, 0, 9830]$(m) and $\boldsymbol{V}_0 = [212, 0, -459]$(m/s), and the terminal vectors are $\boldsymbol{r}_f = \boldsymbol{0}$ and $\boldsymbol{V}_f = \boldsymbol{0}$. The time of flight is set to 33.8 s. The homotopy steps used in ICP are set to $\omega_1 = 0.1$ and $\omega_2 = 0.4$.

With the above parameter settings, there are eight iterations (including the starter iteration) in ICP until the solution converges. The converged terminal rocket mass is 29,664.7 kg. The total CPU time is 2.07 s, of which 1.82 s is used to solve the SOCP problems and 0.25 s is used for transaction processing and data I/O. The CPU time of every iteration and the convergence process are shown in Figs. 4 and 5.

Table 1 Rocket parameters

Variable	m_0(kg)	m_{dry}(kg)	$T_{\text{min, max}}$(kN)	I_{sp}(s)	S_{ref}(m^2)	C_D
Value	35,000	25,000	300, 860	400	10.75	0.5

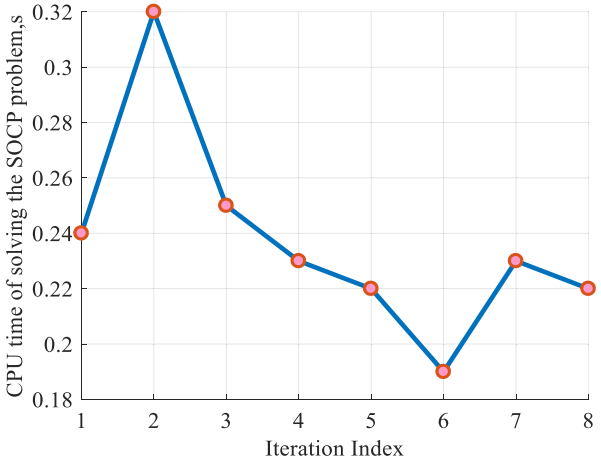

Fig. 4 CPU time by iteration

Figure 4 shows that the average CPU time required to solve the SOCP problems in ICP is 0.228 s, while the greatest CPU time (0.32 s) is required by the second iteration, in which the aerodynamic drag is first added to the problem. Figure 5 illustrates the variation in the average value of non-dimensionalized $|\Delta \boldsymbol{x}|$, which is defined in Algorithm 1. It can be observed that convergence is fast and smooth, and in the last few iterations, the accuracy is already quite high (approximately or below 10^{-4}) before the termination criterion is met.

The convergence history of the state and the control trajectories are shown from Figs. 6, 7, 8, 9, 10, 11, 12 and Fig. 13, while Fig. 14 is a local zoom plot of the thrust trajectory around the bang-bang switching point. Figure 15 illustrates the exact coincidence of the slack variable and the thrust magnitude. The r_y, V_y, and T_y trajectories are consistently kept as zeros, and they are not shown here.

