Robust powered descent guidance considering mass and fuel consumption uncertainties: A convex optimization approach

Chinese Journal of Aeronautics, (2026), 39(3): 103914

Chinese Society of Aeronautics and Astronautics
& Beihang University

Chinese Journal of Aeronautics

cja@buaa.edu.cn
www.sciencedirect.com

FULL LENGTH ARTICLE

Robust powered descent guidance considering mass and fuel consumption uncertainties: A convex optimization approach

Duozhi GAO [a], Yanning GUO [a,b], Edoardo FADDA [c], Youmin GONG [d,*], Chuanjiang LI [a], Paolo BRANDIMARTE [c]

[a] *Department of Control Science and Engineering, Harbin Institute of Technology, Harbin 150001, China*
[b] *Zhengzhou Research Institute, Harbin Institute of Technology, Zhengzhou 450000, China*
[c] *Department of Mathematical Sciences 'Giuseppe Luigi Lagrange', DISMA, Politecnico di Torino, Torino 10129, Italy*
[d] *School of Intelligence Science and Engineering, Harbin Institute of Technology (Shenzhen), Shenzhen 518055, China*

Received 24 January 2025; revised 19 March 2025; accepted 18 May 2025
Available online 1 November 2025

KEYWORDS

Uncertainty;
Trajectory optimization;
Convex optimization;
Powered descent;
Closed-loop guidance

Abstract This paper addresses the challenge of mass uncertainty during the powered descent phase of a Mars lander and proposes a robust powered descent guidance algorithm that accounts for uncertainties in mass and fuel consumption. First, the traditional trajectory optimization method based on convex optimization is improved by developing a fast and accurate solution approach using sequential convex optimization. Second, the effects of mass uncertainty on position are modeled and analyzed, with corresponding computational methods provided for different scenarios. Third, the worst-case scenario under mass uncertainty is analyzed through both geometric and theoretical approaches, and a modified glide-slope constraint method is proposed to ensure safe landing even in adverse conditions. Moreover, a closed-loop receding horizon based guidance is developed to further mitigate the effects of mass uncertainty and improve terminal landing accuracy. Finally, the proposed improved convex optimization algorithm and robust trajectory optimization algorithm are validated through simulation cases and compared with a probabilistic approach. The simulations further test various initial positions, velocities, and glide-slope angles, demonstrating that the solutions are both accurate and robust.

* Corresponding author.
E-mail address: gongyoumin@hit.edu.cn (Y. GONG).
Peer review under responsibility of Editorial Committee of CJA.

Production and hosting by Elsevier

1. Introduction

In recent years, the rapid advancement of space technology has significantly heightened global interest in planetary surface exploration, particularly targeting celestial bodies such as the Moon, Mars, and asteroids. Concurrently, the development

https://doi.org/10.1016/j.cja.2025.103914

of reusable launch vehicles has emerged as a transformative innovation, capturing widespread attention and reshaping the future of space exploration.[1–4] These missions all include a powered descent phase, that is, deceleration and control with engines during the final approach to the surface. Among them, missions such as manned lunar landing,[5] Mars surface exploration[6] and asteroid sampling[7] require safe landing on planetary surfaces and then carrying out scientific research. For reusable rockets,[8] a soft landing to a retrieval point is required. These high-value landing missions bring safer and more robust requirements for power descent guidance.

During the powered descent phase, the engines decelerate and control the lander by consuming fuel to generate thrust. However, the fuel carried by the lander and its consumption rate are difficult to measure precisely.[9] For instance, the initial fuel mass for the powered descent of a reusable rocket depends on the remaining mass from the launch stage, and the fuel consumption during Mars landings cannot be accurately determined. Mass estimation errors can lead to deviations in the lander's acceleration, resulting in errors in velocity and position. In severe cases, this can cause collisions with obstacles or the surface, ultimately leading to mission failure. Therefore, developing a powered descent guidance algorithm that accounts for uncertainties in mass is critically important. Additionally, the powered descent phase represents the final stage in planetary surface exploration missions. During this phase, the main engine of the lander is ignited to provide thrust for deceleration and guidance, consuming a significant amount of fuel. Thus, reducing fuel consumption during the powered descent phase can allow for heavier scientific payloads, thereby improving the overall mission efficiency.[10]

Currently, guidance methods for the powered descent phase can currently be categorized into three main types[11]: analytical guidance,[12] numerical optimization-based guidance,[13,14] and learning-based guidance.[15] Analytical methods, such as Zero-Effort-Miss/ Zero-Effort-Velocity (ZEM/ZEV) guidance,[16] polynomial guidance, and convex curvature guidance,[17,18] are representative examples of this category. These methods are known for their low computational demand and high processing speed. However, they struggle to handle complex process constraints and often fail to achieve optimal fuel consumption. In contrast, numerical optimization-based methods, including convex programming and pseudospectral optimization, provide more optimal solutions.[19] Among these, convex optimization stands out due to its rapid solution capabilities, enabling real-time onboard computation. Notably, SpaceX employs this technology in the landing guidance of its reusable rockets, highlighting its significant potential for practical applications. More recently, learning-based guidance has garnered increasing attention, particularly reinforcement learning-based methods.[15] However, practical application of such methods remains constrained by issues of stability and reliability.

The aforementioned powered descent guidance methods are primarily designed for deterministic parameters. However, certain parameters are difficult to measure or estimate precisely,[20] and these uncertainties pose significant challenges to guidance and control systems.[21,22] To address these challenges, researchers have developed various approaches to handle uncertainties, which can be broadly classified into parameter uncertainty methods, probabilistic methods, set-based techniques, and learning-based strategies. Parameter uncertainty methods,[23] including robust control and adaptive feedback control, offer systematic frameworks for addressing bounded parameter variations and external disturbances. However, these approaches often face limitations in effectively handling complex constraints and optimizing performance metrics. Probabilistic methods, such as polynomial chaos expansion[24,25] and covariance control,[26] explicitly model the statistical properties of uncertainties, enabling more nuanced and risk-aware decision-making. Nonetheless, they rely heavily on accurate prior knowledge of uncertainty distributions, which may not always be available or reliable. Set-based techniques, including funnel scheduling[27] and tube-based model predictive control,[28] offer strong guarantees for constraint satisfaction under bounded uncertainties. These methods are effective at capturing reachable sets and ensuring safety, but they often require substantial computational resources to propagate uncertainty sets over time. Additionally, they are typically combined with feedback control mechanisms to ensure system stability. Recently, learning-based approaches have gained attention for uncertainty handling,[29,30] leveraging advancements in reinforcement learning and Gaussian process regression to adapt to complex or unknown system dynamics. While promising, these methods usually demand large volumes of data and extensive training time, and often lack formal guarantees on safety and performance. Each of these methods has distinct strengths and limitations, leading to the development of hybrid approaches that combine multiple strategies to balance performance and robustness.[31]

Among the various sources of uncertainty, mass uncertainty is especially critical during the powered descent phase, particularly in Mars landings and reusable rocket missions. This uncertainty can be further divided into fuel quantity uncertainty and fuel consumption rate uncertainty. While probabilistic methods are effective for quantifying uncertainty, they often require extensive data to derive accurate probability distributions, which can be impractical in many real-world scenarios.[32] To overcome this challenge, this study models uncertainties within a bounded, closed set framework.

Inspired by robust optimization principles,[33] this paper proposes a worst-case robust optimization framework that directly incorporates bounded uncertainties into a convex formulation. This approach offers a tractable, non-probabilistic, and safety-focused alternative that ensures robust performance without requiring prior knowledge of uncertainty distributions, large datasets, or intensive computational resources. The proposed framework functions as an open-loop trajectory optimization method. When integrated with receding horizon techniques to form a closed-loop guidance algorithm, it guarantees constraint satisfaction throughout the entire computation process. Furthermore, the theoretical analysis of worst-case scenarios demonstrates that the framework is extensible to cases involving uncertainties in initial position and velocity. This underscores the method's scalability and adaptability, making it well-suited for a wider range of uncertainty scenarios in practical guidance and control applications.

This paper introduces an algorithm that improves the traditional convex optimization framework while considering the initial mass and fuel consumption uncertainties to tackle the

trajectory optimization challenge of achieving safe landing under mass uncertainties. The innovations of this paper are summarized as follows.

(1) An improved convex optimization method based on sequential convex optimization is proposed to reduce the linearization error in the lossless convexification process. The improved algorithm achieves higher solution accuracy and better optimality.
(2) An open-loop robust trajectory optimization approach is developed to address mass uncertainty. Through geometric analysis and theoretical exploration of worst-case scenarios, the method ensures a safe landing even under the most adverse conditions. Theoretical analysis shows that the uncertainties in initial position and velocity can be incorporated using the proposed approach.
(3) A closed-loop receding horizon based guidance is developed to further mitigate the effects of mass uncertainty and improve terminal landing accuracy. This approach combines the computational efficiency of convex optimization with the robustness of the proposed framework, ensuring that system constraints are not violated throughout the guidance process.

2. Problem formulation for fuel optimal powered descent guidance

In this section, a mathematical formulation of the fuel optimal powered descent landing scenario within a Three-Degrees-of-Freedom (3-DoF) framework is presented. First, in Section 2.1, the underlying assumptions are outlined, and the reference coordinate system is defined. Next, the kinematic and dynamic models are established, followed by a discussion of the associated constraints governing the entire process. Then, in Section 2.2, the mathematical models of uncertainties are developed. Finally, in Section 2.3, a comprehensive formulation of the fuel optimal powered descent guidance problem is provided.

2.1. Mathematical model of dynamics and constraints

This study is based on a 3-DoF model, focusing solely on position and velocity. During the powered descent phase, the lander is treated as a point mass subjected to various forces, including gravity, aerodynamic drag, and thrust from the thrusters. In this paper, aerodynamic drag is considered negligible compared to gravity and thrust, and thus its effects are therefore ignored. The landing scenarios considered involve planets with uniform gravitational fields, such as the Earth and Mars. Given the relatively low initial altitude during the powered descent phase (typically within 5 km) and the brief duration of the descent process (on the order of a few minutes), it is reasonable to assume that the gravitational field remains constant and to neglect the effects of planetary rotation. It is assumed that the thrusters are activated at the initial time t_0 and can only be shut off at the final time t_f. The thrust magnitude can be adjusted within a range from minimum to maximum, with the minimum thrust being greater than zero. Additionally, it is assumed that fuel consumption is linearly proportional to the thrust magnitude.

The coordinate system is defined as follows: the surface inertial coordinate system is denoted as $O_I xyz$, with its origin located at the expected landing point. The x-axis points upward, the y-axis points eastward, and the z-axis points northward. A schematic representation of this coordinate system is provided in Fig. 1.

In the surface inertial coordinate system $O_I xyz$, the position of the lander is denoted as $\boldsymbol{r} = \left[r_x, r_y, r_z\right]^{\mathrm{T}} \in \mathbb{R}^3$, and its velocity vector as $\boldsymbol{v} = \left[v_x, v_y, v_z\right]^{\mathrm{T}} \in \mathbb{R}^3$. The thrust force acting on the lander is represented by $\boldsymbol{T} = \left[T_x, T_y, T_z\right]^{\mathrm{T}} \in \mathbb{R}^3$. Additionally, the gravitational acceleration is denoted as $\boldsymbol{g}_{\mathrm{m}}$. The translation dynamics of the lander is:

$$\begin{cases} \dot{\boldsymbol{r}}(t) = \boldsymbol{v}(t) \\ \dot{\boldsymbol{v}}(t) = \boldsymbol{g}_{\mathrm{m}} + \frac{\boldsymbol{T}(t)}{m(t)} \end{cases} \tag{1}$$

Considering the assumptions of fuel consumption, the relationship between the mass variation $\dot{m}(t)$ and the magnitude of thrust $\| \boldsymbol{T}(t) \|_2$ is given by:

$$\begin{cases} \dot{m}(t) = -\alpha \| \boldsymbol{T}(t) \|_2 \\ \alpha = \bar{\alpha} + \Delta\alpha \end{cases} \tag{2}$$

where α is the fuel consumption coefficient, $\bar{\alpha}$ is the nominal value of the fuel consumption coefficient, $\Delta\alpha$ is the uncertainty component of the fuel consumption coefficient. The fuel consumption coefficient $\bar{\alpha}$ is calculated based on the specific impulse I_{sp}, the Earth's gravitational acceleration g_e, and the specific thruster configuration.[34]

The constraints are divided into state constraints and control constraints. State constraints are further categorized into two major types: initial and final state constraints, and process state constraints. Specifically, at the initial and final times, the lander is subject to the following state constraints:

$$\begin{cases} \boldsymbol{r}(t_0) = \boldsymbol{r}_0, \ \boldsymbol{v}(t_0) = \boldsymbol{v}_0, \ m(t_0) = m_{\mathrm{wet}} \\ \boldsymbol{r}(t_{\mathrm{f}}) = \boldsymbol{r}_{\mathrm{f}}, \ \boldsymbol{v}(t_{\mathrm{f}}) = \boldsymbol{v}_{\mathrm{f}}, \ m(t_{\mathrm{f}}) \geqslant m_{\mathrm{dry}} \end{cases} \tag{3}$$

where m_{wet} is the initial total mass including both the lander and the fuel, m_{dry} is the mass of the lander, $m(t_0)$ and $m(t_f)$ are the initial and final masses respectively, $\boldsymbol{r}_0$ and $\boldsymbol{r}_f$ are the initial and final positions respectively, and $\boldsymbol{v}_0$ and $\boldsymbol{v}_f$ are the initial and final velocities respectively.

Regarding process state constraints, obstacle avoidance is necessary during the powered descent phase. To enhance obstacle avoidance capabilities, a conical glide-slope constraint

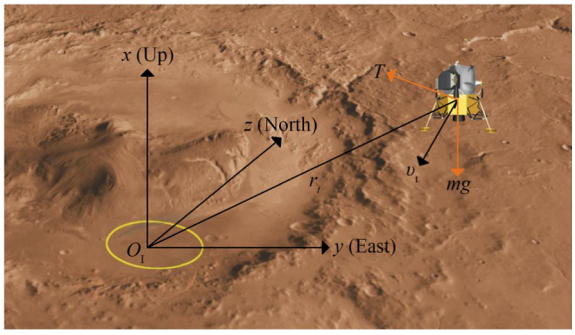


Fig. 1 Coordinate system and powered descent scenario.

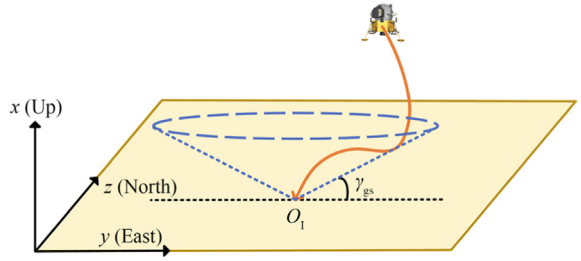


Fig. 2 Illustration of glide-slope constraint.

is introduced. As shown in Fig. 2, this is mathematically represented by a state constraint as follows:

$$\tan\gamma_{\rm gs} \leqslant \frac{r_x}{\sqrt{r_y^2+r_z^2}} \tag{4}$$

where, γ_{gs} denotes the glide-slope constraint angle. r_x, r_y, and r_z represent the components of the position vector $\boldsymbol{r}$.

In addition, control constraints arise from the physical characteristics and assumptions of the thrusters. The constraints on thrust are as follows:

$$0 < T_{\min} \leqslant \| \boldsymbol{T}(t) \|_2 \leqslant T_{\max} \tag{5}$$

where $T_{\min}$ is a positive value, indicating that the engines cannot be shut down during the descent process and must be started from the beginning. $T_{\max}$ represents the maximum thrust the thrusters can provide.

2.2. Mathematical model of mass and fuel consumption uncertainties

Existing literature often treats uncertainty as a probability distribution,[24–26] such as a Gaussian distribution. However, determining the actual probability distribution can be challenging due to the lack of sufficient data to support its accuracy and the difficulty in identifying the exact type of probability distribution. Therefore, unlike the existing literature, this paper constrains uncertainty within a bounded set. As for the mass uncertainty, it can be expressed in the following form:

$$\begin{cases} m_k = \bar{m}_k + \Delta m_k \\ \mathscr{U}_{{\rm a},k} \leqslant \Delta m_k \leqslant \mathscr{U}_{{\rm b},k} \end{cases} \tag{6}$$

where, $\bar{m}_k$ is the nominal mass at time t_k, Δm_k represents the mass uncertainty at time t_k, $\mathscr{U}_{{\rm a},k}$ is the lower bound of the uncertainty at time t_k, and $\mathscr{U}_{{\rm b},k}$ is the upper bound of the uncertainty at time t_k.

2.2.1. Initial mass uncertainty

For initial mass uncertainty, it is assumed that the uncertainty is additive and bounded, this can be mathematically represented as:

$$\begin{cases} m_{\rm wet} = \bar{m}_{\rm wet} + \Delta m_{\rm wet} \\ \mathscr{W}_{\rm a} \leqslant \Delta m_{\rm wet} \leqslant \mathscr{W}_{\rm b} \end{cases} \tag{7}$$

Considering that the optimized thrust is fixed, the fuel consumption at each time step remains constant. Therefore, when the initial mass has uncertainty, the mass uncertainty Δm_k in Eq. (6) is converted to the following form:

$$\begin{cases} \Delta m_k = \Delta m_{\rm wet} \\ \mathscr{U}_{{\rm a},k} = \mathscr{W}_{\rm a},\ \mathscr{U}_{{\rm b},k} = \mathscr{W}_{\rm b} \end{cases} \tag{8}$$

2.2.2. Fuel consumption uncertainty

For fuel consumption uncertainty, we consider that the fuel consumption coefficient α has an uncertainty that is bounded and can be expressed mathematically as:

$$\begin{cases} m_{k+1} = m_k - \alpha \| \boldsymbol{T}_k(t) \|_2 \\ \alpha = \bar{\alpha} + \Delta\alpha \\ \mathscr{F}_{\rm a} \leqslant \Delta\alpha \leqslant \mathscr{F}_{\rm b} \end{cases} \tag{9}$$

Iterate Eq. (9) yields the expression for m_{k+1} with respect to m_0 as:

$$\begin{aligned} m_k &= m_{k-1} - (\bar{\alpha}+\Delta\alpha)\| \boldsymbol{T}_{k-1}(t) \|_2 \\ &= m_{k-2} - (\bar{\alpha}+\Delta\alpha)\big(\| \boldsymbol{T}_{k-1}(t) \|_2 + \| \boldsymbol{T}_{k-2}(t) \|_2\big) \\ &= m_0 - (\bar{\alpha}+\Delta\alpha)\sum_{i=0}^{k-1} \| \boldsymbol{T}_i(t) \|_2 \end{aligned} \tag{10}$$

Therefore, the mass uncertainty Δm_k due to fuel consumption uncertainty is:

$$\Delta m_k = -\Delta\alpha\big(\| \boldsymbol{T}_{k-1}(t) \|_2 + \cdots + \| \boldsymbol{T}_1(t) \|_2 + \| \boldsymbol{T}_0(t) \|_2\big) \tag{11}$$

2.2.3. Combined uncertainty regarding initial mass and fuel consumption

When there is uncertainty in both initial mass and fuel consumption, combining Eqs. (8) and (11), this mixing uncertainty can be mathematically expressed as:

$$\Delta m_k = \Delta m_{\rm wet} - \Delta\alpha\sum_{i=0}^{k-1} \| \boldsymbol{T}_i(t) \|_2 \tag{12}$$

Thus, for these three cases of mass uncertainty, the mass of uncertainty Δm_k is obtained in Eqs. (8), (11), and (12), respectively.

2.3. Fuel optimal problem formulation

In this paper, we aim to minimize fuel consumption during powered descent while achieving a pinpoint landing and satisfying state and control constraints. Therefore, the optimization objective for this problem is the total fuel consumption. The optimized variables are the magnitude and direction of thrust. The fuel optimal powered descent guidance problem is described as an optimal control problem as follows

Problem 1. Fuel optimal powered descent problem

$$\begin{aligned} &\min_{\boldsymbol{T}(t)} -m(t_{\rm f}) \Longleftrightarrow \min_{\boldsymbol{T}(t)} \int_0^{t_{\rm f}} \| \boldsymbol{T}(t) \|_2 {\rm d}t \\ &\text{subject to}: \ \text{Eqs. } (1)-(5) \end{aligned} \tag{13}$$

Problem 1 describes the fuel optimal powered descent guidance problem, which is nonconvex in nature. An effective approach to address this challenge is to transform the original nonconvex problem into a second-order cone programming problem through lossless convexification.[34,35]

3. Improved lossless convexification and problem formulation considering uncertainties in initial mass and fuel consumption

In the previous section, a mathematical description of the fuel optimal powered descent guidance problem was established. Due to the nonlinear dynamics constraints and the minimum thrust amplitude constraint, this optimization problem is inherently non-convex. In Section 3.1, an improved lossless convex optimization method was proposed to transform the non-convex problem into a convex one.

