Hindawi
International Journal of Aerospace Engineering
Volume 2022, Article ID 9104823, 11 pages
https://doi.org/10.1155/2022/9104823

Research Article

Fault-Tolerant Guidance of Rocket Vertical Landing Phase Based on MPC Framework

Jingqi Li,[1] **Yaosong Long**,[2] **Mao Su**,[3] **Lei Liu**,[1] **Bo Wang**,[1] **and Zhongtao Cheng**[1]

[1]*National Key Laboratory of Science and Technology on Multispectral Information Processing, School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan 430074, China*
[2]*School of Aerospace Engineering, Huazhong University of Science and Technology, Wuhan 430074, China*
[3]*Designing Institute of Hubei Space Technology Academy, Wuhan 430074, China*

Correspondence should be addressed to Zhongtao Cheng; ztcheng@hust.edu.cn

Received 30 March 2022; Revised 21 June 2022; Accepted 6 July 2022; Published 12 August 2022

Academic Editor: Lingxia Mu

For the vertical landing process of reusable rockets, the landing accuracy is likely to be affected by disturbances and faults during flight. In this paper, a fault-tolerant guidance method based on the MPC framework is put forward. First, we propose a piecewise guidance algorithm that combines a trajectory optimization algorithm based on convex optimization with the MPC framework. With the fast trajectory optimization algorithm and the MPC framework that recursively introduces the real-time state, this algorithm forms a robust closed loop. Then, we design an integrated guidance, navigation, and control (GNC) system to enhance the fault tolerance and robustness of the guidance method. Simulation experiments verify that this method is fault-tolerant to various fault conditions including navigation system failures, control system failures, drag coefficient deviations, and atmospheric density deviations. This guidance method is robust enough to overcome disturbances and faults, and it has great potential for online use.

1. Introduction

Launching a rocket is a high-cost and high-risk investment, and there is growing concern about how to make it economical. Building reusable rockets is an effective way to reduce costs and create business value. At present, many countries have invested plenty of research in rocket recovery technology. Two US companies, Blue Origin and SpaceX, have successfully conducted rocket recovery experiments, proving the feasibility of vertical landing technology. The rocket recovery mission is generally divided into four flight phases: the attitude adjustment phase, the power deceleration phase, the aerodynamic deceleration phase, and the vertical landing phase [1], so as to ensure that the rocket lands vertically and softly at the predetermined location. The recovery process is confronted with difficulties including large spans of airspace and velocity domain, large changes in the flight environment, complex flight constraints, and strong disturbances and uncertainties, any of which may lead to errors. The accumulated errors generated by preceding phases must be eliminated in the vertical landing phase which is the last phase of the recovery mission [2]. The time is short but the precision is high. Therefore, extremely high requirements are placed on the guidance speed and accuracy of this phase. Research on fast and accurate guidance method of rocket vertical landing phase is the basis for successfully recovering a rocket.

The study of the landing problem began with the Apollo project. Due to the limited computing ability, researchers could only apply analytical guidance law in that era [3]. Unlike the lunar landing mission, more complex path constraints and rigorous terminal conditions must be considered for rocket vertical landing problems. However, the derivation of the analytical method is cumbersome, making it unsuitable for most complex nonlinear problems [4]. Thanks to the research on numerical methods by

mathematicians and the development of computers' computing ability [2], the trajectory optimization of rocket vertical landing problems mainly relies on numerical algorithms these days.

The vertical landing problem can be expressed as an optimal control problem with the objective of minimum fuel consumption, subject to the constraints of dynamic equations and the constraints on state and control variables. It can be transformed into a nonlinear programming problem (NLP) through time discretization. Early studies found that the pseudospectral methods had high accuracy in solving NLP. In recent years, these methods have been widely used for trajectory optimization problems. Chen and Xia [5] used a pseudospectral method to analyze the ascent trajectory characteristics of a solid rocket-powered launch vehicle. Mu et al. [6] planned the Mars landing trajectory by the Gauss pseudospectral method. However, the real-time performance and convergence of these methods cannot meet the requirements of online guidance [2]. Subsequent studies have shown that convex optimization is more advantageous for online trajectory optimization. As long as the problem is transformed into a convex optimization form, it can theoretically be solved in polynomial time by the interior point method [7]. Convex optimization was first applied to Mars landing [8, 9]. In order to convert the landing problem into a convex problem, Açıkmeşe et al. [10] proposed a lossless convexification method to deal with thrust constraints and proved that the lossless convexification problem is equivalent to the original problem, which greatly promoted the development of convex optimization methods in the field of trajectory optimization. However, not all constraints can be convexified by lossless convexification. A more general approach is successive convexification, which linearizes the original nonconvex problem into a series of convex subproblems and then iteratively finds the optimal solution of the subproblems. In 2013, Lu and Liu [11] proposed successive convexification for convexifing dynamic equations and other nonconvex constraints when they studied the rendezvous problem. They later used this method to solve the hypersonic vehicle reentry problem [12] and the rocket landing problem [13]. The advantages of convex optimization including computational efficiency and insensitivity to initial guesses make it an efficient method for solving trajectory optimization problems. It is widely used in the aerospace field, such as low-thrust transfers [14], rendezvous problem [15], re-entry problem [12], and the vertical landing problem studied in this paper.