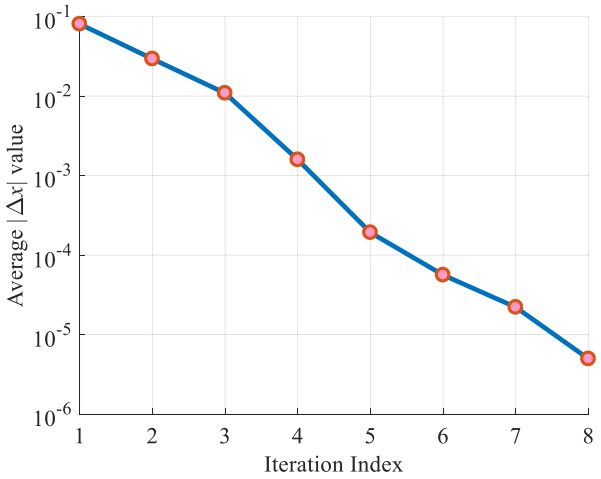

Fig. 5 Average convergence process

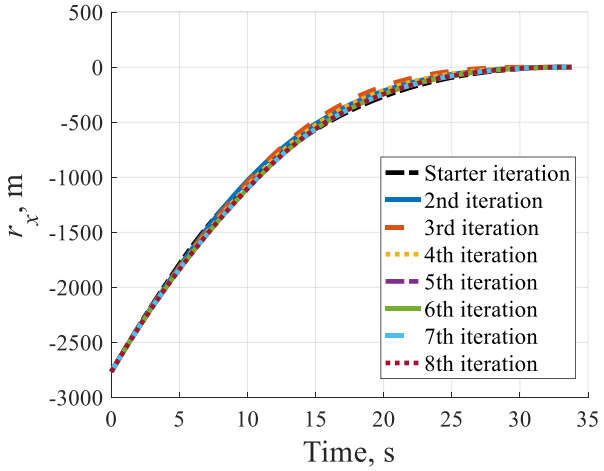

Fig. 6 Iteration history of position r_x

The figures show that no major differences between successive solutions occur after the fourth iteration, which confirms the rapidness of convergence. When the drag force is first added to the problem in the second iteration (solid blue line), the thrust magnitude constraint is not satisfied, as shown in Figs. 11 and 14. The reason is that the bound values of the thrust magnitude in Eq. (20) are approximated by the mass trajectory of the starter solution. The locations of the switching points of the bang-bang thrust trajectory are significantly different between the non-drag flight and the with-drag flight (as shown in Fig. 14); thus, the mass trajectory of the starter solution is significantly different from that of the converged solution, as shown in Fig. 10. After the fourth iteration, the drag force is fully added, and the mass trajectory is sufficiently accurate to guarantee that the thrust constraint is satisfied.

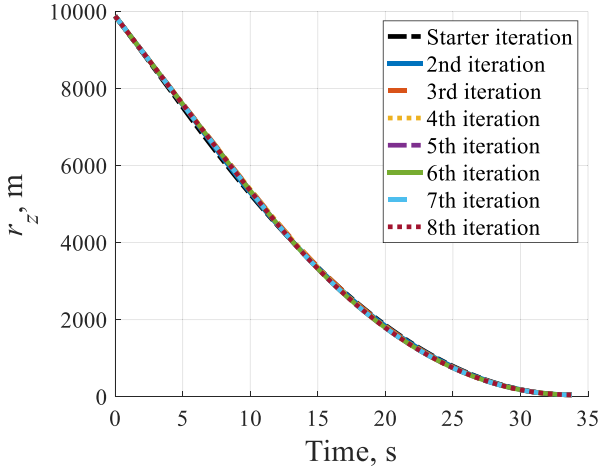

Fig. 7 Iteration history of position r_z

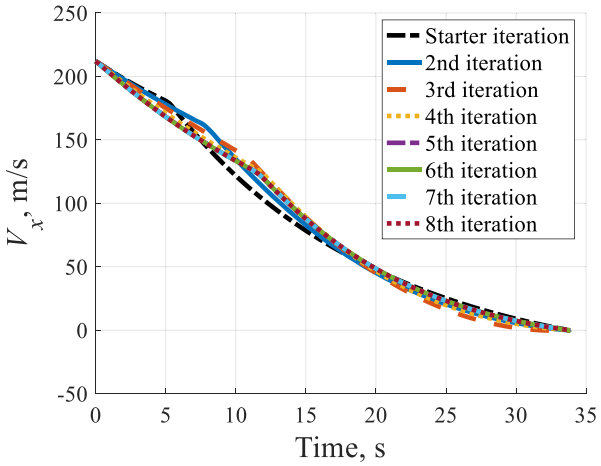

Fig. 8 Iteration history of velocity V_x

In the simulation, the homotopy step ω_2 is set to 0.4, which is quite large, and the drag force is fully added within four iterations. If ω_2 were set to a greater value, the second iteration would be infeasible, and more iterations would be needed to adjust the homotopy parameter, as indicated in Algorithm 1. If ω_2 were set to a smaller value, more iterations would be required until the drag force would be fully added.

Because the homotopy parameter gradually introduces the drag force into the problem, the optimal value of the homotopy step ω_2 is directly related to the magnitude of the drag force. Basically, a small step must be adopted to accommodate a large drag force, and a large step can be used to accelerate convergence when the drag force is small. Using the parameters listed in Table 1, when other parameters remain unchanged, in our experience, the ICP algorithm can adopt drag coefficient values ranging from 0 to 1.22.