Furthermore, in Section 3.2, after developing the improved lossless convexification method for the nominal problem, the uncertainties in initial mass and fuel consumption were considered. The analytical form of the impact of these uncertainties on the state variables was derived. Finally, the effect of these uncertainties on the powered descent trajectory was analyzed.

3.1. Improved lossless convexification

In this section, an improved lossless convex optimization method was proposed based on the lossless convexification.[34] Through variable substitution, Taylor expansion, and iterative correction, an exact convex optimization solution for the original nonconvex problem was obtained. Subsequently, by discretizing the continuous system, a discrete problem was derived that can be rapidly solved using a convex optimization solver. Additionally, a correction method based on sequential convex optimization is proposed in order to solve the error in the linearization process.

For the optimal control problem 1, the nonconvexity primarily arises from two sources: the minimum thrust constraint $T_{\min}$ and the nonlinear dynamics involving division. To address the minimum thrust constraint, a method introducing slack variables was adopted in Ref. 35 Consequently, Problem 1 can be transformed into the following relaxed form, as shown in Problem 2.

Problem 2. Relaxed fuel optimal powered descent problem

$$\min_{\boldsymbol{T}(t),\ \Gamma(t)} -m(t_{\mathrm{f}}) \Longleftrightarrow \min_{\boldsymbol{T}(t),\ \Gamma(t)} \int_0^{t_{\mathrm{f}}} \Gamma(t)\mathrm{d}t$$

$$\text{subject to}: \begin{cases} \dot{\boldsymbol{r}}(t) = \boldsymbol{v}(t) \\ \dot{\boldsymbol{v}}(t) = \boldsymbol{g_m} + \frac{\boldsymbol{T}(t)}{m(t)} \\ \dot{m}(t) = -\alpha\Gamma(t) \\ \| \boldsymbol{T}(t) \|_2 \leqslant \Gamma(t) \\ 0 < T_{\min} \leqslant \Gamma(t) \leqslant T_{\max} \\ \tan\gamma_{\mathrm{gs}} \leqslant \frac{r_x}{\sqrt{r_y^2+r_z^2}} \\ \boldsymbol{r}(t_0) = \boldsymbol{r}_0,\ \boldsymbol{v}(t_0) = \boldsymbol{v}_0,\ m(t_0) = m_{\mathrm{wet}} \\ \boldsymbol{r}(t_{\mathrm{f}}) = \boldsymbol{r}_{\mathrm{f}},\ \boldsymbol{v}(t_{\mathrm{f}}) = \boldsymbol{v}_{\mathrm{f}},\ m(t_{\mathrm{f}}) \geqslant m_{\mathrm{dry}} \end{cases} \tag{14}$$

where, $\Gamma(t)$ is the introduced slack variable. It has been proven in Refs. 34 and 35 that when the optimization objective is fuel optimality, the optimal solution to the relaxed problem is also the optimal solution to the original problem.

To eliminate the nonlinearity in the dynamic constraints involving division, a variable substitution approach is adopted as follows:

$$\begin{cases} z(t) = \ln m(t) \\ \boldsymbol{u}(t) = \frac{\boldsymbol{T}(t)}{m(t)} \\ \sigma(t) = \frac{\Gamma(t)}{m(t)} \end{cases} \tag{15}$$

The time derivative of z is obtained as follows:

$$\dot{z}(t) = \frac{\dot{m}(t)}{m(t)} = \frac{-\alpha\Gamma(t)}{m(t)} = -\alpha\sigma(t) \tag{16}$$

where, Eq. (16) is used to replace the fuel consumption in Eq. (14).

The ordinary differential equation for mass can be obtained as follows:

$$\frac{\dot{m}(t)}{m(t)} = -\alpha\sigma(t) \tag{17}$$

Solving the ordinary differential Eq. (17) yields the mass $m(t)$ as follows:

$$m(t) = m_{\mathrm{wet}} \mathrm{e}^{\int_0^t (-\alpha\sigma(\tau))\mathrm{d}\tau} \tag{18}$$

Consequently, the optimization objective in Eq. (14) can be converted into the following form:

$$\min_{\boldsymbol{u}(t),\sigma(t)} -z(t_{\mathrm{f}}) \Longleftrightarrow \min_{\boldsymbol{u}(t),\sigma(t)} \int_0^{t_{\mathrm{f}}} \sigma(t)\mathrm{d}t \tag{19}$$

For the thrust magnitude constraint in Eq. (14), after applying the variable substitution, it is transformed into the following form:

$$0 < \frac{T_{\min}}{m(t)} \leqslant \sigma(t) \leqslant \frac{T_{\max}}{m(t)} \tag{20}$$

which remains in a nonconvex form. From Eq. (15), we have $m(t) = \mathrm{e}^{z(t)}$. Therefore, Eq. (20) transforms to the following:

$$0 < T_{\min}\mathrm{e}^{-z(t)} \leqslant \sigma(t) \leqslant T_{\max}\mathrm{e}^{-z(t)} \tag{21}$$

To transform the constraint into a convex form, expand Eq. (21) using a Taylor series. Perform a second-order expansion on the left-hand side of the inequality and a first-order expansion on the right-hand side, resulting in the following form:

$$\begin{cases} T_{\min}\mathrm{e}^{-z_l(t)}\left[1 - (z(t) - z_l(t)) + \frac{1}{2}(z(t) - z_l(t))^2\right] \leqslant \sigma(t) \\ \sigma(t) \leqslant T_{\max}\mathrm{e}^{-z_u(t)}[1 - (z(t) - z_u(t))] \end{cases} \tag{22}$$

where, $z_l(t)$ and $z_u(t)$ are computed as:

$$\begin{cases} z_l(t) = \ln(m_{\mathrm{wet}} - \alpha T_{\min} t) \\ z_u(t) = \ln(m_{\mathrm{wet}} - \alpha T_{\max} t) \end{cases} \tag{23}$$

In general, the relaxed Problem 2 can be reformulated by variable substitution and Taylor expansion as shown in Problem 3.

Problem 3. Convexified fuel optimal powered descent problem

$$\min_{\boldsymbol{u}(t),\sigma(t)} -z(t_{\mathrm{f}}) \Longleftrightarrow \min_{\boldsymbol{u}(t),\sigma(t)} \int_0^{t_{\mathrm{f}}} \sigma(t)\mathrm{d}t$$

$$\text{subject to}: \begin{cases} \dot{\boldsymbol{r}}(t) = \boldsymbol{v}(t) \\ \dot{\boldsymbol{v}}(t) = \boldsymbol{g}_{\mathrm{m}} + \boldsymbol{u}(t) \\ \dot{z}(t) = -\alpha\sigma(t) \\ \| \boldsymbol{u}(t) \|_2 \leqslant \sigma(t) \\ T_{\min}\mathrm{e}^{-z_l(t)}\left[1 - (z(t) - z_l(t)) + \frac{1}{2}(z(t) - z_l(t))^2\right] \leqslant \sigma(t) \\ \sigma(t) \leqslant T_{\max}\mathrm{e}^{-z_u(t)}[1 - (z(t) - z_u(t))] \\ \tan\gamma_{\mathrm{gs}} \leqslant \frac{r_x}{\sqrt{r_y^2 + r_z^2}} \\ \boldsymbol{r}(t_0) = \boldsymbol{r}_0,\ \boldsymbol{v}(t_0) = \boldsymbol{v}_0,\ z(t_0) = \ln m_{\mathrm{wet}} \\ \boldsymbol{r}(t_{\mathrm{f}}) = \boldsymbol{r}_{\mathrm{f}},\ \boldsymbol{v}(t_{\mathrm{f}}) = \boldsymbol{v}_{\mathrm{f}},\ z(t_{\mathrm{f}}) \geqslant \ln m_{\mathrm{dry}} \end{cases} \tag{24}$$

As for the glide-slope constraint (4), it can be transformed into the cone constraint as:

$$\left\| \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix} \right\|_2 \leqslant \begin{bmatrix} \frac{1}{\tan\gamma_{\mathrm{gs}}}, & 0, & 0 \end{bmatrix} \begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix} \tag{25}$$

Problem 3 is a convex optimization problem, specifically a Second-Order Cone Programming (SOCP) problem. This problem can be solved quickly using convex optimization solvers. However, these solvers cannot directly solve continuous problems, making it necessary to discretize the continuous problem.

Considering Problem 3 as an optimal control problem and choosing the new state variable as vector $\boldsymbol{X}(t)$. The system is then represented as follows:

$$\dot{\boldsymbol{X}}(t) = \boldsymbol{A}_{\mathrm{c}}\boldsymbol{X}(t) + \boldsymbol{B}_{\mathrm{c}}\begin{bmatrix} \boldsymbol{u}(t) \\ \sigma(t) \end{bmatrix} + \boldsymbol{B}_{\mathrm{c}}\mathscr{G}_{\mathrm{m}} \tag{26}$$

where,

$$\begin{cases} \boldsymbol{X}(t) = \begin{bmatrix} \boldsymbol{r}(t) \\ \boldsymbol{v}(t) \\ z(t) \end{bmatrix},\ \mathscr{G}_{\mathrm{m}} = \begin{bmatrix} \boldsymbol{g}_{\mathrm{m}} \\ 0 \end{bmatrix} \\ \boldsymbol{A}_{\mathrm{c}} = \begin{bmatrix} \boldsymbol{0}_{3\times3} & \boldsymbol{I}_3 & \boldsymbol{0}_{3\times1} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times1} \\ \boldsymbol{0}_{1\times3} & \boldsymbol{0}_{1\times3} & 0_{1\times1} \end{bmatrix},\ \boldsymbol{B}_{\mathrm{c}} = \begin{bmatrix} \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times1} \\ \boldsymbol{I}_3 & \boldsymbol{0}_{3\times1} \\ \boldsymbol{0}_{1\times3} & -\alpha \end{bmatrix} \end{cases} \tag{27}$$

The sampling time Δt is chosen to be:

$$\Delta t = \frac{t_{\mathrm{f}} - t_0}{N+1} \tag{28}$$

When the initial time satisfies $t_0 = 0$, the time at each discrete point is:

$$t_i = i\Delta t, \qquad i = 0, 1, \cdots, N, (N+1) \tag{29}$$

By discretization, the objective function in Eq. (24) is converted to the following form:

$$\min_{\boldsymbol{u}(t_k),\sigma(t_k)} -z(t_{N+1}) \Longleftrightarrow \min_{\boldsymbol{u}(t_k),\sigma(t_k)} \alpha\Delta t\sum_{k=0}^{N}\sigma(t_k) \tag{30}$$

Define $\boldsymbol{X}(t_k)$ as the discrete state variable at time t_k, and $\boldsymbol{u}(t_k)$ and $\sigma(t_k)$ as the discrete control variables. Note that for the state variable $\boldsymbol{X}(t_k)$, the range of k is $k = 0, 1, \cdots, N+1$, while for the control variables $\boldsymbol{u}(t_k)$ and $\sigma(t_k)$, the range of k is $k = 0, 1, \cdots, N$, and the discretized system equations are denoted as:

$$\boldsymbol{X}(t_{k+1}) = \boldsymbol{A}_{\mathrm{d}}\boldsymbol{X}(t_k) + \boldsymbol{B}_{\mathrm{d}}\begin{bmatrix} \boldsymbol{u}(t_k) \\ \sigma(t_k) \end{bmatrix} + \boldsymbol{B}_{\mathrm{d}}\mathscr{G}_{\mathrm{m}} \tag{31}$$

where, $\boldsymbol{X}(t_0)$ is the initial state and $\boldsymbol{X}(t_{N+1})$ is the terminal state. When using a zero-order hold for discretization, the matrices $\boldsymbol{A}_{\mathrm{d}}$ and $\boldsymbol{B}_{\mathrm{d}}$ are calculated as follows:

$$\boldsymbol{A}_{\mathrm{d}} = \mathrm{e}^{\boldsymbol{A}_{\mathrm{c}}\Delta t}, \boldsymbol{B}_d = \int_0^{\Delta t}\mathrm{e}^{\boldsymbol{A}_{\mathrm{c}}(\Delta t - \tau)}\boldsymbol{B}_{\mathrm{c}}\mathrm{d}\tau \tag{32}$$

According to Eqs. (27) and (32), the discrete matrix corresponding to this paper is obtained as:

$$\begin{cases} \boldsymbol{A}_{\mathrm{d}} = \begin{bmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \\ \boldsymbol{B}_{\mathrm{d}} = \begin{bmatrix} \frac{\Delta t^2}{2} & 0 & 0 & 0 \\ 0 & \frac{\Delta t^2}{2} & 0 & 0 \\ 0 & 0 & \frac{\Delta t^2}{2} & 0 \\ \Delta t & 0 & 0 & 0 \\ 0 & \Delta t & 0 & 0 \\ 0 & 0 & \Delta t & 0 \\ 0 & 0 & 0 & -\alpha\Delta t \end{bmatrix} \end{cases} \tag{33}$$

By iteratively calculating Eq. (31), the expression for $\boldsymbol{X}(t_{k+1})$ in terms of $\boldsymbol{X}(t_0)$ can be obtained as follows:

$$\boldsymbol{X}(t_{k+1}) = \boldsymbol{A}_{\mathrm{d}}^{k+1}\boldsymbol{X}(t_0) + \left(\boldsymbol{A}_{\mathrm{d}}^{k}\boldsymbol{B}_{\mathrm{d}} + \cdots + \boldsymbol{A}_{\mathrm{d}}\boldsymbol{B}_{\mathrm{d}} + \boldsymbol{B}_{\mathrm{d}}\right)\begin{bmatrix} \boldsymbol{g}_{\mathrm{m}} \\ 0 \end{bmatrix} + \left[\boldsymbol{A}_{\mathrm{d}}^{k}\boldsymbol{B}_{\mathrm{d}}, \quad \cdots, \quad \boldsymbol{A}_{\mathrm{d}}\boldsymbol{B}_{\mathrm{d}}, \quad \boldsymbol{B}_{\mathrm{d}}\right]\begin{bmatrix} \begin{bmatrix} \boldsymbol{u}_0 \\ \sigma_0 \end{bmatrix} \\ \vdots \\ \begin{bmatrix} \boldsymbol{u}_{k-1} \\ \sigma_{k-1} \end{bmatrix} \\ \begin{bmatrix} \boldsymbol{u}_k \\ \sigma_k \end{bmatrix} \end{bmatrix}, \quad k = 0, 1, \cdots, N \tag{34}$$

The discrete form of the thrust constraints are:

$$\| \boldsymbol{u}(t_k) \|_2 \leqslant \sigma(t_k) \tag{35}$$

$$\begin{cases} T_{\min}\mathrm{e}^{-z_l(t_k)}\left[1 - (z(t_k) - z_l(t_k)) + \frac{1}{2}(z(t_k) - z_l(t_k))^2\right] \leqslant \sigma(t_k) \\ \sigma(t_k) \leqslant T_{\max}\mathrm{e}^{-z_u(t_k)}[1 - (z(t_k) - z_u(t_k))] \\ z_l(t_k) = \ln(m_{\mathrm{wet}} - \alpha T_{\min}t_k) \\ z_u(t_k) = \ln(m_{\mathrm{wet}} - \alpha T_{\max}t_k) \end{cases} \tag{36}$$

The glide-slope constraint takes the discrete form

$$\begin{cases} \left\| \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{x,k} \\ r_{y,k} \\ r_{z,k} \end{bmatrix} \right\|_2 \leqslant \begin{bmatrix} \frac{1}{\tan \gamma_{gs}}, & 0, & 0 \end{bmatrix} \begin{bmatrix} r_{x,k} \\ r_{y,k} \\ r_{z,k} \end{bmatrix} \\ r_{x,k} = [1, \quad \mathbf{0}_{1\times 6}]\boldsymbol{X}(t_k),\ r_{y,k} = [0, \quad 1, \quad \mathbf{0}_{1\times 5}]\boldsymbol{X}(t_k) \\ r_{z,k} = [0, \quad 0, \quad 1, \quad \mathbf{0}_{1\times 4}]\boldsymbol{X}(t_k) \end{cases} \tag{37}$$

The initial and terminal state constraints in Eq. (3) can be converted into a discretized form as follows:

$$\begin{cases} \boldsymbol{X}(t_0) = \begin{bmatrix} \boldsymbol{r}_0 \\ \boldsymbol{v}_0 \\ \ln m_{\text{wet}} \end{bmatrix} \\ \begin{bmatrix} \boldsymbol{I}_3 & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 1} \\ \mathbf{0}_{3\times 3} & \boldsymbol{I}_3 & \mathbf{0}_{3\times 1} \end{bmatrix} \boldsymbol{X}(t_{\text{f}}) = \begin{bmatrix} \boldsymbol{r}_{\text{f}} \\ \boldsymbol{v}_{\text{f}} \end{bmatrix} \\ [\mathbf{0}_{3\times 1} \quad \mathbf{0}_{3\times 1} \quad 1]\boldsymbol{X}(t_{\text{f}}) \geqslant \ln m_{\text{dry}} \end{cases} \tag{38}$$

Thus, the discretized formulation of the convexified fuel optimal powered descent Problem 3 is as shown in Problem 4.

Problem 4. Discretized form of the convexified fuel optimal powered descent problem

$$\begin{aligned} &\min_{\boldsymbol{u}(t_k),\sigma(t_k)} \ -z(t_{N+1}) \Longleftrightarrow \min_{\boldsymbol{u}(t_k),\sigma(t_k)} \ \alpha \Delta t \sum_{k=0}^{N} \sigma(t_k) \\ &\text{subject to}: \ \text{Eqs. } (33)-(38) \end{aligned} \tag{39}$$

where, $k = 0, 1, \cdots, N$. The intermediate variables are calculated as follows:

$$t_k = \Delta t \cdot k = \frac{t_{\text{f}} - t_0}{N+1} \cdot k \tag{40}$$

Problem 4 presents a discretized model that can be directly solved using convex optimization solvers, which can be conveniently modeled and solved using CVX or YALMIP.

However, this model is only a convex approximation of the original nonconvex Problem 2 achieved through a Taylor expansion. The approximation error in this model increases over time t. Therefore, it is necessary to refine the problem to obtain a more accurate solution to the original problem. An iterative refinement algorithm is proposed based on the solution of Problem 4.

Define $\boldsymbol{u}(\cdot)$ as the set $\{\boldsymbol{u}(t_k)|k = 0, 1, \cdots, N\}$, define $\sigma(\cdot)$ as the set $\{\sigma(t_k)|k = 0, 1, \cdots, N\}$, and define $\boldsymbol{X}(\cdot)$ as the set $\{\boldsymbol{X}(t_k)|k = 0, 1, \cdots, N+1\}$. Denote the solution of Problem 4 as $\widetilde{\boldsymbol{u}}(\cdot), \widetilde{\sigma}(\cdot), \widetilde{\boldsymbol{X}}(\cdot)$. Then, $\widetilde{z}(t_k)$ can be expressed as:

$$\widetilde{z}(t_k) = [\mathbf{0}_{1\times 6}, \quad 1]\,\widetilde{\boldsymbol{X}}(t_k) \tag{41}$$

Thus, based on Eq. (41), the original approximation in Eq. (22) is converted into a modified Taylor expansion in terms of $\widetilde{z}(t_k)$ as follows:

$$\begin{cases} \sigma(t_k) \geqslant T_{\min} \mathrm{e}^{-\widetilde{z}(t_k)} \left[1 - (z(t_k) - \widetilde{z}(t_k)) + \frac{1}{2}(z(t_k) - \widetilde{z}(t_k))^2\right] \\ \sigma(t_k) \leqslant T_{\max} \mathrm{e}^{-\widetilde{z}(t_k)} [1 - (z(t_k) - \widetilde{z}(t_k))] \end{cases} \tag{42}$$

Problem 5. Discretized form of the sequential convexified problem

$$\begin{aligned} &\min_{\boldsymbol{u}(t_k),\sigma(t_k)} \ -z(t_{N+1}) \Longleftrightarrow \min_{\boldsymbol{u}(t_k),\sigma(t_k)} \ \alpha \Delta t \sum_{k=0}^{N} \sigma(t_k) \\ &\text{subject to}: \ \text{Eqs. } (33)-(35), (37), (38), (41), (42) \end{aligned} \tag{43}$$

Based on the discussion above, the steps of the improved lossless convexification algorithm is summarized in Algorithm 1 as follows:

Algorithm 1. Improved lossless convexification algorithm.