There are two commonly used coordinate systems for modeling rocket vertical landing problems, the velocity coordinate system [16–20] and the landing point coordinate system [21–25]. In this paper, the three-degree-of-freedom dynamics of the rocket are derived based on the landing point coordinate system. Since Szmuk et al. [21] applied successive convex optimization to solve the rocket vertical landing problem in 2016, research on trajectory optimization of landing problems has become more and more abundant and mature. The current research focus is to improve the convergence performance and solution efficiency of successive convex optimization, so that the approach can meet the needs of online guidance. In Ref. [24], a convex feasible set (CFS) method is proposed to convexify the angle of attack constraint which is a nonconvex-nonconcave inequality. Ref. [25] presents a two-stage successive convexification method. Simulation experiments show that the performance of the two-stage method is more stable and efficient than the single-stage method. Ref. [26] compares the computational performance and solution accuracy of six discretization methods. In Ref. [18], an online update strategy for trust regions is proposed to speed up the convergence of successive convex optimization.

However, the rocket will be affected by wind interference and environmental uncertainties after entering the atmosphere, and modules such as the navigation system and the control system may malfunction during the landing process. Trajectory optimization alone cannot overcome these disturbances and faults. Therefore, it is necessary to design an online fault-tolerant guidance method to ensure landing accuracy. A fault-tolerant control method has been proposed to deal with parametric uncertainties and unknown actuator failures [27]. But there is a lack of research on guidance methods for vertical landing problems at present. In Ref. [22], a receding horizon guidance method based on convex optimization is proposed. Ref. [28] and Ref. [29] both propose to construct a guidance, navigation, and control (GNC) system for closed-loop guidance, but they do not give a specific algorithm.

In recent years, some researchers put forward a guidance method based on trajectory optimization and model predictive control (MPC) framework. MPC is a control strategy that recursively solves an optimal control problem with updated system states at each sampling time. Ref. [30] presents the MPC algorithm for the optimal guidance and reconfiguration of swarms of spacecraft. In Ref. [31], MPC is used for asteroid landing. Ref. [32] reviews applications of MPC in the aerospace guidance field. For the online guidance problem of the rocket vertical landing phase, a successive convexification + MPC guidance algorithm is proposed by Ref. [33]. Ref. [19] embeds a pseudospectral-improved successive convexification (PISC) algorithm in the MPC framework to construct a parallel feasibility-guaranteed guidance algorithm. Ref. [34] designs an antidelay model predictive control (AD-MPC) scheme for carrier landing. Ref. [35] implements a successive convexification MPC-based guidance algorithm to solve the six-degree-of-freedom powered descent guidance problem.

In this paper, we propose a piecewise guidance algorithm that embeds a convex optimization-based trajectory optimization algorithm in the MPC framework. An integrated GNC system is then designed to further improve the fault tolerance and robustness of the entire system. The rest of this paper is organized as follows. In Section 2, a mathematical description of the rocket vertical landing problem is given. Section 3 elaborates the online trajectory optimization algorithm based on convex optimization, including convexification and discretization methods for transforming the original problem into a second-order cone programming (SOCP) problem. In Section 4, the trajectory optimization algorithm is embedded in the MPC framework, and a

piecewise guidance method is proposed to cope with the bang-bang characteristic of the control. Section 5 designs an integrated GNC system. In Section 6, simulation experiments are carried out to verify the reliability, accuracy, robustness, and fault tolerance of the guidance method. Section 7 concludes the whole work.

2. Problem Formulation

In this section, we use the inertial reference frame to formulate the fuel-optimal rocket vertical landing problem as a standard optimal control problem.

2.1. Dynamics and Constraints. The rocket's flight time during the vertical landing phase is so short that the Earth's surface can be assumed to be a horizontal plane. Based on this assumption, we build an inertial reference frame with the origin O located at the predetermined landing point as shown in Figure 1. The OX axis is perpendicular to the plane, and the upward direction is positive. The OY axis is parallel to the plane, and the direction pointing to the rocket's launch point is positive. The OZ axis and the other two axes form a right-handed Cartesian coordinate system.