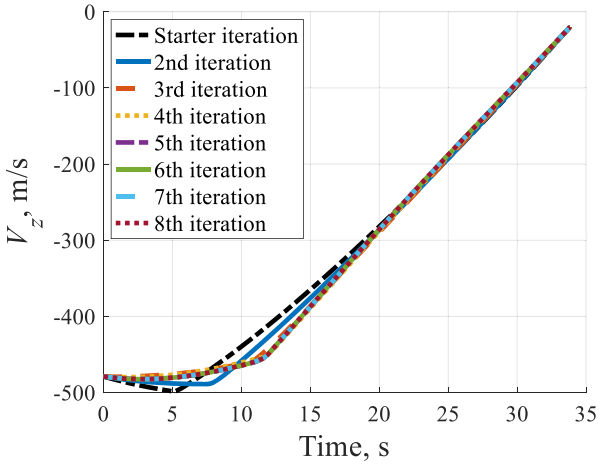

Fig. 9 Iteration history of velocity V_z

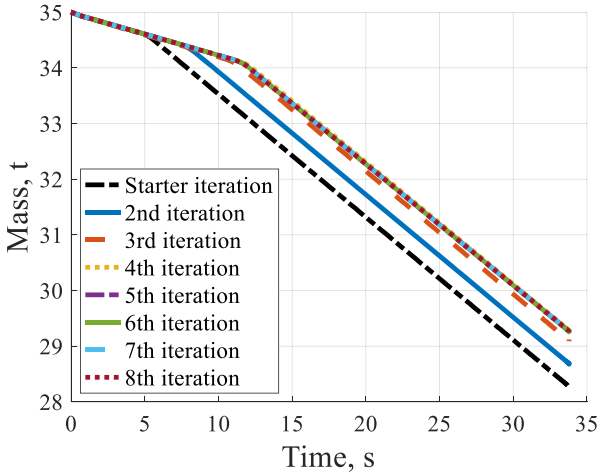

Fig. 10 Iteration history of vehicle mass

Next, the ICP solution is compared with the solutions obtained via PiSC and GPOPS-II. The algorithm parameters used in PiSC are tuned to optimize the performance for the current problem. Each of the three algorithms is performed 100 times, and the average results are listed in Table 2. The obtained state and control trajectories are nearly identical for the three algorithms, while the computational efficiency and the performance index of ICP outperform those of PiSC and GPOPS-II. Regarding computational efficiency, which is one of the most important factors for evaluating online algorithms, the CPU time of ICP is only 3.53% of that of GPOPS-II and 61.49% of that of PiSC.

To further demonstrate the solution accuracy of the ICP algorithm, we extract the optimized control trajectories obtained by the three algorithms and use them to propagate the nonlinear dynamical equations Eqs. (1)–(3). The fourth-order Runge-Kutta method and a 0.01 s time step are adopted. The termination