1. Define tolerance $\varepsilon_{\boldsymbol{X}} > 0$.
2. Solve Problem 4 to obtain $\widetilde{\boldsymbol{u}}(\cdot), \widetilde{\sigma}(\cdot), \widetilde{\boldsymbol{X}}(\cdot)$.
3. Assign $z_l(t_k) = \widetilde{z}(t_k), z_u(t_k) = \widetilde{z}(t_k)$ according to Eqs. (36) and (42).
4. Solve Problem 5 based on the modified Taylor expansion (42). Denote the solution as $\hat{\boldsymbol{u}}(\cdot), \hat{\sigma}(\cdot), \hat{\boldsymbol{X}}(\cdot)$.
5. **while** $\| \hat{\boldsymbol{X}}(\cdot) - \widetilde{\boldsymbol{X}}(\cdot) \|_2 \geqslant \varepsilon_{\boldsymbol{X}}$ **do**
6. Assign $\widetilde{\boldsymbol{X}}(\cdot) = \hat{\boldsymbol{X}}(\cdot)$.
7. Assign $z_l(t_k) = \widetilde{z}(t_k), z_u(t_k) = \widetilde{z}(t_k)$ according to Eqs. (36) and (42).
8. Solve Problem 5 based on the modified Taylor expansion in Step 7. Denote the solution as $\hat{\boldsymbol{u}}(\cdot), \hat{\sigma}(\cdot), \hat{\boldsymbol{X}}(\cdot)$.
9. Calculate the error $\| \hat{\boldsymbol{X}}(\cdot) - \widetilde{\boldsymbol{X}}(\cdot) \|_2$.
10. **end while**
11. Assign $\boldsymbol{u}^*(\cdot) = \hat{\boldsymbol{u}}(\cdot), \sigma^*(\cdot) = \hat{\sigma}(\cdot), \boldsymbol{X}^*(\cdot) = \hat{\boldsymbol{X}}(\cdot)$.
12. **return** $\boldsymbol{u}^*(\cdot), \sigma^*(\cdot), \boldsymbol{X}^*(\cdot)$ as the optimal solution.

Algorithm 1 is primarily designed to reduce the error caused by Taylor expansion in Eq. (22). Although this error has a minor impact on the optimized trajectory and is generally ignored in existing literature.[34,35] However, based on the method of sequential convex optimization,[36,37] Algorithm 1 can significantly minimize this error. The convergence proof of Algorithm 1 is provided in Appendix A.

3.2. Problem formulation considering uncertainties in initial mass and fuel consumption

In the previous subsection, the improved lossless convexification algorithm for the fuel optimal powered descent problem was constructed. However, this construction assumes that the initial mass and fuel consumption are fixed and certain. In practice, the initial mass may be uncertain, particularly in the context of reusable rockets. Additionally, fuel consumption is not strictly linear with thrust amplitude and may also exhibit uncertainties. Therefore, this subsection builds upon the previous one to construct a problem formulation considering uncertainties in initial mass and fuel consumption.

3.2.1. Problem formulation considering uncertainties

Based on Problem 4 and Algorithm 1, we obtain the optimal trajectory $\boldsymbol{u}^*(\cdot), \sigma^*(\cdot), \boldsymbol{X}^*(\cdot)$ for the nominal mass $\bar{m}$ without considering uncertainties, and since the variable replacement, $\boldsymbol{u}^*(t) = \frac{\boldsymbol{T}(t)}{m(t)}$. Therefore, what we get by Algorithm 1 is the set of thrust magnitudes $\boldsymbol{T}(\cdot)$ based on the nominal masses.

First, since the optimized thrust obtained based on the nominal mass is fixed. When the mass has uncertainties as shown in Eq. (6), we find that discrete-time state Eqs. (31) and (34), which employ variable substitution, are not applicable for analyzing the impact on states, trajectory, and constraints satisfaction. Therefore, we return to the original, state equations without variable substitution as in Eqs. (1) and (2). Assuming that mass and

thrust are discretized using a zero-order hold, we obtain the discrete forms of Eqs. (1) and (2):

$$\begin{cases} \begin{bmatrix} \boldsymbol{r}_{k+1} \\ \boldsymbol{v}_{k+1} \end{bmatrix} = \widehat{\boldsymbol{A}}_{\mathrm{d}} \begin{bmatrix} \boldsymbol{r}_k \\ \boldsymbol{v}_k \end{bmatrix} + \widehat{\boldsymbol{B}}_{\mathrm{d}} \left(\frac{\boldsymbol{T}_k}{m_k} + \boldsymbol{g}_{\mathrm{m}} \right) \\ m_{k+1} = m_k - \alpha \| \boldsymbol{T}_k(t) \|_2 \end{cases} \tag{44}$$

where, matrices for discrete equations $\hat{\boldsymbol{A}}_{\mathrm{d}}$ and $\hat{\boldsymbol{B}}_{\mathrm{d}}$ are calculated as follows:

$$\begin{cases} \widehat{\boldsymbol{A}}_{\mathrm{c}} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \widehat{\boldsymbol{B}}_{\mathrm{c}} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ \widehat{\boldsymbol{A}}_{\mathrm{d}} = \mathrm{e}^{\widehat{\boldsymbol{A}}_{\mathrm{c}} \Delta t} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \\ \widehat{\boldsymbol{B}}_{\mathrm{d}} = \int_0^{\Delta t} \mathrm{e}^{\widehat{\boldsymbol{A}}_{\mathrm{c}}(\Delta t - \tau)} \widehat{\boldsymbol{B}}_{\mathrm{c}} \mathrm{d}\tau = \begin{bmatrix} \frac{\Delta t^2}{2} \\ \Delta t \end{bmatrix} \end{cases} \tag{45}$$

Thus, states containing uncertainty can be represented in the following form:

$$\begin{aligned} \begin{bmatrix} \boldsymbol{r}_{k+1} \\ \boldsymbol{v}_{k+1} \end{bmatrix} &= \widehat{\boldsymbol{A}}_{\mathrm{d}} \begin{bmatrix} \boldsymbol{r}_k \\ \boldsymbol{v}_k \end{bmatrix} + \widehat{\boldsymbol{B}}_{\mathrm{d}} \left(\frac{\boldsymbol{T}_k}{m_k} + \boldsymbol{g}_{\mathrm{m}} \right) \\ &= \widehat{\boldsymbol{A}}_{\mathrm{d}} \begin{bmatrix} \boldsymbol{r}_k \\ \boldsymbol{v}_k \end{bmatrix} + \widehat{\boldsymbol{B}}_{\mathrm{d}} \left(\frac{\boldsymbol{T}_k}{m_k + \Delta m} + \boldsymbol{g}_{\mathrm{m}} \right) \\ &= \widehat{\boldsymbol{A}}_{\mathrm{d}} \begin{bmatrix} \boldsymbol{r}_k \\ \boldsymbol{v}_k \end{bmatrix} + \widehat{\boldsymbol{B}}_{\mathrm{d}} \boldsymbol{g}_{\mathrm{m}} + \widehat{\boldsymbol{B}}_{\mathrm{d}} \frac{\boldsymbol{T}_k}{m_k} \cdot \varrho_k \end{aligned} \tag{46}$$

where, $k = 0, 1, \cdots, N$. $\bar{m}_k$ is the nominal mass. $\varrho_k = \frac{\bar{m}_k}{\bar{m}_k + \Delta m_k}$ is the coefficient.

Iterating Eq. (46) yields the following:

$$\begin{aligned} \begin{bmatrix} \boldsymbol{r}_{k+1} \\ \boldsymbol{v}_{k+1} \end{bmatrix} &= \widehat{\boldsymbol{A}}_{\mathrm{d}} \left[\widehat{\boldsymbol{A}}_{\mathrm{d}} \begin{bmatrix} \boldsymbol{r}_{k-1} \\ \boldsymbol{v}_{k-1} \end{bmatrix} + \widehat{\boldsymbol{B}}_{\mathrm{d}} \boldsymbol{g}_{\mathrm{m}} + \widehat{\boldsymbol{B}}_{\mathrm{d}} \frac{\boldsymbol{T}_{k-1}}{m_{k-1}} \cdot \varrho_{k-1} \right] \\ &\quad + \widehat{\boldsymbol{B}}_{\mathrm{d}} \boldsymbol{g}_{\mathrm{m}} + \widehat{\boldsymbol{B}}_{\mathrm{d}} \frac{\boldsymbol{T}_k}{m_k} \cdot \varrho_k \\ &= \widehat{\boldsymbol{A}}_{\mathrm{d}}^2 \begin{bmatrix} \boldsymbol{r}_{k-1} \\ \boldsymbol{v}_{k-1} \end{bmatrix} + \widehat{\boldsymbol{A}}_{\mathrm{d}} \widehat{\boldsymbol{B}}_{\mathrm{d}} \boldsymbol{g}_{\mathrm{m}} + \widehat{\boldsymbol{B}}_{\mathrm{d}} \boldsymbol{g}_{\mathrm{m}} \\ &\quad + \widehat{\boldsymbol{A}}_{\mathrm{d}} \widehat{\boldsymbol{B}}_{\mathrm{d}} \frac{\boldsymbol{T}_{k-1}}{m_{k-1}} \cdot \varrho_{k-1} + \widehat{\boldsymbol{B}}_{\mathrm{d}} \frac{\boldsymbol{T}_k}{m_k} \cdot \varrho_k \\ &= \widehat{\boldsymbol{A}}_{\mathrm{d}}^{k+1} \begin{bmatrix} \boldsymbol{r}_0 \\ \boldsymbol{v}_0 \end{bmatrix} + \left(\widehat{\boldsymbol{A}}_{\mathrm{d}}^k \widehat{\boldsymbol{B}}_{\mathrm{d}} + \cdots + \widehat{\boldsymbol{A}}_{\mathrm{d}} \widehat{\boldsymbol{B}}_{\mathrm{d}} + \widehat{\boldsymbol{B}}_{\mathrm{d}} \right) \boldsymbol{g}_{\mathrm{m}} \\ &\quad + \widehat{\boldsymbol{A}}_{\mathrm{d}}^k \widehat{\boldsymbol{B}}_{\mathrm{d}} \frac{\boldsymbol{T}_0}{m_0} \cdot \varrho_0 + \cdots + \widehat{\boldsymbol{A}}_{\mathrm{d}} \widehat{\boldsymbol{B}}_{\mathrm{d}} \frac{\boldsymbol{T}_{k-1}}{m_{k-1}} \cdot \varrho_{k-1} + \widehat{\boldsymbol{B}}_{\mathrm{d}} \frac{\boldsymbol{T}_k}{m_k} \cdot \varrho_k \end{aligned} \tag{47}$$

For terms containing uncertainty, they can be transformed into the following form:

$$\begin{aligned} &\widehat{\boldsymbol{A}}_{\mathrm{d}}^k \widehat{\boldsymbol{B}}_{\mathrm{d}} \frac{\boldsymbol{T}_0}{m_0} \cdot \varrho_0 + \cdots + \widehat{\boldsymbol{A}}_{\mathrm{d}} \widehat{\boldsymbol{B}}_{\mathrm{d}} \frac{\boldsymbol{T}_{k-1}}{m_{k-1}} \cdot \varrho_{k-1} + \widehat{\boldsymbol{B}}_{\mathrm{d}} \frac{\boldsymbol{T}_k}{m_k} \cdot \varrho_k \\ &= \left[\widehat{\boldsymbol{A}}_{\mathrm{d}}^k \widehat{\boldsymbol{B}}_{\mathrm{d}}, \quad \cdots, \quad \widehat{\boldsymbol{A}}_{\mathrm{d}} \widehat{\boldsymbol{B}}_{\mathrm{d}}, \quad \widehat{\boldsymbol{B}}_{\mathrm{d}} \right] \begin{bmatrix} \frac{\boldsymbol{T}_0}{m_0} \cdot (1 + \Delta \varrho_0) \\ \vdots \\ \frac{\boldsymbol{T}_{k-1}}{m_{k-1}} \cdot (1 + \Delta \varrho_{k-1}) \\ \frac{\boldsymbol{T}_k}{m_k} \cdot (1 + \Delta \varrho_k) \end{bmatrix} \\ &= \left[\widehat{\boldsymbol{A}}_{\mathrm{d}}^k \widehat{\boldsymbol{B}}_{\mathrm{d}}, \quad \cdots, \quad \widehat{\boldsymbol{A}}_{\mathrm{d}} \widehat{\boldsymbol{B}}_{\mathrm{d}}, \quad \widehat{\boldsymbol{B}}_{\mathrm{d}} \right] \begin{bmatrix} \frac{\boldsymbol{T}_0}{m_0} \\ \vdots \\ \frac{\boldsymbol{T}_{k-1}}{m_{k-1}} \\ \frac{\boldsymbol{T}_k}{m_k} \end{bmatrix} \\ &\quad + \left[\widehat{\boldsymbol{A}}_{\mathrm{d}}^k \widehat{\boldsymbol{B}}_{\mathrm{d}}, \quad \cdots, \quad \widehat{\boldsymbol{A}}_{\mathrm{d}} \widehat{\boldsymbol{B}}_{\mathrm{d}}, \quad \widehat{\boldsymbol{B}}_{\mathrm{d}} \right] \begin{bmatrix} \frac{\boldsymbol{T}_0}{m_0} \cdot \Delta \varrho_0 \\ \vdots \\ \frac{\boldsymbol{T}_{k-1}}{m_{k-1}} \cdot \Delta \varrho_{k-1} \\ \frac{\boldsymbol{T}_k}{m_k} \cdot \Delta \varrho_k \end{bmatrix} \end{aligned} \tag{48}$$

where, $\Delta \varrho_k = \varrho_k - 1 = \frac{-\Delta m_k}{m_k + \Delta m_k}$. Eq. (48) shows that the impact of mass uncertainty can be divided into two parts: one part is the nominal term, and the other part is the uncertainty term. The uncertainty term in Eq. (48) can be simplified as:

$$\begin{aligned} \begin{bmatrix} \Delta \boldsymbol{r}_{k+1} \\ \Delta \boldsymbol{v}_{k+1} \end{bmatrix} &= \left[\widehat{\boldsymbol{A}}_{\mathrm{d}}^k \widehat{\boldsymbol{B}}_{\mathrm{d}}, \quad \cdots, \quad \widehat{\boldsymbol{A}}_{\mathrm{d}} \widehat{\boldsymbol{B}}_{\mathrm{d}}, \quad \widehat{\boldsymbol{B}}_{\mathrm{d}} \right] \begin{bmatrix} \frac{\boldsymbol{T}_0}{m_0} \Delta \varrho_0 \\ \vdots \\ \frac{\boldsymbol{T}_{k-1}}{m_{k-1}} \Delta \varrho_{k-1} \\ \frac{\boldsymbol{T}_k}{m_k} \Delta \varrho_k \end{bmatrix} \\ &= \left[\widehat{\boldsymbol{A}}_{\mathrm{d}}^k \widehat{\boldsymbol{B}}_{\mathrm{d}}, \quad \cdots, \quad \widehat{\boldsymbol{A}}_{\mathrm{d}} \widehat{\boldsymbol{B}}_{\mathrm{d}}, \quad \widehat{\boldsymbol{B}}_{\mathrm{d}} \right] \begin{bmatrix} \Delta \boldsymbol{a}_0 \\ \vdots \\ \Delta \boldsymbol{a}_{k-1} \\ \Delta \boldsymbol{a}_k \end{bmatrix} \end{aligned} \tag{49}$$

From Eq. (45), we obtain:

$$\hat{\boldsymbol{A}}_{\mathrm{d}}^k = \begin{bmatrix} 1 & \Delta t \cdot k \\ 0 & 1 \end{bmatrix}, \hat{\boldsymbol{A}}_{\mathrm{d}}^k \hat{\boldsymbol{B}}_{\mathrm{d}} = \begin{bmatrix} \frac{\Delta t^2}{2} \\ \Delta t \end{bmatrix} + \begin{bmatrix} \Delta t^2 \cdot k \\ 0 \end{bmatrix} \tag{50}$$

By substituting Eq. (50), the uncertainty term in Eq. (49) can be expressed as:

$$\begin{aligned} \begin{bmatrix} \Delta \boldsymbol{r}_{k+1} \\ \Delta \boldsymbol{v}_{k+1} \end{bmatrix} &= \begin{bmatrix} \frac{\Delta t^2}{2} \\ \Delta t \end{bmatrix} (\Delta \boldsymbol{a}_0 + \cdots + \Delta \boldsymbol{a}_{k-1} + \Delta \boldsymbol{a}_k) \\ &+ \begin{bmatrix} \Delta t^2 \cdot k \\ 0 \end{bmatrix} \cdot \Delta \boldsymbol{a}_0 + \cdots + \begin{bmatrix} \Delta t^2 \cdot 0 \\ 0 \end{bmatrix} \Delta \boldsymbol{a}_k \\ &= \begin{bmatrix} \frac{\Delta t^2}{2} \\ \Delta t \end{bmatrix} \left(\sum_{i=0}^{k} \Delta \boldsymbol{a}_i \right) + \sum_{i=0}^{k-1} \begin{bmatrix} \Delta t^2 \cdot (k-i) \\ 0 \end{bmatrix} \Delta \boldsymbol{a}_i \end{aligned} \tag{51}$$

where, $\Delta \boldsymbol{a}_k$ is calculated as follows:

$$\begin{aligned} \Delta \boldsymbol{a}_k &= \frac{\boldsymbol{T}_k}{m_k} \cdot \Delta \varrho_k \\ &= \boldsymbol{u}^*(t_k) \cdot \frac{-\Delta m_k}{\bar{m}_k + \Delta m_k} \end{aligned} \tag{52}$$

In Eq. (52), $\Delta \boldsymbol{a}_k$ can be seen as the equivalent acceleration resulting from mass uncertainty. This equivalent acceleration depends on the optimized thrust magnitude $\boldsymbol{T}_k$, nominal mass $\bar{m}_k$, and mass uncertainty Δm_k.

By substituting the uncertain mass Δm_k in Eqs. (8), (11), and (12) and the nominal mass $\bar{m}_k$ together with the optimized thrust $\boldsymbol{T}_k$ in Algorithm 1, the equivalent acceleration error $\Delta \boldsymbol{a}_k$ caused by uncertainty can be determined using Eq. (52). Then, the states error due to mass uncertainty can be calculated according to Eq. (51).

4. Robust powered descent guidance considering worst-case scenarios

Due to uncertainties in mass, there may be deviations between the actual position and the theoretical position, which can lead to violations of glide-slope constraints during descent and result in failed obstacle avoidance. In this paper, we define the mass uncertainty as a bounded closed set, aiming to ensure a safe landing even in the worst-case scenario, while allowing the lander to approach the landing point with optimal fuel efficiency.

4.1. Effect of mass uncertainty on position

By solving the improved lossless convexification Algorithm 1, we obtained the trajectory optimization results for the nominal case. The acceleration $\boldsymbol{u}^*(t)$ in the results is part of the calculation for $\Delta \boldsymbol{a}_k$ in Eq. (52).

According to Eq. (52), we draw the following conclusions:

(1) When $\Delta m_k < 0$, meaning the actual mass is smaller than the nominal mass, the magnitude of acceleration increases under constant thrust, leading to faster acceleration and deceleration.
(2) When $\Delta m_k > 0$, meaning the actual mass is greater than the nominal mass, the magnitude of acceleration decreases under constant thrust, resulting in slower acceleration and deceleration.

Based on Newton's laws of kinematics, the position error is the double integral of the acceleration error and can be expressed as:

$$\Delta \boldsymbol{r}(t) = \Delta \boldsymbol{a}(t)\mathrm{d}t\mathrm{d}t \tag{53}$$

where, $\Delta \boldsymbol{r}(t)$ represents the position error, and $\Delta \boldsymbol{a}(t)$ represents the velocity error.

On the basis of Eqs. (51) and (52), the position error $\Delta \boldsymbol{r}_k$ and velocity error $\Delta \boldsymbol{v}_k$ at any given time t_k caused by mass uncertainty can be calculated. Based on the proposed improved lossless convexification Algorithm 1, different initial positions and velocities result in varying thrusts, which in turn lead to different accelerations. For the same initial position but with varying initial velocities, mass uncertainty may result in different disturbance trajectories, as shown in Fig. 3.

Different initial positions and velocities lead to varying thrusts and, consequently, different position errors due to mass uncertainty. However, the position error exhibit the following characteristics: First, the error is independent along the x, y, and z axes. Second, when mass uncertainty is defined as a bounded closed set, the resulting disturbance error also forms a bounded set. Lastly, if position uncertainty is a continuous, bounded set of real values as in Eqs. (7)–(12), the position error is geometrically represented as a rectangular shape.

4.2. Geometric analysis of worst-case scenarios

Based on the characteristic that the position error is bounded, we adopt the worst-case analysis to ensure that the powered descent phase is safe when approaching the landing site. First, a description of the worst-case scenario is provided, followed by an analysis of its geometric characteristics and constraint conditions. Finally, the problem formulation and the iterative solution algorithm are presented.

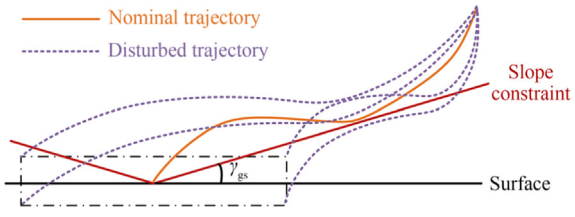


Fig. 3 Illustration of disturbed trajectory.

In the worst-case scenario, it is required that the position always remains within glide-slope constraints while aiming for the final position to be as close to the landing point as possible. Therefore, the worst-case condition occurs when the disturbed trajectory is exactly tangent to the glide-slope constraint. The worst-case scenario is illustrated in Fig. 4.

Based on the worst-case analysis, this paper proposes a modified glide-slope constraint to ensure that the perturbed trajectory can still satisfy the original glide-slope constraint, as shown in Fig. 5. One thing that needs to be made clear is that the position error is time-varying, so the modified glide-slope constraint needs to be determined by an iterative algorithm. In each iteration a new modified glide-slope constraint is computed based on the previous result. Fig. 5 shows the relationship between the original glide-slope constraints, the old modified glide-slope constraint and the new modified glide-slope constraint. The new modified glide-slope constraint is determined by the modified terminal position $\hat{\boldsymbol{r}}_\mathrm{f}$ and the A_1 point. The calculation of the modified glide-slope constraint will be given below.

To simplify the calculation of the modified glide-slope constraint, the three-dimensional space $Oxyz$ is projected onto a

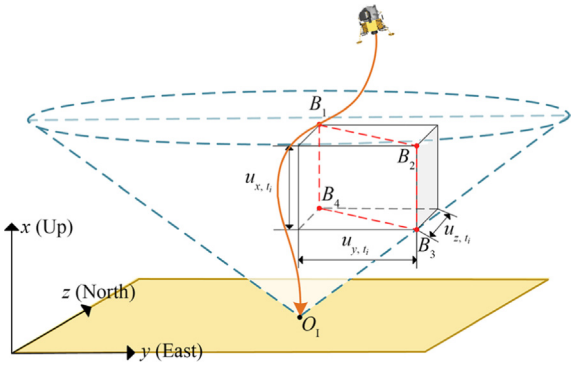


Fig. 4 Three-dimensional schematic of worst-case scenario.

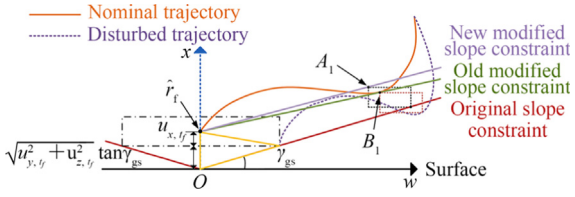


Fig. 5 Illustration of modified glide-slope constraint.

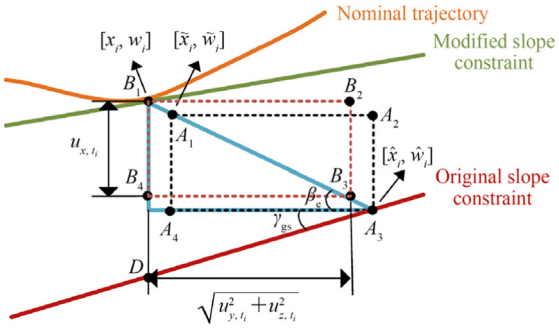


Fig. 6 The first scenario for region around points A_1 and B_1.

two-dimensional plane Oxw where the rectangle $B_1B_2B_3B_4$ is located. The Oxw Cartesian coordinate system is defined using the origin O, the Ox-axis, and the position coordinates $[x_i, y_i, z_i]$ at time t_i. The positive direction of the original y-axis is adopted as the positive direction of the w-axis, while the x-axis remains aligned with the x-axis of the original $Oxyz$ coordinate system.

Since the glide-slope constraint forms a conical region with its central axis aligned vertically, the adjusted terminal position lies directly above the landing site. As shown in Fig. 5, the modified terminal position $\hat{\boldsymbol{r}}_{\mathrm{f}}$ is computed as follows:

$$\hat{\boldsymbol{r}}_{\mathrm{f}} = \begin{bmatrix} \mathscr{U}_{x,t_{\mathrm{f}}} + \sqrt{\mathscr{U}_{y,t_{\mathrm{f}}}^2 + \mathscr{U}_{z,t_{\mathrm{f}}}^2}\tan\gamma_{\mathrm{gs}} \\ 0 \\ 0 \end{bmatrix} \tag{54}$$

To better illustrate the relationship between uncertainty and the glide-slope constraint in Fig. 4, the three-dimensional spatial relationships are projected onto a two-dimensional plane at each point on the trajectory. A zoomed-in view of the rectangle $B_1B_2B_3B_4$ is provided in Figs. 6 and 7, highlighting two possible scenarios.

In the first scenario, point B_1 is far from the original glide-slope constraint, as illustrated in Fig. 6. For instance, at the initial time, the lander's position is distant from the original glide-slope constraint. Even with the presence of uncertainty, represented by the dashed box, the original glide-slope constraint is still satisfied.

In the second scenario, point B_1 is close to the original glide-slope constraint, as shown in Fig. 7. For example, at time t_i, the lander's position is near the original glide-slope constraint. When uncertainty, depicted by the dashed box, is introduced, the disturbed position no longer satisfies the original glide-slope constraint.

Although there are two scenarios, we can use a uniform method to calculate the coordinates of point A_1. The position of point B_1 corresponds to the nominal trajectory at time t_i, which is known. In the $Oxyz$ coordinate system, it is represented as $[x_i, y_i, z_i]$, and in the Oxw coordinate system, it is expressed as $[x_i, w_i]$. The relationship between the two coordinate systems is given as follows:

$$\begin{cases} w_i = \mathrm{sign}(y_i)\sqrt{y_i^2 + z_i^2} \\ \mathrm{sign}(y_i) = \begin{cases} 1, & y_i \geqslant 0 \\ -1, & y_i < 0 \end{cases} \end{cases} \tag{55}$$

The length of segment B_1D represents the vertical distance from point B_1 to the original glide-slope constraint at time t_i. It is calculated as follows:

$$\left|\overline{B_1D}\right| = x_i - \sqrt{y_i^2 + z_i^2}\tan\gamma_{\mathrm{gs}} \tag{56}$$

The angle β_{e} in Figs. 6 and 7 is calculated based on the magnitude of uncertainty in the vertical and horizontal directions.

$$\beta_{\mathrm{e}}(i) = \tan^{-1}\left(\left|\frac{\mathscr{U}_{x,t_i}}{\sqrt{\mathscr{U}_{y,t_i}^2 + \mathscr{U}_{z,t_i}^2}}\right|\right) \tag{57}$$

where, the uncertainty terms correspond to the perturbation terms for the position as $\mathscr{U}_{x,t_i} = \Delta r_{x,t_i}$, $\mathscr{U}_{y,t_i} = \Delta r_{y,t_i}, \mathscr{U}_{z,t_i} = \Delta r_{z,t_i}$.

Using the law of sines for triangle $\triangle B_1A_3D$, we have the following relationship:

$$\frac{\left|\overline{B_1D}\right|}{\sin\left(\beta_{\mathrm{e}} + \gamma_{\mathrm{gs}}\right)} = \frac{\left|\overline{B_1A_3}\right|}{\sin\left(\frac{\pi}{2} - \gamma_{\mathrm{gs}}\right)} \tag{58}$$

The diagonal length of the uncertainty rectangle is given by

$$\left|\overline{A_1A_3}\right| = \left|\overline{B_1B_3}\right| = \sqrt{\mathscr{U}_{x,t_i}^2 + \mathscr{U}_{y,t_i}^2 + \mathscr{U}_{z,t_i}^2} \tag{59}$$

In order to calculate $\overrightarrow{A_1B_1}$ in a uniform way, we define the direction from A_1 to A_3 as positive. In Fig. 6, $\overrightarrow{A_1B_1}$ is less than zero. Conversely, as shown in Fig. 7, $\overrightarrow{A_1B_1}$ is greater than zero. The value of $\overrightarrow{A_1B_1}$ is calculated as follows:

$$\begin{aligned} \overrightarrow{A_1B_1} &= \left|\overline{A_1A_3}\right| - \left|\overline{B_1A_3}\right| \\ &= \sqrt{\mathscr{U}_{x,t_i}^2 + \mathscr{U}_{y,t_i}^2 + \mathscr{U}_{z,t_i}^2} - \frac{\left|\overline{B_1D}\right|\sin\left(\frac{\pi}{2} - \gamma_{\mathrm{gs}}\right)}{\sin\left(\beta_{\mathrm{e}}(i) + \gamma_{\mathrm{gs}}\right)} \end{aligned} \tag{60}$$

At this point, $\overrightarrow{A_1B_1}$ has been calculated, and since the position of point B_1 is known, the coordinates $[\widetilde{x}_i, \widetilde{w}_i]$ of point A_1 can be determined as follows:

$$\begin{cases} \widetilde{x}_i = x_i + \overrightarrow{A_1B_1} \cdot \sin\left(\beta_{\mathrm{e}}(i)\right) \\ \widetilde{w}_i = w_i - \mathrm{sign}(y_i) \cdot \overrightarrow{A_1B_1} \cdot \cos\left(\beta_{\mathrm{e}}(i)\right) \end{cases} \tag{61}$$

With this, the new modified glide-slope constraint at time t_i in Fig. 5 can be computed as:

$$\zeta_{\mathrm{gs}}(i) = \tan^{-1}\left(\frac{\widetilde{x}_i - \mathscr{U}_{x,t_{\mathrm{f}}} - \sqrt{\mathscr{U}_{y,t_{\mathrm{f}}}^2 + \mathscr{U}_{z,t_{\mathrm{f}}}^2}\tan\gamma_{\mathrm{gs}}}{\widetilde{w}_i}\right) \tag{62}$$

We have obtained $\zeta_{\mathrm{gs}}(i)$ for different time moments t_i. However, instead of directly selecting the minimum value of $\zeta_{\mathrm{gs}}(i)$ as the new modified glide-slope constraint, we must choose the $\zeta_{\mathrm{gs}}(i)$ value at the moment corresponding to the position closest to the glide-slope constraint, while taking into account the uncertainties. The distance from B_1 to the glide-slope is denoted as $\xi_{\mathrm{gs}}(i)$, the revised distance considering uncertainties is denoted as $\Omega_{\mathrm{gs}}(i)$. $\xi_{\mathrm{gs}}(i)$ and $\Omega_{\mathrm{gs}}(i)$ are calculated as follows:

$$\begin{cases} \xi_{\mathrm{gs}}(i) = \left(x_i - \sqrt{y_i^2 + z_i^2}\tan\gamma_{\mathrm{gs}}\right) \cdot \cos\gamma_{\mathrm{gs}} \\ \Omega_{\mathrm{gs}}(i) = \xi_{\mathrm{gs}}(i) - \sqrt{\mathscr{U}_{x,t_i}^2 + \mathscr{U}_{y,t_i}^2 + \mathscr{U}_{z,t_i}^2} \cdot \sin\left(\gamma_{\mathrm{gs}} + \beta_{\mathrm{e}}(i)\right) \end{cases} \tag{63}$$

Let the time corresponding to this first local minimum of $\Omega_{\mathrm{gs}}(i)$ be denoted as t_p. The new modified glide-slope constraint is expressed as

$$\gamma_{\mathrm{new}} = \max\left(0, \zeta_{\mathrm{gs}}(p)\right) \tag{64}$$

where, in order to maintain the convex character of the glide-slope constraint, it is necessary to limit the glide-slope constraint angle to be greater than or equal to zero.

Using the new modified glide-slope constraint angle γ_{new} and the terminal position $\hat{\boldsymbol{r}}_{\mathrm{f}}$, the nominal trajectory under the modified glide-slope constraint can be determined with Algorithm 1. Based on this, we present the following iterative solving Algorithm 2 that accounts for uncertainties in mass and fuel consumption.

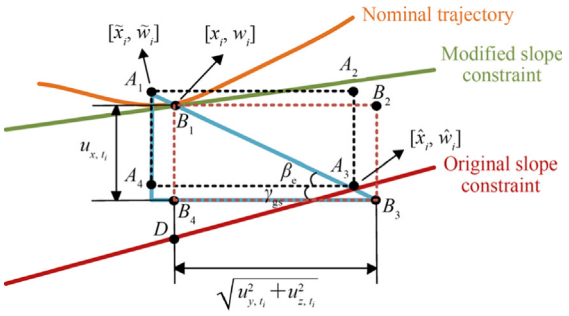


Fig. 7 The second scenario for region around points A_1 and B_1.

Algorithm 2. Robust powered descent guidance algorithm considering mass and fuel consumption uncertainties.

1. Define tolerance $\varepsilon > 0$, let $\gamma_{\text{old}} = \gamma_{\text{gs}}$.
2. Solving the nominal problem using Algorithm 1 to obtain $\boldsymbol{u}(\cdot), \sigma(\cdot), \boldsymbol{X}(\cdot)$.
3. Calculate the uncertainty of the mass Δm_k at each time t_k using Eqs. (7)–(11).
4. Calculate the effect of uncertainty on position Δr_k using Eqs. (51) and (52).
5. Calculate the new modified glide-slope constraint γ_{new} using Eqs. (54)–(64).
6. **while** $|\gamma_{\text{new}} - \gamma_{\text{old}}| \geqslant \varepsilon$ **do**
7. Replacing the modified terminal position $\hat{\boldsymbol{r}}_f$ and modified glide-slope constraint γ_{new} with the original parameters.
8. Let $\gamma_{\text{old}} = \gamma_{\text{new}}$, solving the nominal problem using Algorithm 1 to obtain $\boldsymbol{u}(\cdot), \sigma(\cdot), \boldsymbol{X}(\cdot)$.
9. Calculate the uncertainty of the mass Δm_k at each time t_k using Eqs. (7)–(11).
10. Calculate the effect of uncertainty on position Δr_k using Eqs. (51) and (52).
11. Calculate the new modified glide-slope constraint γ_{new} using Eqs. (54)–(64), determine if the error $|\gamma_{\text{new}} - \gamma_{\text{old}}| \geqslant \varepsilon$ is satisfied.
12. **end while**
13. **return** $\boldsymbol{u}(\cdot), \sigma(\cdot), \boldsymbol{X}(\cdot)$ as the robust solution.

The position uncertainty arising from mass uncertainty is related to the nominal trajectory, so an iterative approach is required to obtain a convergent solution. Since Algorithm 2 also involves worst-case scenario analysis, providing a rigorous proof of convergence remains challenging at this stage. However, qualitative analysis demonstrates that convergence can still be achieved, as shown in Appendix B.

4.3. Theoretical perspectives on worst-case scenarios

Section 4.2 provided a geometric analysis of worst-case scenarios under uncertainties. In this section, a theoretical analysis approach is presented.

In this paper, the obstacle avoidance constraint for the descent process is Eq. (37). As shown in Eq. (57), at time t_k, the position uncertainty is expressed as $\mathscr{U}_{x,t_k} = \Delta r_{x,t_k}$, $\mathscr{U}_{y,t_k} = \Delta r_{y,t_k}$, and $\mathscr{U}_{z,t_k} = \Delta r_{z,t_k}$. The glide-slope constraint incorporating uncertainty is expressed as:

$$\tan\gamma_{\text{gs}}\sqrt{\left(r_{y,k}+\Delta r_{y,t_k}\right)^2+\left(r_{z,k}+\Delta r_{z,t_k}\right)^2} \leqslant \left(r_{x,k}+\Delta r_{x,t_k}\right) \tag{65}$$

Based on Eqs. (51) and (52), The set of uncertainty can be computed and expressed as:

$$\begin{cases} \Delta r_{x,t_k} \in \mathscr{O}_{x,t_k}, \mathscr{O}_{x,t_k} = \left[\Delta r_{x,t_k,\min}, \Delta r_{x,t_k,\max}\right] \\ \Delta r_{y,t_k} \in \mathscr{O}_{y,t_k}, \mathscr{O}_{y,t_k} = \left[\Delta r_{y,t_k,\min}, \Delta r_{y,t_k,\max}\right] \\ \Delta r_{z,t_k} \in \mathscr{O}_{z,t_k}, \mathscr{O}_{z,t_k} = \left[\Delta r_{z,t_k,\min}, \Delta r_{z,t_k,\max}\right] \end{cases} \tag{66}$$

Therefore, the worst-case scenario with uncertainty is expressed as follows: for any $\Delta r_{x,t_k} \in \mathscr{O}_{x,t_k}$, $\Delta r_{y,t_k} \in \mathscr{O}_{y,t_k}$, $\Delta r_{z,t_k} \in \mathscr{O}_{z,t_k}$, Eq. (65) will hold. This is an optimization problem with an infinite number of constraints that needs to be converted to a problem with finite constraints.

Building on the concept of constructing a new modified glide-slope constraint in Section 4.2, it is expressed as follows:

$$\tan\gamma_{\text{new}}\sqrt{\left(\hat{r}_{y,k}\right)^2+\left(\hat{r}_{z,k}\right)^2} \leqslant \left(\hat{r}_{x,k}-\Delta r_{x,\text{f}}\right) \tag{67}$$

where, $\Delta r_{x,\text{f}}$ and γ_{new} are the vertex and slope angle of the new modified glide-slope constraint. $\hat{r}_{x,k}, \hat{r}_{y,k}, \hat{r}_{z,k}$ are the optimized trajectory results under the new modified glide-slope constraint.

The worst-case scenario can be described as follows: for the optimized trajectory results $\hat{r}_{x,k}, \hat{r}_{y,k}, \hat{r}_{z,k}$, the original glide-slope constraint also holds:

$$\begin{cases} \tan\gamma_{\text{gs}}\sqrt{\left(\hat{r}_{y,k}+\Delta r_{y,t_k}\right)^2+\left(\hat{r}_{z,k}+\Delta r_{z,t_k}\right)^2} \leqslant \left(\hat{r}_{x,k}+\Delta r_{x,t_k}\right) \\ \Delta r_{x,t_k} \in \mathscr{O}_{x,t_k}, \Delta r_{y,t_k} \in \mathscr{O}_{y,t_k}, \Delta r_{z,t_k} \in \mathscr{O}_{z,t_k} \end{cases} \tag{68}$$

The following mathematical construction is used to solve for $\Delta r_{x,\text{f}}$ and $\tan\gamma_{\text{new}}$.

(1) First solve for $\Delta r_{x,\text{f}}$, when $\hat{r}_{y,k} = 0, \hat{r}_{z,k} = 0$, at this point Eq. (67) is irrelevant to $\tan\gamma_{\text{new}}$ and becomes:

$$\Delta r_{x,\text{f}} \leqslant \hat{r}_{x,k} \tag{69}$$

At the same time, Eq. (68) becomes:

$$\tan\gamma_{\text{gs}}\sqrt{\left(\Delta r_{y,t_k}\right)^2+\left(\Delta r_{z,t_k}\right)^2} - \Delta r_{x,t_k} \leqslant \hat{r}_{x,k} \tag{70}$$

Combine Eqs. (69) and (70), one obtains:

$$\begin{cases} \Delta r_{x,\text{f}} \geqslant \max\left(\tan\gamma_{\text{gs}}\sqrt{\left(\Delta r_{y,t_k}\right)^2+\left(\Delta r_{z,t_k}\right)^2} - \Delta r_{x,t_k}\right) \\ \forall\Delta r_{x,t_k} \in \mathscr{O}_{x,t_k}, \forall\Delta r_{y,t_k} \in \mathscr{O}_{y,t_k}, \forall\Delta r_{z,t_k} \in \mathscr{O}_{z,t_k} \end{cases} \tag{71}$$

Eq. (71) and Eq. (54) are corresponding, with Eq. (54) providing the specific maximum value. Thus, the value of $\Delta r_{x,\text{f}}$ is obtained.