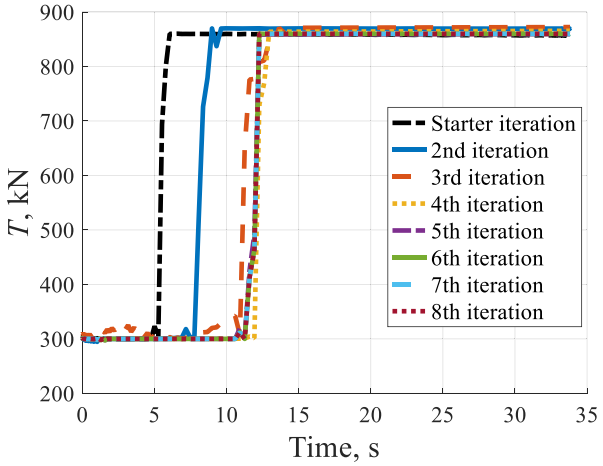

Fig. 11 Iteration history of thrust magnitude

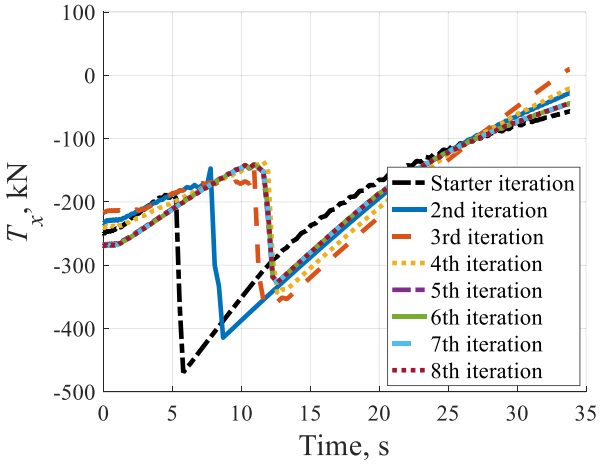

Fig. 12 Iteration history of thrust component T_x

condition of propagation is that the propagated r_z reaches 0 m. The deviations (denoted as δ_Δ for the variable Δ) between the optimized terminal state and the propagated terminal state are reported in Table 2. It can be observed that the solution accuracies of the three algorithms are acceptable and comparable. When a closed-loop guidance strategy is adopted, the above methodical errors can be eliminated by the receding horizon implementation, as in Ref. [36].

Conclusions

The rocket Earth-landing trajectory optimization problem is studied in this paper. Using the proposed iterative convex programming algorithm, the problem can be

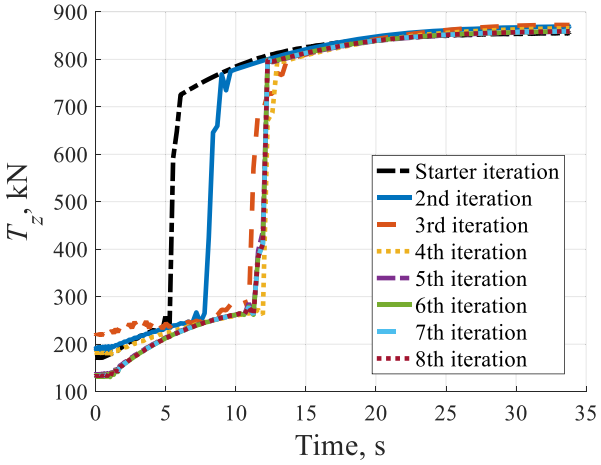

Fig. 13 Iteration history of thrust component T_z

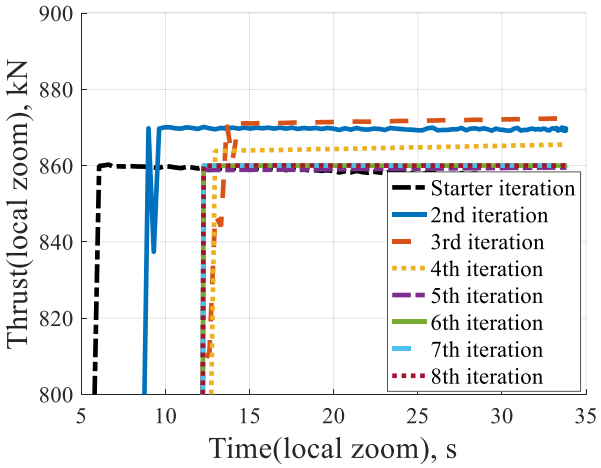

Fig. 14 Iteration history of thrust magnitude (local zoom)

solved rapidly, accurately, and automatically. Based on the characteristics of the rocket, a homotopic-iterative strategy is proposed to convexify the associated nonlinear dynamics. In addition to the rapid and deterministic convergence properties inherited from convex optimization, the algorithm does not rely on any reference trajectory or initial guess; thus, the algorithm is fully automated, which is desirable for online applications. In addition, the implementation and tuning of the algorithm are simple and straightforward. The numerical results demonstrate that the computational efficiency and solution accuracy of the algorithm are satisfactory. The method developed in this paper exhibits the potential to be used online and provides an option for rocket landing guidance applications.

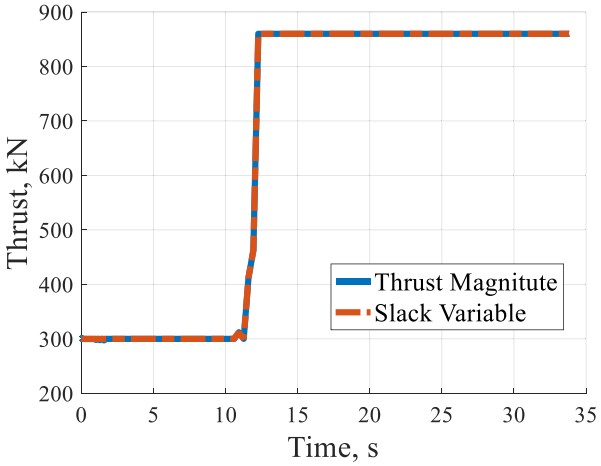

Fig. 15 Converged thrust magnitude and slack variable

Table 2 Comparison of the three algorithms (averaged over 100 trials)