(2) Second solve for $\tan\gamma_{\text{new}}$, when $\sqrt{\left(\hat{r}_{y,k}\right)^2+\left(\hat{r}_{z,k}\right)^2} \neq 0$, combining Eqs. (67) and (68), we have:

$$\begin{cases} \tan\gamma_{\text{new}}\sqrt{\left(\hat{r}_{y,k}\right)^2+\left(\hat{r}_{z,k}\right)^2} + \Delta r_{x,\text{f}} \geqslant \\ \max\left(\tan\gamma_{\text{gs}}\sqrt{\left(\hat{r}_{y,k}+\Delta r_{y,t_k}\right)^2+\left(\hat{r}_{z,k}+\Delta r_{z,t_k}\right)^2} - \Delta r_{x,t_k}\right) \\ \forall\Delta r_{x,t_k} \in \mathscr{O}_{x,t_k}, \forall\Delta r_{y,t_k} \in \mathscr{O}_{y,t_k}, \forall\Delta r_{z,t_k} \in \mathscr{O}_{z,t_k} \end{cases} \tag{72}$$

Simplify to:

$$\begin{cases} \tan\gamma_{\text{new}} \geqslant \max\left(\dfrac{\tan\gamma_{\text{gs}}\cdot \text{temp1} - \Delta r_{x,t_k} - \Delta r_{x,\text{f}}}{\text{temp2}}\right) \\ \text{temp1} = \max\left(\sqrt{\left(\widehat{r}_{y,k} + \Delta r_{y,t_k}\right)^2 + \left(\widehat{r}_{z,k} + \Delta r_{z,t_k}\right)^2}\right) \\ \text{temp2} = \sqrt{\left(\widehat{r}_{y,k}\right)^2 + \left(\widehat{r}_{z,k}\right)^2} \\ \forall \Delta r_{x,t_k} \in \mathcal{O}_{x,t_k}, \forall \Delta r_{y,t_k} \in \mathcal{O}_{y,t_k}, \forall \Delta r_{z,t_k} \in \mathcal{O}_{z,t_k} \end{cases} \tag{73}$$

Thus, Eqs. (71) and (73) provide the calculation formulas for calculating the parameters of the new modified glide-slope constraint at time t_k. Both $\Delta r_{x,\text{f}}$ and $\tan\gamma_{\text{new}}$ need to be selected as the maximum values across all time instances. Additionally, to ensure that the problem is a convex optimization problem, it is required that $\tan\gamma_{\text{new}} \geqslant 0$. Thus, there are

$$\begin{cases} \Delta r_{x,\text{f}} = \max\left(\Delta r_{x,\text{f}}(t_k)\right) \\ \tan\gamma_{\text{new}} = \max\left(0, \tan\gamma_{\text{new}}(t_k)\right) \end{cases} \tag{74}$$

Eq. (74) and Eq. (64) correspond to each other. This demonstrates that a mathematically derived approach can also yield the corresponding modified glide-slope constraint. Furthermore, as shown in Eqs. (71) and (73), this method can naturally incorporate both position and velocity uncertainties, highlighting its scalability and adaptability to broader uncertainty scenarios.

4.4. Receding horizon optimization based guidance

The previously mentioned Algorithm 2 is an open-loop trajectory optimization method. In contrast, the receding horizon approach directly integrates optimization with closed-loop guidance, continuously updating an optimal trajectory for a system by solving an optimization problem over a finite planning horizon.

By repeatedly re-solving the optimization problem, traditional receding horizon methods can effectively adapt to disturbances and uncertainties. However, their performance heavily depends on computational speed and update frequency. Although convex optimization offers relatively fast computation, it remains more time-consuming compared to analytical guidance methods.

The key advantage of combining the proposed robust Algorithm 2 with the receding horizon approach lies in its insensitivity to computational time constraints. Traditional receding horizon methods typically require a solution update within 0.1 s to ensure real-time responsiveness. In contrast, the proposed robust Algorithm 2 guarantees collision-free operation throughout the computation cycle, even when updates occur on a second-level timescale. This significantly enhances the robustness of the receding horizon approach, making it more practical for real-world applications where computational resources and update rates may be limited.

Building on the receding horizon approach and Algorithm 2, a closed-loop guidance strategy is proposed in Algorithm 3.

Algorithm 3. Receding horizon optimization based guidance.

1. Define pause conditions $\delta_{\text{r}}, \delta_{\text{v}}$. Define update cycle time ε_{t}.
2. Define terminal time t_{f}, remaining landing time $t_{\text{m}} = t_{\text{f}}$.
3. Define the current state as $\{\boldsymbol{r}_{\text{c}}, \boldsymbol{v}_{\text{c}}, m_{\text{c}}\}$.
4. Calculate initial control input $\boldsymbol{u}_{\text{c}}$ based on Algorithm 2.
5. **while** $\| \boldsymbol{r}_{\text{c}} - \boldsymbol{r}_{\text{f}} \|_2 \geqslant \delta_{\text{r}}$ or $\| \boldsymbol{v}_{\text{c}} - \boldsymbol{v}_{\text{f}} \|_2 \geqslant \delta_{\text{v}}$ **do**
6. Update states $\{\boldsymbol{r}_{\text{c}}, \boldsymbol{v}_{\text{c}}, m_{\text{c}}\}$ based on control input $\boldsymbol{u}_{\text{c}}$ over the cycle time ε_{t}.
7. Update remaining landing time $t_{\text{m}} = t_{\text{m}} - \varepsilon_{\text{t}}$.
8. Calculate Algorithm 2.
9. **if** Algorithm 2 is solved **then**
10. Update current control input $\boldsymbol{u}_{\text{c}}$.
11. **else**
12. Increase remaining landing time $t_{\text{m}} = t_{\text{m}} - \varepsilon_{\text{t}}$.
13. Update current control $\boldsymbol{u}_{\text{c}}$ by recalculating Algorithm 2.
14. **end if**
15. **end while**

It is worth noting that landing time plays a crucial role in solving the problem. Due to the impact of uncertainty, simulations revealed that insufficient remaining landing time can render the problem unsolvable. To address this issue, when the problem becomes infeasible, the remaining landing time is increased to ensure the solvability of Algorithm 2.

The greatest advantage of combining the receding horizon approach with the robust open-loop Algorithm 2 is its ability to ensure that process constraints are never violated at any time, including during the computation process. Additionally, the closed-loop nature of Algorithm 3 significantly enhances the precision of the final landing position.

5. Numerical demonstration

In this section, we apply the proposed robust powered descent guidance algorithm to the Mars landing problem in the presence of mass and fuel consumption uncertainties. First, we provide simulation scenarios and parameter settings. Then, we validate the effectiveness of the proposed algorithm by comparing it with an algorithm that does not take uncertainty into account. Finally, we verify the robustness of the algorithm proposed in this paper under different initial conditions and different uncertainties.

The program construction of the robust powered descent guidance algorithm proposed in this paper uses the CVX framework, with the semidefinite programming solver SDPT3. All simulations were performed on a computer with an Intel Core i9-13900HX processor running at 2.20 GHz and 32 GB of RAM. The MATLAB R2023b software platform was employed for implementing and executing the computational tasks.

5.1. Scenarios and parameters

The kinematic model utilized in this section, detailed in Section 2, is formulated as Problem 1. The specific parameter settings for the lander and constraints used in this paper are presented in Table 1, which are based on the parameters pro-

Table 1 Simulation parameters for lander and constraints.

Parameter	Value
m_{wet} (kg)	1 905
m_{dry} (kg)	1 400
$\boldsymbol{r}_{\text{f}}$ (m)	$\mathbf{0}_{3\times1}$
(m/s)	$\mathbf{0}_{3\times1}$
$T_{\max}$ (N)	13 258.17
$T_{\min}$ (N)	4 971.82
$\bar{\alpha}$ (s/m)	5.09×10^{-4}
$\boldsymbol{g}_{\text{m}}$ (m/s^2)	$[-3.711\,4,\ 0,\ 0]^{\text{T}}$
γ_{gs} (°)	10

vided in Ref. 34. In addition to describing the landing scenario, the parameters ε and $\varepsilon_{\boldsymbol{X}}$ used in Algorithm 1 and Algorithm 2 are set as $\varepsilon_{\boldsymbol{X}} = 1 \times 10^{-3}$ and $\varepsilon = 1 \times 10^{-3}$, respectively. Table 1 is the baseline for the parameter settings, and the initial position and velocity will be adjusted in the different examples below to illustrate the effectiveness of the algorithm.

5.2. Algorithm verification

In this section, we validate the effectiveness of Algorithms 1 and Algorithm 2 through simulation examples. First, we compare the thrust results obtained from the improved lossless convexification algorithm with those from the traditional convex optimization algorithm[34,35] to demonstrate the effectiveness of Algorithm 1. Second, we compare the results without considering mass uncertainty with our proposed algorithm that takes uncertainty into account to validate the effectiveness of Algorithm 2.

When the number of discrete points is set to $N = 100$, the average CPU time for solving discrete Problems 4 or 5 is 0.03 s. For Algorithm 1, the average CPU time is 0.09 s, and for Algorithm 2, it is 0.18 s. The above CPU times do not include the time required for problem transformation using the CVX framework, which takes approximately 2–3 s for the problems considered in this paper. The main factor affecting solution time is the number of discrete points. When N is increased to 200, the CPU time per problem rises to 0.07 s. Additionally, using a direct solver implementation, without the CVX framework, can eliminate transformation time and improve efficiency.

5.2.1. Validation of Algorithm 1

In the simulation, the initial position and velocity are set to $\boldsymbol{r}_0 = [1\,500, 0, 1\,000]^{\text{T}}$m and $\boldsymbol{v}_0 = [-75, 0, 80]^{\text{T}}$m/s, respectively. All other parameters are configured according to Table 1. The results are presented in Figs. 8, 9 and 10.

The subplots in Fig. 8 demonstrate that as the terminal time increases, the thrust profiles obtained using the traditional convex optimization method gradually deviate from the max-min-max control profile, resulting in solutions that increasingly diverge from the optimal trajectory. In contrast, the improved convex optimization method effectively mitigates the linearization error that grows with terminal time, thereby producing solutions that are closer to the true optimal solution.

Figs. 9 and 10 illustrate the evolution of the convergence metric and objective function over iterations. As shown in Fig. 9, the convergence criterion is satisfied as early as the second iteration. Fig. 10 further demonstrates that the terminal mass increases progressively with each iteration.

5.2.2. Validation of Algorithm 2

In this paper, we chose two primary scenarios involving mass and fuel consumption uncertainties. The first scenario considers initial mass uncertainty alone, while the second scenario accounts for both initial mass and fuel consumption uncertainties. To demonstrate the effectiveness of the proposed robust approach, a baseline method using convex optimization without uncertainty consideration, as introduced in Ref. 34, is also implemented and referred to as Deterministic Optimization (DO). The proposed worst-case robust optimization method, outlined in Algorithm 2, is denoted as WC-RO. Additionally, we compare our approach with the robust trajectory optimization method in Ref. 25, which is based on polynomial chaos expansion and probabilistic methods, referred to as PCE-RO.

In the first scenario, the initial position and velocity are set to $\boldsymbol{r}_0 = [1\,500, 0, 1\,000]^{\text{T}}$m and $\boldsymbol{v}_0 = [-75, 0, 80]^{\text{T}}$m/s, respectively. The initial mass uncertainty, corresponding to Eq. (7), is defined as $\Delta m_{\text{wet}} \in [-20, 20]$ kg. The curves of mass under initial mass uncertainty are displayed in Fig. 11. The trajectories and glide-slope angles using DO are presented in Figs. 12 and 13. The results obtained by WC-RO are presented in Figs. 14 and 15.

Fig. 11 illustrates the impact of initial mass uncertainty on the mass at each time step, forming a uniform tubular structure. As shown in Figs. 12 and 13, the nominal trajectory satisfies the glide-slope angle constraint. However, the trajectories with uncertainties partially violate the glide-slope angle constraint, resulting in obstacle avoidance failure and, in some cases, collision with the ground at the terminal time. In contrast, the results shown in Figs. 14 and 15 employ the robust powered descent guidance algorithm proposed in this paper, which accounts for the effects of initial mass uncertainty. It can be observed that both the nominal trajectory and the uncertain trajectories satisfy the glide-slope angle constraint and maintain an altitude above the ground at the terminal time. The same robust optimization results are obtained through geometric analysis and theoretical analysis.

5.2.3. Comparison with PCE-RO

In the second scenario, the initial position and velocity are set to $\boldsymbol{r}_0 = [1\,500, -500, -2\,000]^{\text{T}}$m and $\boldsymbol{v}_0 = [-75, 0, -50]^{\text{T}}$m/s, respectively. The initial mass uncertainty, corresponding to Eq. (12), is defined as $\Delta m_{\text{wet}} \in [-20, 20]$ kg. The fuel consumption uncertainty, corresponding to Eq. (12), is defined as $-0.05\,\bar{\alpha} \leqslant \Delta\alpha \leqslant 0.05\,\bar{\alpha}$. The parameter settings for PCE-RO are kept consistent with those in Ref. 25. Assuming that the distribution of uncertainty factors is uniform with upper and lower bounds. The curves of nominal and uncertain mass are displayed in Fig. 16. The trajectories and glide-slope angles obtained by DO are presented in Figs. 17 and 18. The results obtained by WC-RO and PCE-RO are presented in Figs. 19–22.

Fig. 16 illustrates the impact of initial mass and fuel consumption uncertainties on the mass at each time step, forming a "gradually expanding tubular" structure. This occurs because the engine remains active throughout the descent, and as time progresses, the uncertainty in fuel consumption

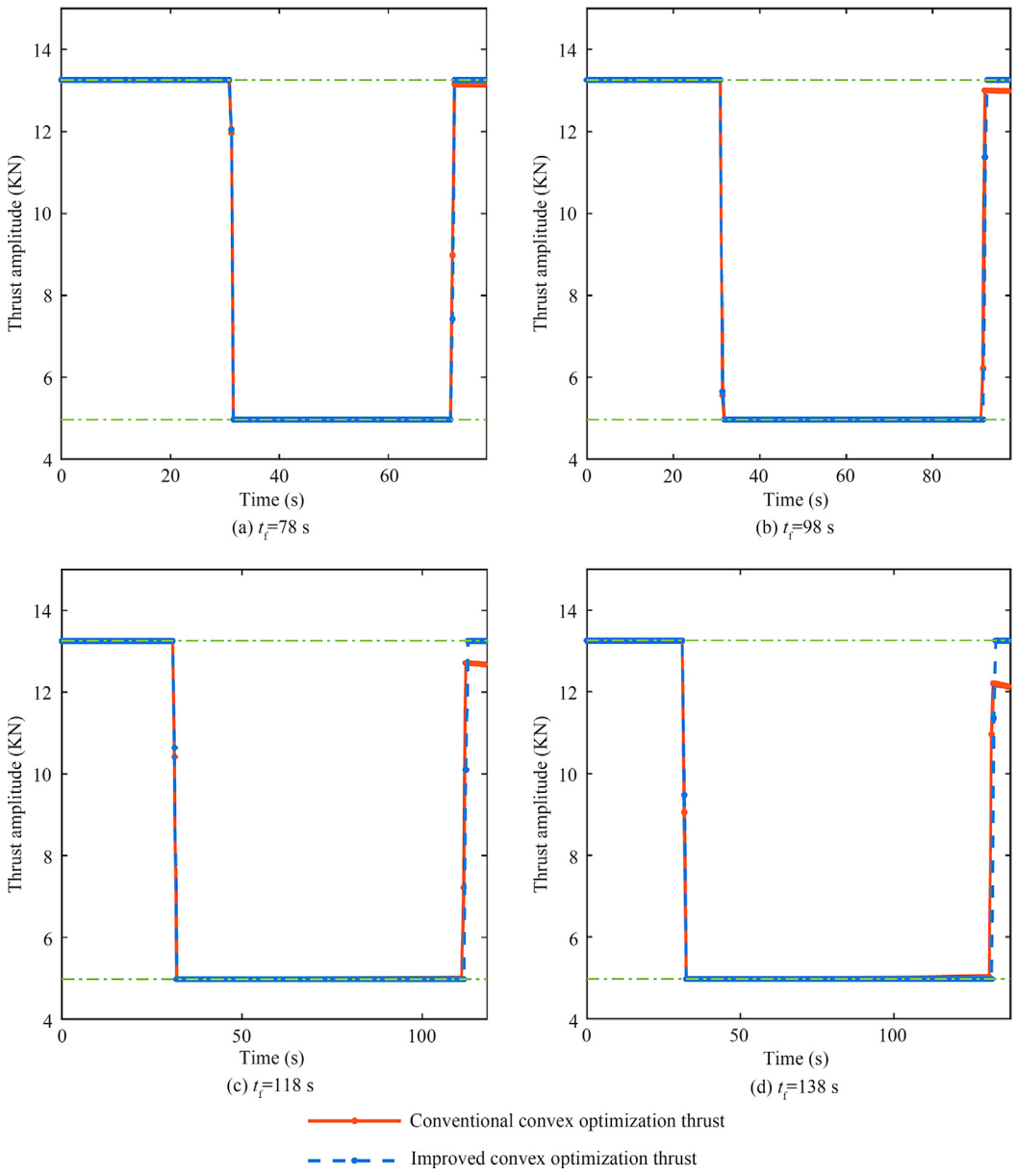


Fig. 8 Thrust comparison between conventional and improved convex optimization.

accumulates, leading to an increase in mass uncertainty. Similar to the first scenario, the nominal trajectory in Figs. 17 and 18 satisfies the glide-slope angle constraint. However, when uncertainties such as those arising from initial mass and fuel consumption are introduced, some resulting trajectories violate the constraint in specific regions. These violations compromise the system's ability to perform effective obstacle avoidance and, in some cases, even lead to terminal ground collisions.

In contrast, the results shown in Figs. 19–22 employ WC-RO and PCE-RO, which takes into account the effects of initial mass and fuel consumption uncertainties. It can be seen that both the nominal trajectory and the uncertain trajectories satisfy the glide-slope angle constraint and maintain an altitude above the ground at the terminal time.

Both the probability-based method PCE-RO and the proposed algorithm WC-RO demonstrate the capability to generate robust optimal trajectories while accounting for uncertainties in both initial mass and fuel consumption coefficient. Although the results of the two approaches are comparable, our method offers distinct advantages:

(1) As shown in Figs. 20–22, the WC-RO method enforces the constraints more precisely than the PCE-RO method, ensuring that the worst-case scenario is exactly satisfied. In contrast, the PCE-RO method tends to be more conservative. This is primarily because probability-based approaches approximate bounded closed sets using probability distributions, which could lead to more conservative constraint handling.
(2) The PCE-RO method requires expansion of stochastic states into higher-order deterministic states, which significantly increases the scale of the resulting optimization problem along with computational burden. As state variables are expanded, both input and path con-

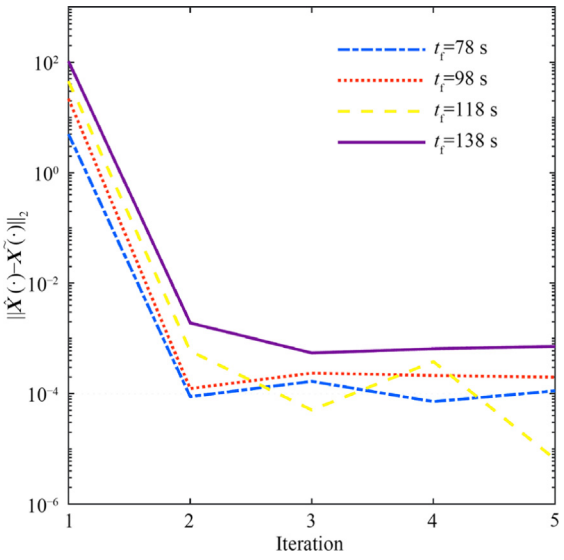


Fig. 9 Convergence metric at each iteration.

straints become functions of higher-order states, hindering their direct handling through lossless convexification methods. For instance, while the glide-slope constraint constitutes a second-order cone constraint with respect to original state variables, it transforms into a non-convex form when expressed through higher-order states in the stochastic setting. This necessitates successive linearization for convexification, thereby further increasing computational time.

5.3. Algorithm robustness verification

In this section, numerical simulations are conducted to validate the robustness of the proposed robust powered descent guidance algorithm, which accounts for mass and fuel consumption uncertainties. The simulations evaluate the algorithm's effectiveness and adaptability under varying initial positions and velocities. Additionally, the robustness of the algorithm is assessed under different glide-slope angle constraints.

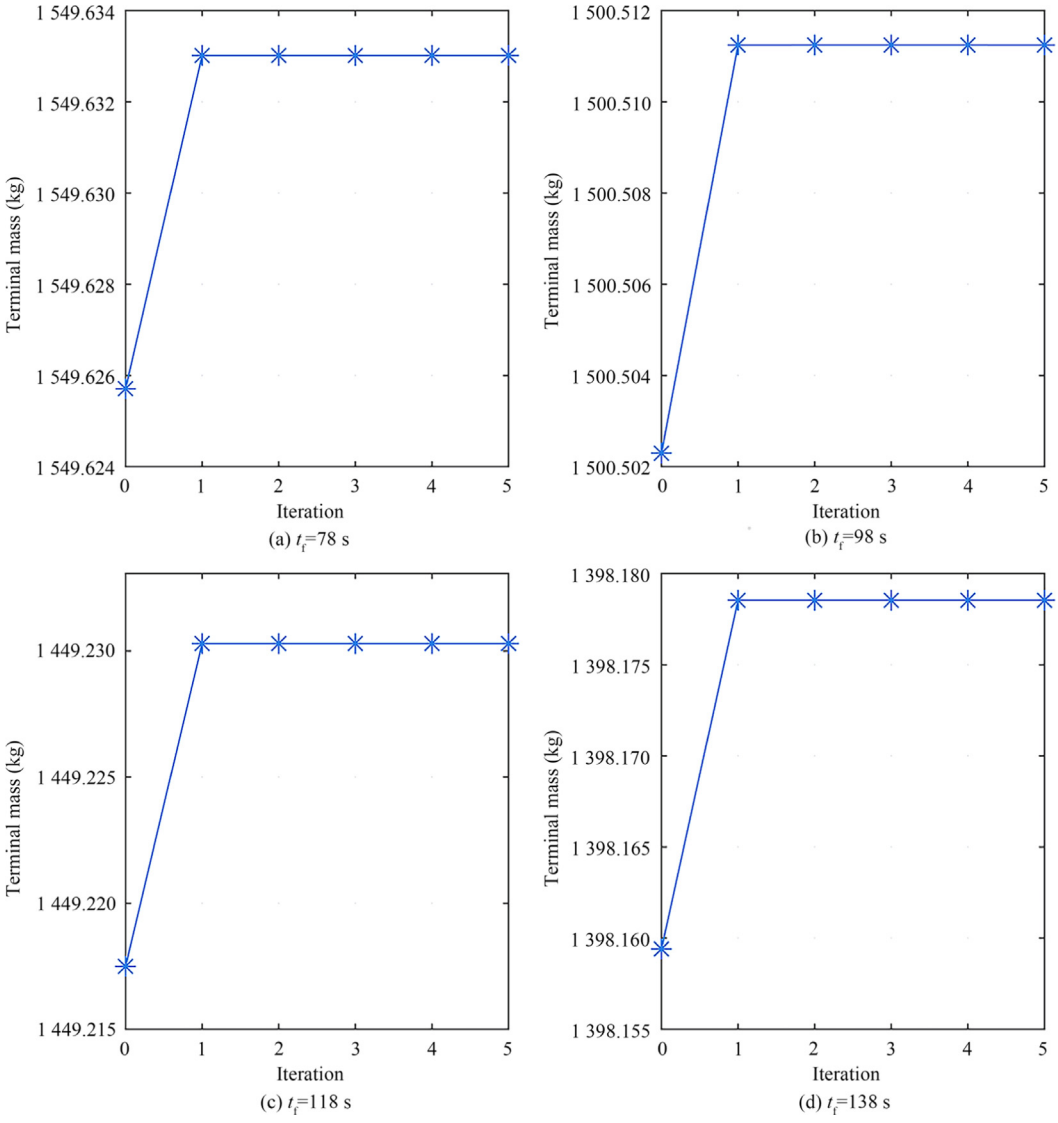


Fig. 10 Terminal mass at each iteration.

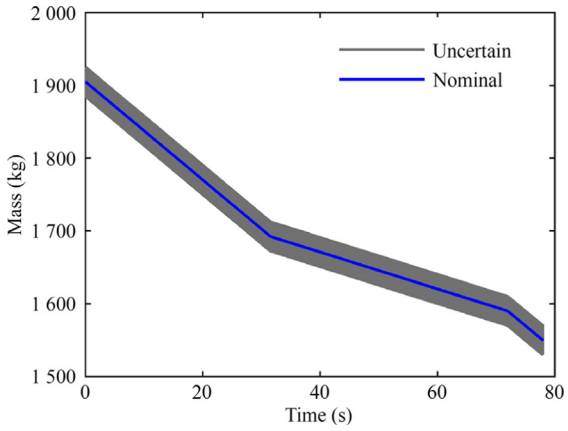


Fig. 11 Mass curves under initial mass uncertainty (the first scenario).

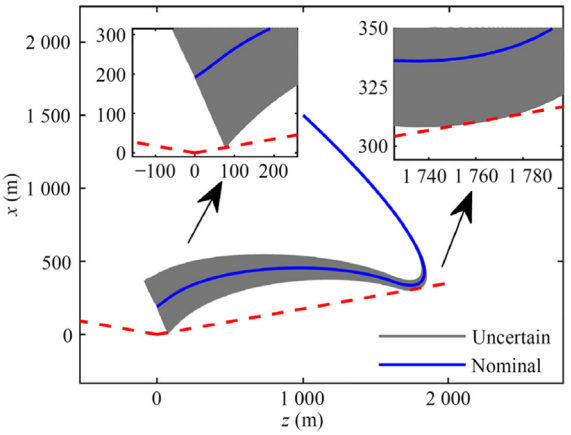


Fig. 14 Nominal and uncertain trajectories obtained by WC-RO (the first scenario).

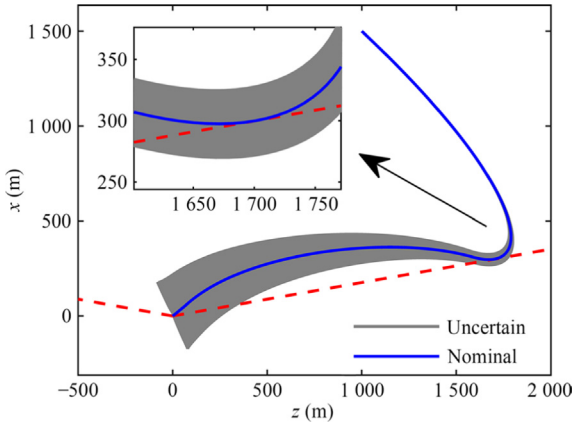


Fig. 12 Nominal and uncertain trajectories obtained by DO (the first scenario).

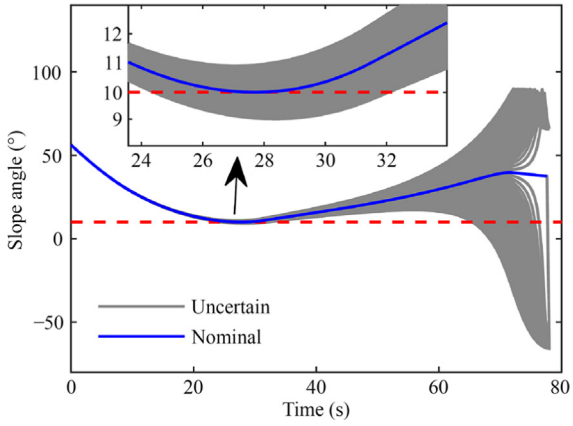


Fig. 13 Nominal and uncertain glide-slope angles obtained by DO (the first scenario).

In the case of different initial positions, the initial position is set to $\boldsymbol{r}_0 = [1\,500, 0, z]^{\mathrm{T}}$m, where z ranges from $-3\,000$ m to 3 000 m with a step of 1 000 m. The initial velocity is fixed at $\boldsymbol{v}_0 = [-75, 0, -10]^{\mathrm{T}}$m/s. All other parameters are configured according to Table 1. The simulation results obtained are shown in Figs. 23–25.

In the case of different initial velocities, the initial position is set to $\boldsymbol{r}_0 = [1\,500, 0, 2\,000]^{\mathrm{T}}$ m. The initial velocity is set to $\boldsymbol{v}_0 = [-75, 0, v_z]^{\mathrm{T}}$ m/s, where v_z ranges from -60 m/s to 60 m/s with a step of 30 m/s. All other parameters are configured according to Table 1. The simulation results obtained are shown in Figs. 26–28.

From the simulation results, we can observe that the robust powered descent algorithm proposed in this paper can satisfy the process constraints in the presence of initial mass and fuel consumption uncertainties under different initial positions and velocities. Additionally, trajectories obtained under different initial conditions can achieve the purpose of fuel optimal.

In the case of different glide-slope angle constraints, the initial position is set to $\boldsymbol{r}_0 = [1\,500, 0, 500]^{\mathrm{T}}$ m. The initial velocity is set to $\boldsymbol{v}_0 = [-75, 0, 90]^{\mathrm{T}}$ m/s. The glide-slope angle is set from 4° to 13° with a step of 3°. All other parameters are configured according to Table 1. The simulation results obtained are shown in Figs. 29 and 30.

Remark. All of the nominal trajectories in this paper are directly above the landing site, with the final touchdown accomplished by a vertical landing.

5.4. Receding horizon based closed-loop guidance

In the closed-loop scenario, the initial position and velocity are set to $\boldsymbol{r}_0 = [1\,500, 0, -2\,000]^{\mathrm{T}}$m and $\boldsymbol{v}_0 = [-75, 0, -50]^{\mathrm{T}}$m/s, respectively. The initial mass uncertainty, corresponding to Eq. (12), is defined as $\Delta m_{\mathrm{wet}} \in [-20, 20]$ kg. The fuel consumption uncertainty, corresponding to Eq. (12), is defined as $-0.05\,\bar{\alpha} \leqslant \Delta\alpha \leqslant 0.05\,\bar{\alpha}$. The update cycle time is set to $\varepsilon_{\mathrm{t}} = 5$ s and $\varepsilon_{\mathrm{t}} = 2$ s. A shorter update period will significantly

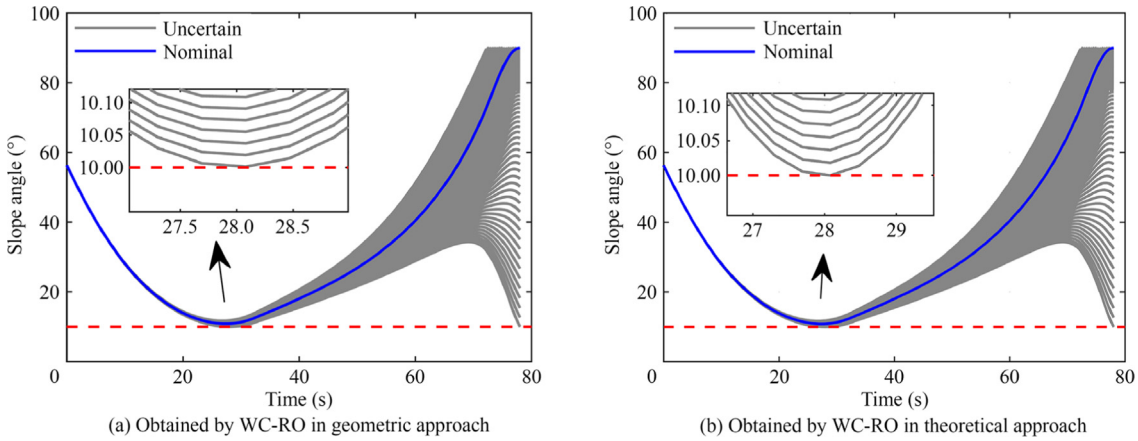


Fig. 15 Nominal and uncertain glide-slope angles obtained by WCRO (the first scenario).

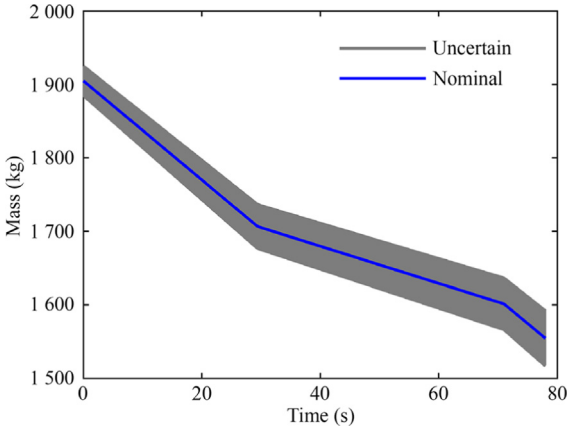


Fig. 16 Mass curves under initial mass and fuel consumption uncertainties (the second scenario).

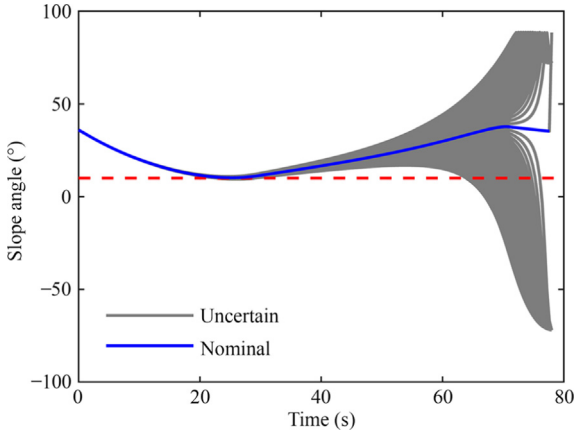


Fig. 18 Nominal and uncertain glide-slope angles obtained by DO (the second scenario).

improve the accuracy of terminal position and velocity, but it cannot reflect the superiority of the robust algorithm in this paper. Using Monte Carlo simulation, random experiments were conducted. The results are shown in Figs. 31–34.

Figs. 31 and 32 demonstrate that under uncertainty, the closed loop algorithm effectively avoids obstacles and guides the system to a terminal position close to the landing site.

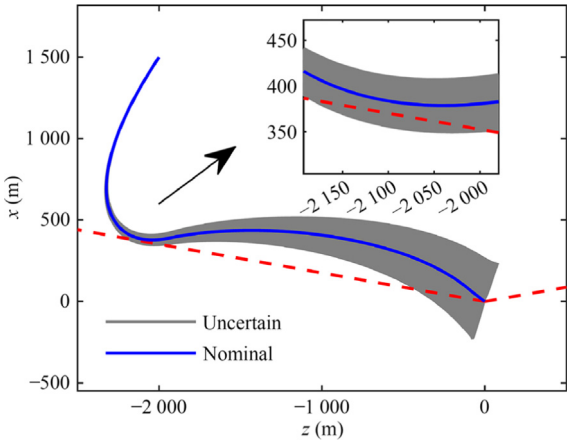


Fig. 17 Nominal and uncertain glide-slope angles obtained by DO (the second scenario).

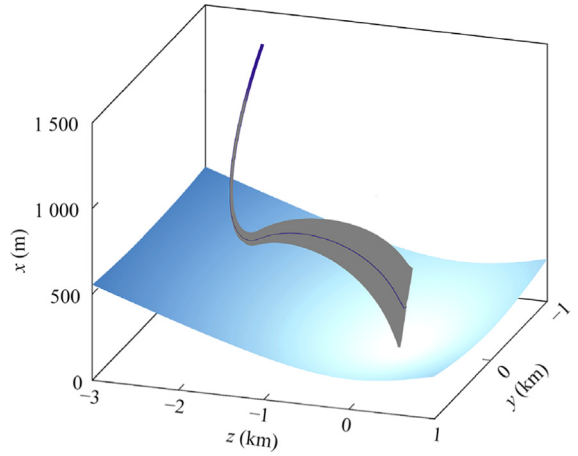
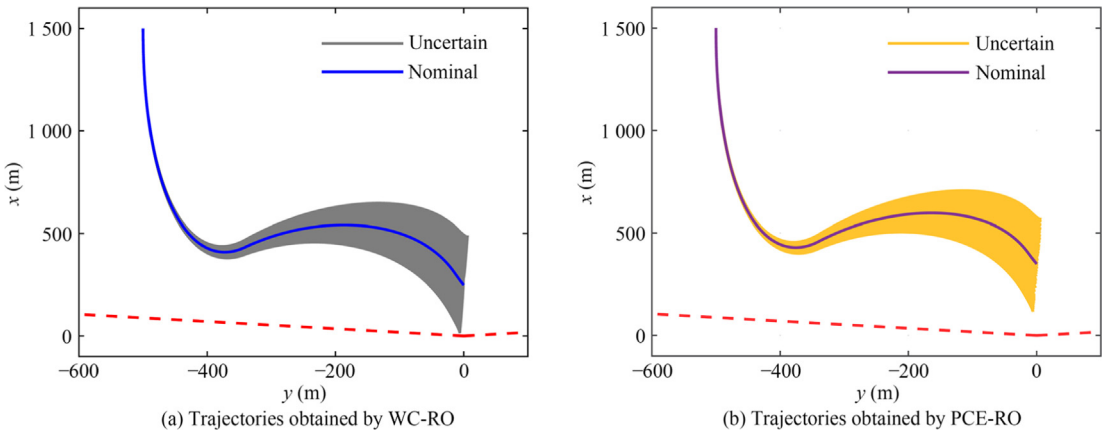
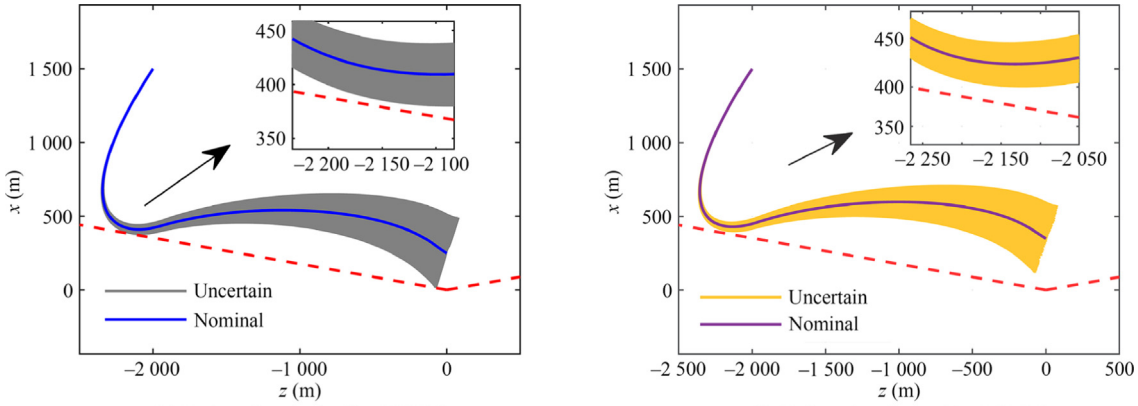


Fig. 19 Nominal and uncertain 3D trajectories obtained by WC-RO (the second scenario).



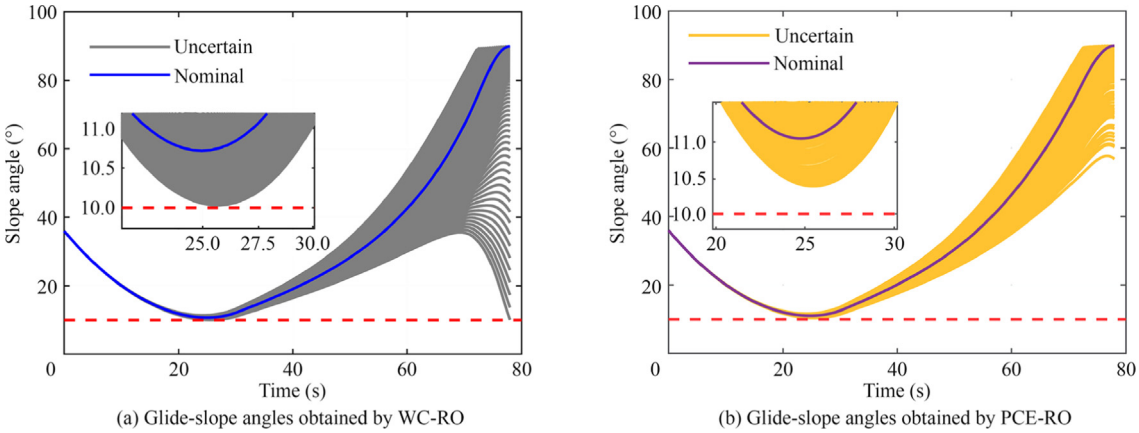
(a) Trajectories obtained by WC-RO (b) Trajectories obtained by PCE-RO

Fig. 20 Nominal and uncertain trajectories in Oxy plane (the second scenario).



(a) Trajectories obtained by WC-RO (b) Trajectories obtained by PCE-RO

Fig. 21 Nominal and uncertain trajectories in Oxz plane (the second scenario).



(a) Glide-slope angles obtained by WC-RO (b) Glide-slope angles obtained by PCE-RO

Fig. 22 Nominal and uncertain glide-slope angles (the second scenario).