Algorithm	CPU time, s	m_f, kg	$\delta_{V_{x\_f}}$, m/s	$\delta_{V_{z\_f}}$, m/s	$\delta_{r_{x\_f}}$, m	δ_{m_f}, kg
ICP	1.98	29,664.7	0.11	0.08	2.24	4.19
PiSC	3.22	29,492.5	0.14	0.35	3.11	6.46
GPOPS-II	56.12	29,611.3	0.10	0.06	1.92	3.47

Funding This research is supported by the Fundamental Research Funds for the Central Universities (191gpy288).

Data Availability Not applicable.

Compliance with Ethical Standards

Conflicts of Interest/Competing Interests No conflict of interest exits in this paper.

Code Availability Not applicable.

References

1. Açıkmeşe, B., Ploen, S.R.: Convex programming approach to powered descent guidance for Mars landing. J. Guid. Control. Dyn. **30**, 1353–1366 (2007)
2. Açıkmeşe, B., Carson, J., Blackmore, L.: Lossless Convexification of nonconvex control bound and pointing constraints of the soft landing optimal control problem. IEEE Trans. Control Syst. Technol. **21**, 2104–2113 (2013)
3. Andersen, E.D., Roos, C., Terlaky, T.: On implementing a primal-dual interior-point method for conic quadratic optimization. Math. Program. **95**, 249–277 (2003)
4. Bai, X., Turner, D., Junkins, J.: Bang-bang Control Design by Combing Pseudospectral Method with a novel Homotopy Algorithm. AIAA Guidance, Navigation, and Control Conference, pp. 2009–5955. AIAA Paper, Chicago (2009)
5. Betts, J.T.: Survey of numerical methods for trajectory optimization. J. Guid. Control. Dyn. **21**, 193–207 (1998)
6. Blackmore, L., Açıkmeşe, B., Scharf, D.P.: Minimum-landing-error powered-descent guidance for Mars landing using convex optimization. J. Guid. Control. Dyn. **33**, 1161–1171 (2010)
7. Bollino, K.P.: High-Fidelity Real-Time Trajectory Optimization for Reusable Launch Vehicles. Monterey California USA, Naval Postgraduate School (2006)
8. Brendel, E., Hérissé, B., Bourgeois, E.: Optimal Guidance for Toss Back Concepts of Reusable Launch Vehicles. 8th European Conference for Aeronautics and Aerospace Science, Madrid (2019)
9. Cheng, X., Li, H., Zhang, R., Cheng, X., Li, H., Zhang, R.: Efficient Ascent Trajectory Optimization using Convex Models based on the Newton–Kantorovich/Pseudospectral Approach. Aerosp. Sci. Technol. **66**, 140–151 (2017)
10. Dueri, D., Açıkmeşe, B., Scharf, D.P., Harris, M.: Customized real time interior-point methods for onboard powered-descent guidance. J. Guid. Control. Dyn. **40**, 197–212 (2017)
11. Garg, D.: Advances in Global Pseudospectral Methods for Optimal Control. Univ. of Florida, Gainesville Florida (2011)
12. Gill, P., Murray, W., Saunders, M.: SNOPT: an SQP algorithm for large-scale constrained optimization. SIAM Rev. **47**, 99–131 (2005)
13. Guo, T., Jiang, F., Li, J.: Homotopic approach and Pseudospectral method applied jointly to low thrust trajectory optimization. Acta Astronaut. **71**, 38–50 (2012)