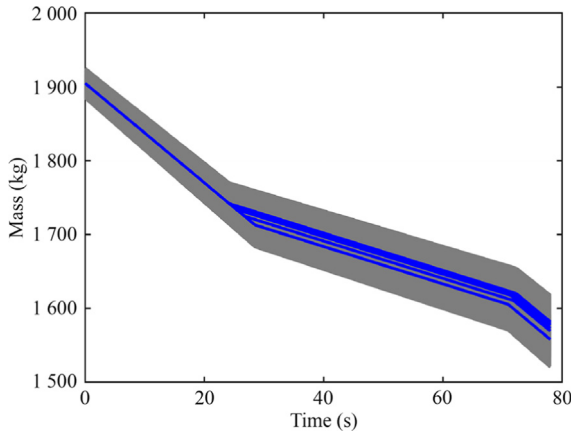


Fig. 23 Nominal and uncertain mass under different initial positions.

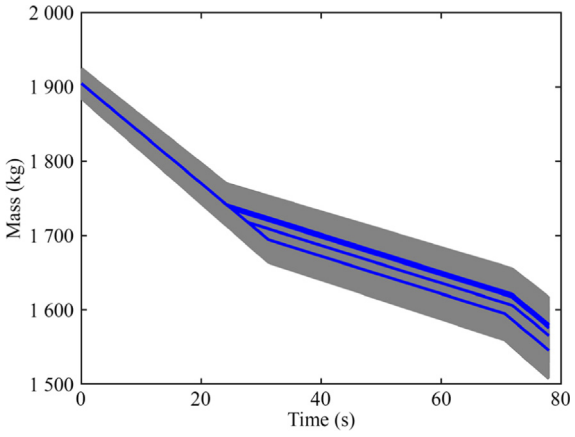


Fig. 26 Nominal and uncertain mass under different initial velocities.

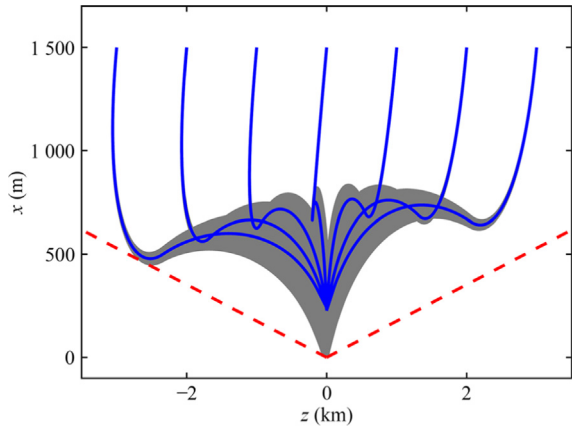


Fig. 24 Nominal and uncertain trajectories considering uncertainties under different initial positions.

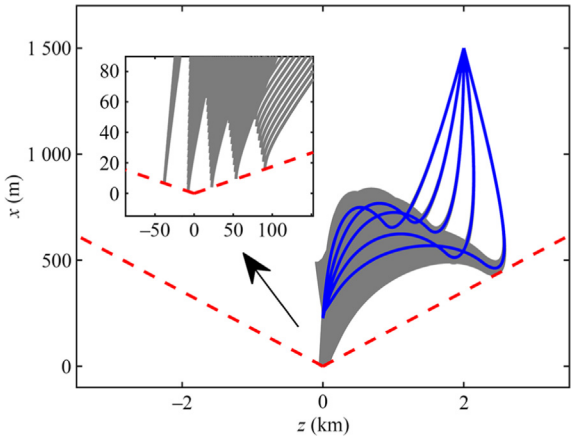


Fig. 27 Nominal and uncertain trajectories considering uncertainties under different initial velocities.

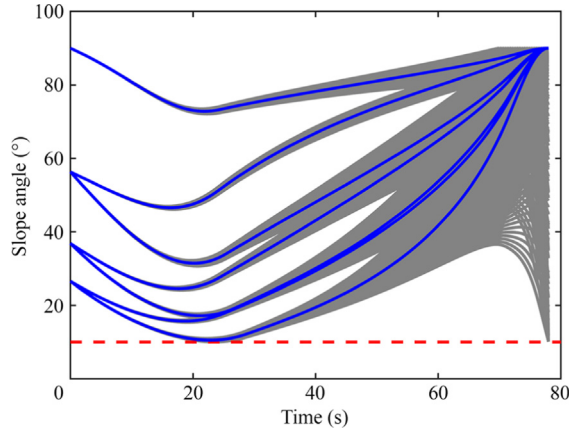


Fig. 25 Nominal and uncertain glide-slope angles considering uncertainties under different initial positions.

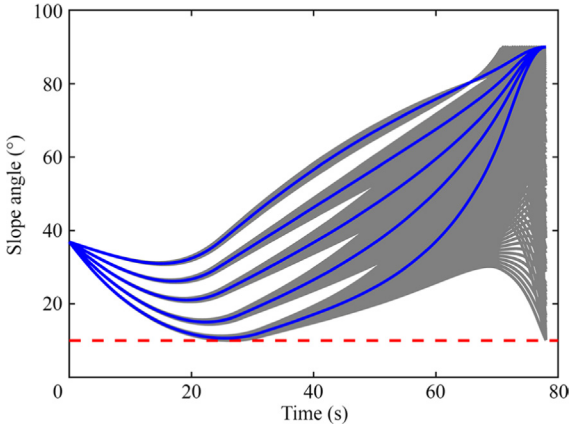


Fig. 28 Nominal and uncertain glide-slope angles considering uncertainties under different initial velocities.

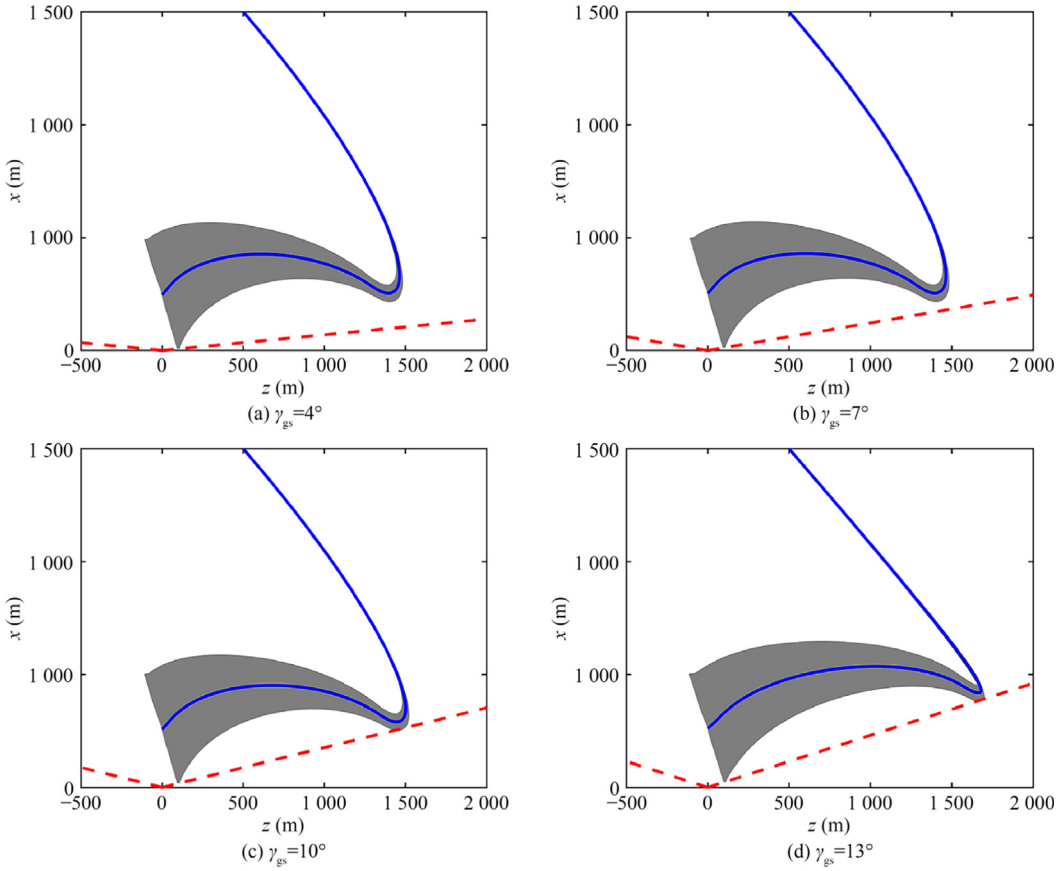


Fig. 29 Nominal and uncertain trajectories considering uncertainties under different glide-slope angle constraints.

In Figs. 33 and 34, the distributions of terminal position and velocity appear linear. This is because the x and z directions are independent and both are influenced by mass uncertainty in a way that results in a linear effect on position and velocity.

6. Conclusions

This paper presents a robust powered descent guidance algorithm designed to ensure safe and accurate landings under uncertainties in mass and fuel consumption. Building upon the standard lossless convexification framework, an improved sequential convex optimization method is proposed to reduce linearization errors and enhance solution optimality and accuracy. To handle mass uncertainty, an open-loop robust trajectory optimization approach is developed. Through geometric reasoning and theoretical analysis of worst-case scenarios, the method guarantees constraint satisfaction and safe landing performance, even under adverse conditions. Additionally, the framework allows for the incorporation of initial position and velocity uncertainties within the same robust optimization scheme. To further improve terminal accuracy and robustness, a closed-loop receding horizon guidance method is introduced. This approach integrates the computational efficiency of convex optimization with a feedback mechanism that ensures continued satisfaction of process constraints throughout the descent. Simulation results confirm that the proposed algorithm achieves robust performance across various uncertainty scenarios and landing conditions, maintaining feasibility with respect to glide-slope and terminal state constraints. The proposed framework is applicable to a wide range of scenarios, including planetary landings and reusable rocket guidance, where reliability under uncertainty is critical.

CRediT authorship contribution statement

Duozhi GAO: Investigation, Writing – original draft, Visualization, Writing – review & editing. **Yanning GUO:** Writing – review & editing, Supervision, Funding acquisition, Resources, Project administration. **Edoardo FADDA:** Supervision, Writing – review & editing. **Youmin GONG:** Supervision, Writing – review & editing, Validation, Resources. **Chuanjiang LI:** Resources, Supervision, Funding acquisition. **Paolo BRANDIMARTE:** Supervision, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

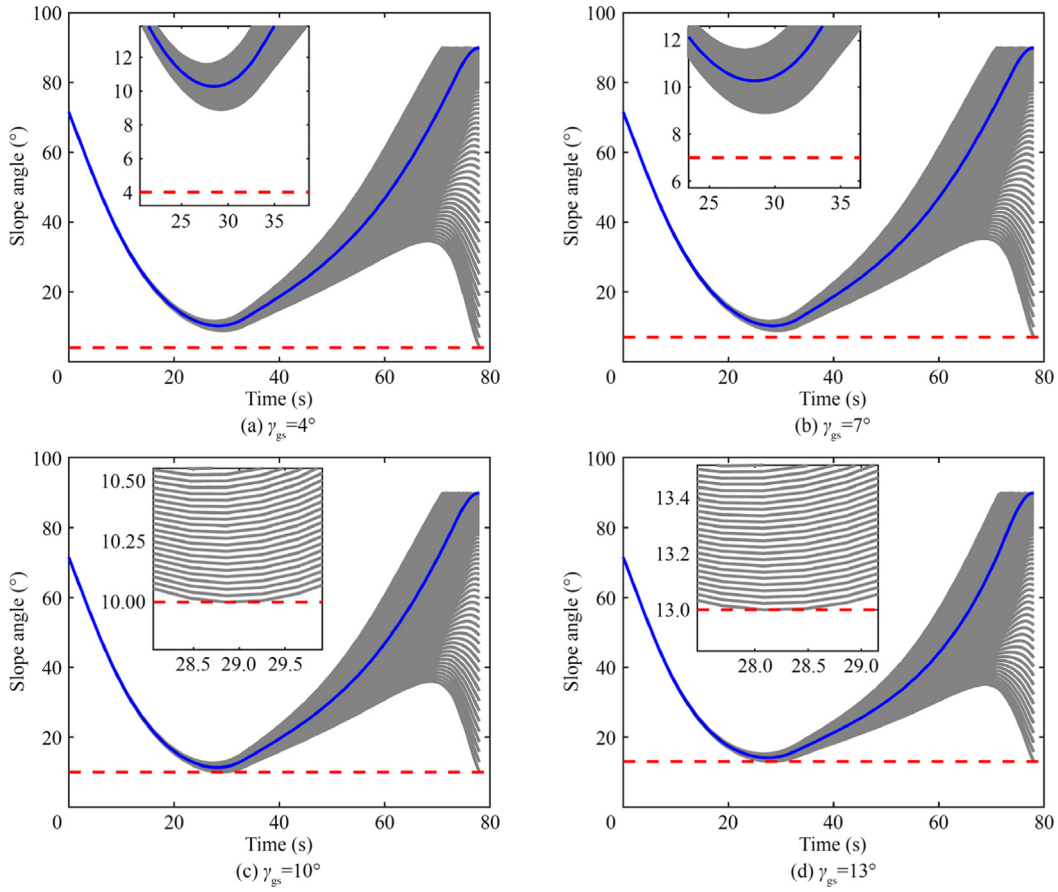


Fig. 30 Nominal and uncertain glide-slope angles considering uncertainties under different glide-slope angle constraints.

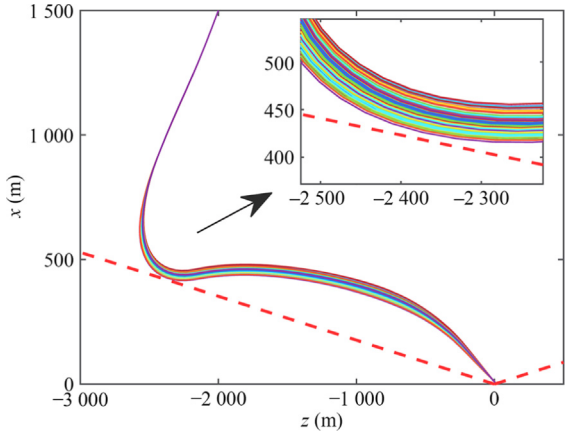


Fig. 31 Monte Carlo trajectory results under 5 s.

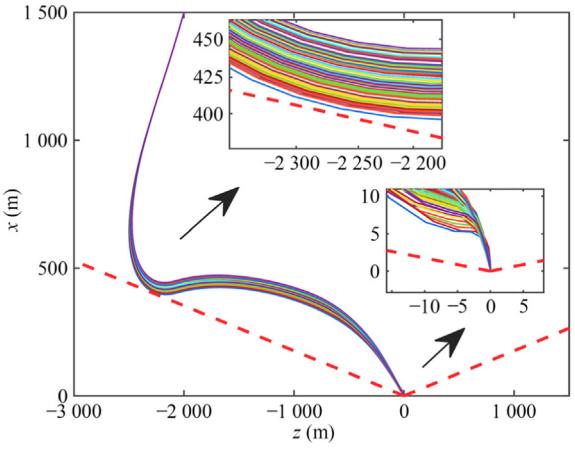


Fig. 32 Monte Carlo trajectory results under 2 s.

Acknowledgements

This study was co-supported by the National Natural Science Foundation of China (No. U23B6001), the Natural Science Foundation of Heilongjiang Province, China (No. LH2022F023), and the Fundamental Research Funds for the Central Universities, China (No. HIT.OCEF.2023009).

Appendix A. Convergence analysis of Algorithm 1

Based on Refs. 36–39, the proof of the convergence of Algorithm 1 is divided into two parts. The first part proves the existence of an optimal solution, while the second part demonstrates the convergence of the sequential convex opti-

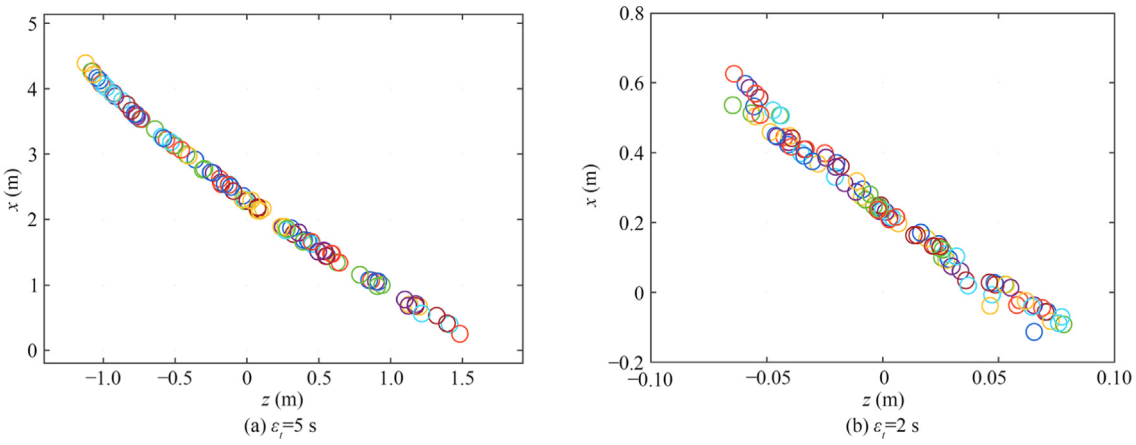


Fig. 33 Distribution of terminal positions.

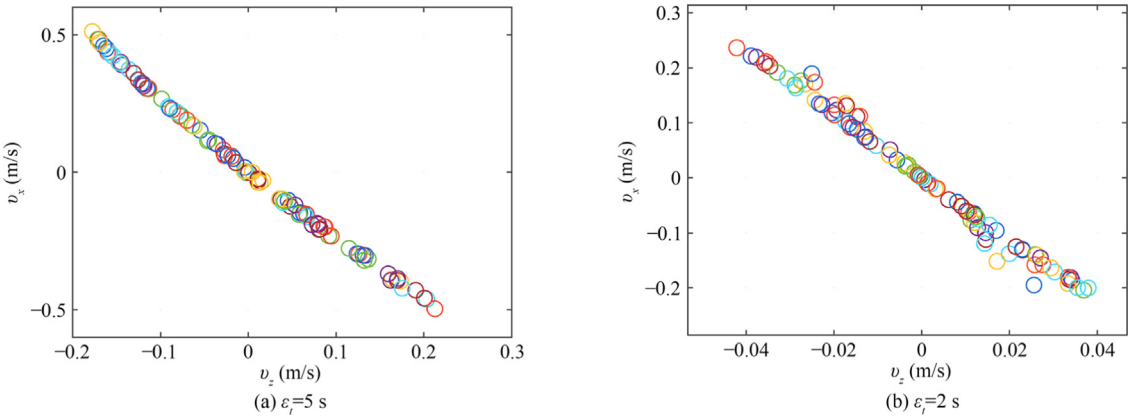


Fig. 34 Distribution of terminal velocities.

mization, showing that the convergence result is at least a local optimal solution of the original problem.

First, based on Refs. 34,38,39, the existence of the optimal solution for Problem 5 is discussed.

Theorem A1. Assume that the set $\mathscr{R}$ of all feasible state and control variables for Problem 5 are nonempty, consider the following general optimal control problem[40]:

$$\min_{\boldsymbol{x},\boldsymbol{u}} J[\boldsymbol{x}(\cdot),\boldsymbol{u}(\cdot)] = \varphi[\boldsymbol{x}(t_{\mathrm{f}})] + \int_{t_0}^{t_{\mathrm{f}}} g(\boldsymbol{x},\boldsymbol{u})\mathrm{d}t$$

$$\text{subject to}: \begin{cases} \dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x},\boldsymbol{u}) \\ \boldsymbol{x}(t) \in \boldsymbol{X}, \quad \boldsymbol{u}(t) \in U, \quad \forall t \in [t_0, t_{\mathrm{f}}] \\ \boldsymbol{x}(t_0) \in \boldsymbol{X}_0, \quad \boldsymbol{x}(t_{\mathrm{f}}) \in \boldsymbol{X}_{\mathrm{f}} \end{cases}$$

Suppose that

(1) There exists a compact set $\mathscr{C}$ such that all feasible trajectories satisfy $(t, \boldsymbol{x}(t)) \in \mathscr{C}$ for all $t \in [t_0, t_{\mathrm{f}}]$.
(2) The set of all feasible $[\boldsymbol{x}(t_0), t_{\mathrm{f}}, \boldsymbol{x}(t_{\mathrm{f}})] \in \mathscr{F}$ is closed such that $\boldsymbol{x}(t_0) \in \boldsymbol{X}_0, \boldsymbol{x}(t_{\mathrm{f}}) \in \boldsymbol{X}_{\mathrm{f}}$.
(3) The control set U is compact.
(4) Functions $\boldsymbol{f}$ and g are continuous on $\boldsymbol{X} \times U$.
(5) For each $(t, \boldsymbol{x}(t)) \in \mathscr{C}$, the set $\mathscr{H}$ defined below is convex

$$\mathscr{H}(t,\boldsymbol{x}) = \{(z_1, \boldsymbol{z}_2) : z_1 \geqslant g(\boldsymbol{x},\boldsymbol{u}), \boldsymbol{z}_2 = \boldsymbol{f}(\boldsymbol{x},\boldsymbol{u}), \boldsymbol{u} \in U\} \tag{A1}$$

Then, there exists an optimal solution pair $[\boldsymbol{x}^*(\cdot), \boldsymbol{u}^*(\cdot)]$, such that

$$J(\boldsymbol{x}^*, \boldsymbol{u}^*) \leqslant J[\boldsymbol{x},\boldsymbol{u}], \quad \forall [\boldsymbol{x},\boldsymbol{u}] \in \mathscr{R}$$

The complete proof of this theorem is provided in Ref. 40. This result has been applied in Ref. 34 to establish the existence of an optimal solution for Problem 2. Furthermore, Refs. 38 and 39 extended its application to prove the existence of optimal solutions for sequential convex optimization in both entry and ascent phases. Next, the conditions are verified one by one to establish the existence of an optimal solution for Problem 5.

(1) For Problem 5, the terminal time t_{f} is given and fixed, all the state variables are bounded. Thus, Condition 1 is satisfied.
(2) For Problem 5, the initial and final states are given and fixed, then $[\boldsymbol{x}(t_0), t_{\mathrm{f}}, \boldsymbol{x}(t_{\mathrm{f}})] \in \mathscr{F}$ is closed. Condition 2 is satisfied.
(3) The control set U is compact as constrained in Eqs. (35) and (42).
(4) The system is linear, and the objective function is affine. Therefore, Condition 4 is satisfied.

(5) The constraints in Problem 5 are either linear or second-order cone, so corresponding the set $\mathscr{H}$ will be convex, and Condition 5 is satisfied.

Thus, there exists an optimal solution for Problem 5.

Second, based on Refs. 36 and 37, the convergence of Algorithm 1 is discussed. Problem 5 is an SOCP problem based on trajectory $\widetilde{X}(\cdot)$. Specifically, the sequence problem is constructed based on a series of $\widetilde{z}(\cdot)$. Problem 5 with initial trajectory $\widetilde{X}^{[q]}(\cdot)$ is abbreviated as $\mathrm{P5}^{[q]}$. With the following assumptions and lemmas, it is shown that Algorithm 1 guarantees convergence, with the converged solution being at least a local optimal solution of Problem 2.

Assumption A1. Assume that after the discretization process, the existence of an optimal solution to discretized problem $\mathrm{P5}^{[q]}$ and each subproblem (proved above) is guaranteed.

Assumption A2. Assume that the quadratic-cone constraints in each problem $\mathrm{P5}^{[q]}$ are strictly feasible, and there exists a unique solution for each problem $\mathrm{P5}^{[q]}$.

Lemma A1. The feasible set of problem $\mathrm{P5}^{[q]}$ is bounded.

Proof. The control set is bounded by Eqs. (37) and (42), and the system in $\mathrm{P5}^{[q]}$ is a linear discrete system, with the system and input matrices given by Eq. (33). The terminal time t_f is given and fixed. Thus, the state set is also bounded. Moreover, since the states are constrained by Eqs. (37) and (38), their bounded nature remains unchanged.

Lemma A2. The optimal solution of problem $\mathrm{P5}^{[q]}$, if exists, is unique.

Proof. Since the objective function in problem $\mathrm{P5}^{[q]}$ is $-z(t_{N+1})$, the objective function $f(z)$ is strictly convex, then for any $z_1 \neq z_2$ and $\lambda \in (0,1)$, we have:

$$f(\lambda z_1 + (1-\lambda)z_2) < \lambda f(z_1) + (1-\lambda) f(z_2) \tag{A2}$$

Assume z_1^* and z_2^* are both optimal solutions, meaning $f(z_1^*) = f(z_2^*) = f^*$.

Consider $z_\lambda = \lambda z_1^* + (1-\lambda) z_2^*$, which belongs to the convex set $\mathscr{C}$. By strict convexity,

$$f(z_\lambda) < \lambda f(z_1^*) + (1-\lambda) f(z_2^*) = f^* \tag{A3}$$

This contradicts the assumption that z_1^* and z_2^* are optimal. Thus, $z_1^* = z_2^*$, proving uniqueness.

Lemma A3. Any feasible solution to $\mathrm{P5}^{[q]}$ is also feasible to Problem 2.

Proof. Based on Assumption 1, disregarding differences between the discrete and continuous problems, the only difference between $\mathrm{P5}^{[q]}$ and Problem 2 is the thrust constraint, thus it is only necessary to prove that any solution satisfying Eq. (42) also satisfies Eq. (21). For any $z \in [\widetilde{z}(t_k), z_u]$, there exists $\bar{z} \in [\widetilde{z}(t_k), z_u]$ such that

$$\mathrm{e}^{-z} = \mathrm{e}^{-\widetilde{z}(t_k)}\left[1 - [z(t_k) - \widetilde{z}(t_k)] + \tfrac{1}{2}[z(t_k) - \widetilde{z}(t_k)]^2\right] - \mathrm{e}^{-\bar{z}}\frac{[z-\widetilde{z}(t_k)]^3}{6} \tag{A4}$$

Since $-\mathrm{e}^{-\bar{z}}\frac{[z-\widetilde{z}(t_k)]^3}{6} \leqslant 0$, the first inequality in Eq. (42) represents a stricter scaled constraint compared to Eq. (21). In the same way, for any $z \in [z_l, z_u]$, there exists $\bar{z} \in [z_l, z_u]$ such that

$$\mathrm{e}^{-z} = \mathrm{e}^{-\widetilde{z}(t_k)}[1 - (z(t_k) - \widetilde{z}(t_k))] + \mathrm{e}^{-\bar{z}}\frac{(z-\widetilde{z}(t_k))^2}{2} \tag{A5}$$

Since $\mathrm{e}^{-\bar{z}}\frac{(z-\widetilde{z}(t_k))^2}{2} \geqslant 0$, the second inequality in Eq. (42) also represents a stricter scaled constraint compared to Eq. (21). Therefore, the solution to $\mathrm{P5}^{[q]}$ is also feasible to Problem 2.

Lemma A4. Under Assumption 2, if the optimal solution of $\mathrm{P5}^{[q]}$ is $\widetilde{X}^{[q]}(\cdot)$ itself, then $\widetilde{X}^{[q]}(\cdot)$ is a Karush-Kuhn-Tucker (KKT) solution of Problem 2.

Proof. Based on Ref. 36, $\mathrm{P5}^{[q]}$ can be can be described in the following standardized form:

$$\min_{\boldsymbol{y}} \boldsymbol{c}^{\mathrm{T}}\boldsymbol{y}$$
$$\text{subject to}: \begin{cases} \boldsymbol{H}\boldsymbol{y} \leqslant \boldsymbol{p} \\ \boldsymbol{g}_i(\boldsymbol{y}^{[k]}) + \nabla \boldsymbol{g}_i^{\mathrm{T}}(\boldsymbol{y}^{[k]})(\boldsymbol{y} - \boldsymbol{y}^{[k]}) \leqslant 0, \quad i = 1,2,\cdots,l \\ \boldsymbol{\Lambda}\boldsymbol{y} - \boldsymbol{b} \geqslant_K 0 \end{cases} \tag{A6}$$

Problem 2 can be can be described in the following standardized form:

$$\begin{aligned} &\min_{\boldsymbol{y}} \boldsymbol{c}^{\mathrm{T}}\boldsymbol{y} \\ &\text{subject to}: \boldsymbol{H}\boldsymbol{y} \leqslant \boldsymbol{p} \\ &\boldsymbol{g}_i(\boldsymbol{y}) \leqslant 0, \quad i = 1,2,\cdots,l \\ &\boldsymbol{\Lambda}\boldsymbol{y} - \boldsymbol{b} \geqslant_K 0 \end{aligned} \tag{A7}$$

With the help of Duality theorem, it can be obtained that

$$\boldsymbol{H}^{\mathrm{T}}\boldsymbol{\iota}^* + \sum_{i=1}^{l} \boldsymbol{v}_i^* \nabla \boldsymbol{g}_i(\boldsymbol{y}^{[k]}) + \boldsymbol{c} = \boldsymbol{\Lambda}^{\mathrm{T}}\boldsymbol{s}^* \tag{A8}$$

$$\boldsymbol{\iota}^{*\mathrm{T}}(\boldsymbol{H}\boldsymbol{y}^{[k]} - \boldsymbol{p}) = 0 \tag{A9}$$

$$\boldsymbol{v}_i^* \boldsymbol{g}_i(\boldsymbol{y}^{[k]}) = 0, \quad i = 1,2,\cdots,l \tag{A10}$$

$$\boldsymbol{s}^{*\mathrm{T}}(\boldsymbol{\Lambda}\boldsymbol{y}^{[k]} - \boldsymbol{b}) = 0 \tag{A11}$$

$$\boldsymbol{H}\boldsymbol{y}^{[k]} \leqslant \boldsymbol{p} \tag{A12}$$

$$\boldsymbol{g}_i(\boldsymbol{y}^{[k]}) \leqslant 0, \quad i = 1,2,\cdots,l \tag{A13}$$

$$\boldsymbol{\iota}^* \geqslant 0, \quad \boldsymbol{v}^* \geqslant 0, \quad \boldsymbol{\Lambda}\boldsymbol{y}^{[k]} - \boldsymbol{b} \geqslant_K 0, \quad \boldsymbol{s}^* \in K^* \tag{A14}$$

The result of Eqs. (A8)–(89) proves that the Lagrange multipliers $(\boldsymbol{\iota}^*, \boldsymbol{v}^*, \boldsymbol{s}^*)$ and $\boldsymbol{y}^* = \boldsymbol{y}^{[k]}$ satisfy the KKT conditions for the problem in Eq. (A7). Specifically, Eqs. (A9)–(A10) are conditions for complementary slackness, Eqs. (A12)–(A14) are conditions for feasibility. A detailed proof can be found in Ref. 36.

Lemma A5. (Recursive Feasibility) If $P5^{[q-1]}$ has an optimal solution denoted as $\widetilde{X}^{[q-1]}(\cdot)$, then $P5^{[q]}$ has an optimal solution.

Proof. To establish the existence of an optimal solution for $P5^{[q]}$, it is sufficient to demonstrate that its feasible set is non-empty.[36] According to Lemma A3, any feasible solution to $P5^{[q]}$ is also feasible for Problem 2. Therefore, the feasible set of $P5^{[q]}$ is non-empty. Furthermore, Lemma A1 confirms that the feasible set of $P5^{[q]}$ is bounded. Consequently, $P5^{[q]}$ has an optimal solution.

Lemma A6. (Monotone Decreasing Property) The solution sequence $\left\{\widetilde{X}^{[q]}(\cdot)\right\}_{q=1,2,\cdots}$ generated by Algorithm 1 satisfies the monotone decreasing property $-z^{[q+1]}(t_{N+1}) \leqslant -z^{[q]}(t_{N+1})$. If $\widetilde{X}^{*}(\cdot)$ is the optimal solution, and $\widetilde{X}^{*}(\cdot) \neq \widetilde{X}^{[q]}(\cdot)$, it holds $-z^{*}(t_{N+1}) < -z^{[q]}(t_{N+1})$.

Proof. Assume that $\widetilde{X}^{*}(\cdot) = \widetilde{X}^{[q]}(\cdot)$. Because $\widetilde{X}^{[q]}(\cdot)$ is feasible to problem $P5^{[q]}$, this means that $-z^{[q]}(t_{N+1})$ is also optimal, which contradicts the uniqueness of the solution of $P5^{[q]}$. Therefore, only the strict inequality can hold.

Appendix B. Convergence discussion of Algorithm 2

Algorithm 2 is a combination of two optimization algorithms. The first optimization algorithm calculates the modified slope constraint under the reference trajectory that accounts for uncertainty. The second optimization algorithm computes the new trajectory under the modified slope constraint. Since uncertainty is trajectory-dependent, this algorithm is a recursive optimization algorithm. The characteristic of such algorithms is that the parameter settings in each iteration are adjusted based on the results of the previous optimization. When dealing with this type of iterative algorithm, convergence analysis becomes more complex, as each iteration depends not only on the current input but also on the previous results.

Due to the difficulty in quantitatively analyzing the convergence of Algorithm 2, providing a complete theoretical proof of its convergence remains an open challenge. Therefore, a qualitative analysis approach is adopted. The analysis primarily relies on Figs. 6 and 7.

First, it is important to note that uncertainty is bounded. When the uncertainty is as shown in Fig. 6, where the trajectory deviates from the slope constraint, a suitably relaxed modified glide-slope constraint is applied to reduce fuel consumption in the convex optimization problem, while reducing the gap between the uncertain trajectory and the slope. When the uncertainty is as shown in Fig. 7, where the uncertainty causes the trajectory to collide with the slope constraint, a stricter modified slope constraint is used to prevent the trajectory from colliding with the slope constraint. These two strategies ensure that the uncertain trajectory eventually aligns precisely with the slope constraint.

References

1. Wang KD, Wang YL, Dong BW, et al. Trajectory optimization of near-Earth asteroids exploration by using reusable probes from cislunar space. *Chin J Aeronaut* 2025;**38**(3):103234.
2. Wen TG, Zeng XY, Li ZW, et al. A comparative assessment of gravitational field modeling methods for binary asteroid landing. *Astrodynamics* 2024;**8**(3):417–35.
3. Zhao YJ, Yang HW, Li S, et al. On-board modeling of gravity fields of elongated asteroids using Hopfield neural networks. *Astrodynamics* 2023;**7**(1):101–14.
4. Chai JX, Gong YM, Mei J, et al. Modeling and trajectory tracking control for a novel multinode flexible small-body lander based on a port-Hamilton framework. *Space Sci Technol* 2024;**4**:113.
5. Lin YT, Yang W, Zhang H, et al. Return to the Moon: new perspectives on lunar exploration. *Sci Bull* 2024;**69**(13):2136–48.
6. Li CL, Zhang RQ, Yu DY, et al. China's Mars exploration mission and science investigation. *Space Sci Rev* 2021;**217**(4):57.
7. Hu RH, Huang XY, Xu C. Integrated visual navigation based on angles-only measurements for asteroid final landing phase. *Astrodynamics* 2023;**7**(1):69–82.
8. Baiocco P. Overview of reusable space systems with a look to technology aspects. *Acta Astronaut* 2021;**189**:10–25.
9. Huang XY, Xu C, Hu JC, et al. Powered-descent landing GNC system design and flight results for Tianwen-1 mission. *Astrodynamics* 2022;**6**(1):3–16.
10. Lu P, Callan R. Propellant-optimal powered descent guidance revisited. *J Guid Control Dyn* 2023;**46**(2):215–30.
11. Liu X, Li S, Xin M. Comparison of powered descent guidance laws for planetary pin-point landing. *Acta Astronaut* 2021;**187**:101–14.
12. Gong YM, Guo YN, Li DY, et al. Mars landing feedback guidance for non-concave trajectory construction. *Aerosp Sci Technol* 2023;**137**:108303.
13. Liu C, Yang HW, Li S, et al. Convex optimization of stochastic path-constrained trajectories near asteroids. *Aerosp Sci Technol* 2024;**153**:109463.
14. Gao DZ, Gong YM, Li CJ, et al. Adaptive pseudospectral successive convex optimization for six-degree-of-freedom powered descent guidance. *Aerosp Sci Technol* 2024;**155**:109544.
15. Zhang Y, Zeng TY, Guo YN, et al. Mars powered descent phase guidance law based on reinforcement learning for collision avoidance. *Int J Robust Nonlinear Control* 2023;**33**(17):10378–92.
16. Gong YM, Guo YN, Lyu YY, et al. Hybrid zero-effort-miss/zero-effort-velocity guidance for powered descent phase. *J Guid Control Dyn* 2024;**47**(5):1026–37.
17. Cui PY, Zhao DY, Zhu SY. Obstacle avoidance guidance for planetary landing using convex trajectory and adaptive curvature regulation. *Acta Astronaut* 2022;**199**:313–26.
18. Gong YM, Guo YN, Lyu YY, et al. Fast fixed-time three-dimensional terminal guidance with non-concave trajectory constraint. *Chin J Aeronaut* 2024;**37**(3):208–18.
19. Liu X, Li S, Xin M. Pseudospectral convex optimization based model predictive static programming for constrained guidance. *IEEE Trans Aerosp Electron Syst* 2023;**59**(3):2232–44.
20. Dong Y, Ding JZ, Wang CJ, et al. Soft landing stability analysis of a Mars lander under uncertain terrain. *Chin J Aeronaut* 2022;**35**(11):377–88.
21. Xiu Y, Zhu SY, Xu R, et al. Optimal crater landmark selection based on optical navigation performance factors for planetary landing. *Chin J Aeronaut* 2023;**36**(3):254–70.
22. Gao DZ, Gong YM, Guo YN, et al. Robust trajectory optimization for Mars lander powered descent considering navigation errors. *Sci Sin Phys Mech Astron* 2024;**55**(2):224513.

23. Chen X, Zhang R, Li H. Optimal feedback guidance with disturbance rejection for endoatmospheric powered descent. *Chin J Aeronaut* 2025;**38**(12):103336.
24. Wang FG, Yang SX, Xiong FF, et al. Robust trajectory optimization using polynomial chaos and convex optimization. *Aerosp Sci Technol* 2019;**92**:314–25.
25. Xiao YZ, Gong YM, Mei J, et al. Robust optimal powered descent guidance via model predictive convex programming. *Aerosp Sci Technol* 2025;**159**:109999.
26. Benedikter B, Zavoli A, Wang ZB, et al. Convex approach to covariance control for low-thrust trajectory optimization with mass uncertainty. Reston: AIAA; 2023. Report No.: AIAA-2023-2321.
27. Liu YZ, Wang MM, Luo JJ, et al. A tracking guidance method with funnel scheduling for Mars landing. *Aerosp Sci Technol* 2024;**155**:109738.
28. Cheng L, Wen H, Kang JJ, et al. Dynamic tube model predictive control for powered-descent guidance. *J Aerosp Eng* 2022;**35**(6):04022098.
29. Yang HW, Hu JC, Li S, et al. Reinforcement-learning-based robust guidance for asteroid approaching. *J Guid Control Dyn* 2024;**47**(10):2058–72.
30. Li WB, Song Y, Cheng L, et al. Closed-loop deep neural network optimal control algorithm and error analysis for powered landing under uncertainties. *Astrodynamics* 2023;**7**(2):211–28.
31. Chen XL, Zhang R, Zhang XY. Combined disturbance compensation guidance for powered descent in atmosphere. *Acta Aeronaut Astronaut Sin* 2023;**44**(23): 528465 [Chinese].
32. Zhang H, Li J, Wang Z, et al. Guidance navigation and control for Chang'E-5 powered descent. *Space Sci Technol* [Internet]. 2021 Jul [cited 2025 Jan 24]. Available from: https://spj.science.org/doi/10.34133/2021/9823609.
33. Bertsimas D, Brown DB, Caramanis C. Theory and applications of robust optimization. *SIAM Rev* 2011;**53**(3):464–501.
34. Acikmese B, Ploen SR. Convex programming approach to powered descent guidance for Mars landing. *J Guid Control Dyn* 2007;**30**(5):1353–66.
35. Blackmore L, Açikmeşe B, Scharf DP. Minimum-landing-error powered-descent guidance for Mars landing using convex optimization. *J Guid Control Dyn* 2010;**33**(4):1161–71.
36. Liu XF, Lu P. Solving nonconvex optimal control problems by convex optimization. *J Guid Control Dyn* 2014;**37**(3):750–65.
37. Liu XF. Convergence-guaranteed trajectory planning for a class of nonlinear systems with nonconvex state constraints. *IEEE Trans Aerosp Electron Syst* 2022;**58**(3):2243–56.
38. Zhang P, Li WB, Gong SP. Ascent trajectory optimization for boost-glide vehicle using homotopy approximation function sequential convex programming. *IEEE Trans Aerosp Electron Syst* 2025;**61**(3):7576–96.
39. Wang ZB, Grant MJ. Constrained trajectory optimization for planetary entry *via* sequential convex programming. *J Guid Control Dyn* 2017;**40**(10):2603–15.
40. Berkovitz LD. *Optimal control theory*. Berlin: Springer-Verlag; 1975. p. 61.