14. Harris, M.W., Açıkmeşe, B.: Maximum divert for planetary landing using convex optimization. J. Optim. Theory Appl. **162**, 975–995 (2013)
15. Huntington, G.T.: Advancement and Analysis of a Gauss Pseudospectral Transcription for Optimal Control. Cambridge Massachusetts USA. Massachusetts Inst. of Technology, USA (2007)
16. Liu, X., Shen, Z., Lu, P.: Entry trajectory optimization by second order cone programming. J. Guid. Control. Dyn. **39**, 227–241 (2016)
17. Liu, X., Lu, P., Pan, B.: Survey of convex optimization for aerospace applications. Astrodynamics. **1**, 23–40 (2017)
18. Liu, X.: Fuel-optimal rocket landing with aerodynamic controls. J. Guid. Control. Dyn. **42**, 65–77 (2018)
19. Lu, P., Sun, H., Tsai, B.: Closed-loop Endoatmospheric ascent guidance. J. Guid. Control. Dyn. **26**, 283–294 (2003)
20. Lu, P.: Entry guidance: a unified method. J. Guid. Control. Dyn. **37**, 713–728 (2014)
21. Lu, P.: Introducing computational guidance and control. J. Guid. Control. Dyn. **40**, 193–193 (2017)
22. Mao, Y., Szmuk, M., Açıkmeşe, B.: Successive Convexification of Non-Convex Optimal Control Problems and its Convergence Properties. IEEE 55th Conference on Decision and Control (CDC), pp. 3636–3641. IEEE Publ, Piscataway (2016)
23. Pan, B., Lu, P., Pan, X., Ma, Y.: Double-Homotopy method for solving optimal control problems. J. Guid. Control. Dyn. **39**, 1706–1720 (2016)
24. Patterson, M.A., 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 Trans. Math Software. **41**, 1–37 (2014)
25. Ross, I.M., Sekhavat, P., Fleming, A., Gong, Q.: Optimal feedback control: foundations, examples, and experimental results for a new approach. J. Guid. Control. Dyn. **31**, 307–321 (2008)
26. Sagliano, M., Mooij, E., Theil, S.: Onboard trajectory generation for entry vehicles via adaptive multivariate Pseudospectral interpolation. J. Guid. Control. Dyn. **40**, 466–476 (2017)
27. Sagliano, M.: Pseudospectral convex optimization for powered descent and landing. J. Guid. Control. Dyn. **41**, 320–334 (2018)
28. Sagliano, M.: Generalized hp Pseudospectral-convex programming for powered descent and landing. J. Guid. Control. Dyn. **42**, 1562–1570 (2019)
29. Sagliano, M., Mooij, E.: Optimal Drag-Energy Entry Guidance via Pseudospectral Convex Optimization. AIAA Guidance, Navigation, and Control Conference, pp. 2018–1871. AIAA Paper, Kissimmee (2018)
30. Simplício, P., Andrés, M., Samir, B.: Guidance of reusable launchers: improving descent and landing performance. J. Guid. Control. Dyn. **42**, 2206–2219 (2019)
31. Simplício, P., Andrés, M.: Reusable launchers: development of a coupled flight mechanics, guidance, and control benchmark. J. Spacecraft. Rockets. **57**, 74–89 (2020)
32. Scharf, D.P., Açıkmeşe, B., Dueri, D., Benito, J., Casoliva, J.: Implementation and experimental demonstration of onboard powered descent guidance. J. Guid. Control. Dyn. **40**, 213–229 (2017)
33. Szmuk, M., Reynolds, T., Açıkmeşe, B.: Successive Convexification for real-time six-degree-of-freedom powered descent guidance with state-triggered constraints. J. Guid. Control. Dyn. **43**, 1399–1413 (2020)
34. Wang, J., Cui, N.: A Pseudospectral-Convex Optimization Algorithm for Rocket Landing Guidance. AIAA Guidance, Navigation, and Control Conference, pp. 2018–1871. AIAA Paper, Kissimmee (2018)
35. Wang, J., Cui, N., Wei, C.: Rapid trajectory optimization for hypersonic entry using convex optimization and Pseudospectral method. Aircr. Eng. Aerosp. Technol. **91**, 669–679 (2019a)
36. Wang, J., Cui, N., Wei, C.: Optimal rocket landing guidance using convex optimization and model predictive control. J. Guid. Control. Dyn. **42**, 1078–1092 (2019b)
37. Yang, H., Li, S., Bai, X.: Fast Homotopy method for asteroid landing trajectory optimization using approximate initial Costates. J. Guid. Control. Dyn. **42**, 585–597 (2019)
38. Zhou, D., Zhang, Y., Li, S.: Receding horizon guidance and control using sequential convex programming for spacecraft 6-DoF close proximity. Aerosp. Sci. Technol. **87**, 459–477 (2019)