In this reference frame, the rocket's three-degree-of-freedom dynamics are

$$\begin{cases} \dot{\boldsymbol{r}}(t) = \boldsymbol{v}(t), \\ \dot{\boldsymbol{v}}(t) = \dfrac{\boldsymbol{T}(t) + \boldsymbol{D}(t)}{m(t)} + \boldsymbol{g}, \\ \dot{m}(t) = -\dfrac{\|\boldsymbol{T}(t)\|}{I_{sp} g_0}, \end{cases} \tag{1}$$

where $\boldsymbol{r}(t)$, $\boldsymbol{v}(t)$, and $m(t)$ represent the position vector, velocity vector, and mass of the rocket, respectively. $\boldsymbol{T}(t)$ is the thrust vector of the rocket engine. $\|\cdot\|$ represents the 2-norm of the vector.

During the landing phase, the constraints on thrust magnitude and direction are expressed as

$$T_{\min} \le \|\boldsymbol{T}(t)\| \le T_{\max}, \tag{2}$$

$$\|\boldsymbol{T}(t)\| \cos \eta_{\max} \le \mathbf{e}_x \boldsymbol{T}(t), \tag{3}$$

where $T_{\min}$ and $T_{\max}$ represent the minimum and maximum thrust magnitude that the engine can provide, respectively. $\eta_{\max}$ is the maximum allowable value of the angle between the thrust direction and the OX direction. $\boldsymbol{D}(t)$ is the aerodynamic drag calculated by the following formula:

$$\boldsymbol{D}(t) = -\frac{1}{2} \rho S_D C_D \|\boldsymbol{v}(t)\| \boldsymbol{v}(t), \tag{4}$$

where ρ is the air density, S_D is the drag reference area, and C_D is the drag coefficient.

To prevent the rocket from colliding with the ground or being interfered with by the ground protrusion during flight, a glide-slope constraint is imposed to restrict the path of the rocket to lie within an upward-facing cone:

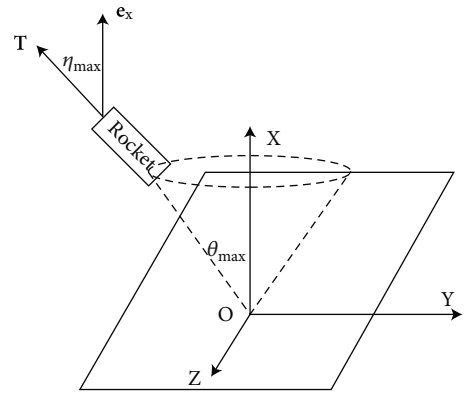

Figure 1: Inertial reference frame.

$$\|\boldsymbol{r}(t)\| \cos \theta_{\max} \le \boldsymbol{e}_x \boldsymbol{r}(t), \tag{5}$$

where $\mathbf{e}_x$ represents the unit vector in the OX direction. $\theta_{\max}$ is the maximum allowable half-cone angle.

We also need to consider boundary conditions. The rocket's initial position vector, velocity vector, and mass are specified as fixed parameters. Its final position and velocity vectors are fixed, too. The thrust vector at the terminal time must be in the OX direction. The fuel remaining of the rocket must be nonnegative, which means that the landing mass of the rocket must be greater than the dry mass which is denoted as m_{dry}. The boundary conditions are summarized as

$$\boldsymbol{r}(0) = \boldsymbol{r}_0, \boldsymbol{v}(0) = \boldsymbol{v}_0, m(0) = m_0, \tag{6}$$

$$\boldsymbol{r}(t_f) = 0, \boldsymbol{v}(t_f) = 0, \boldsymbol{T}(t_f) = \|\boldsymbol{T}(t_f)\| e_x, m(t_f) \ge m_{\text{dry}}. \tag{7}$$

2.2. Performance Index. The performance index of the fuel-optimal rocket vertical landing problem is selected as minimizing the fuel consumption, which is equivalent to maximizing the terminal mass of the rocket. Therefore, the objective function is expressed as

$$J = -m(t_f). \tag{8}$$

To sum up, with thrust vector $\boldsymbol{T}$ as the control variable and $[\boldsymbol{r}^T, \boldsymbol{v}^T, m]^T$ as the state variables, the rocket vertical landing problem can be formulated as an optimal control problem with free terminal time:

$$\begin{aligned} &\text{Problem0} : \min_{\boldsymbol{T}} J = -m(t_f), \\ &\text{subject to } (1)(2)(3)(5)(6)(7). \end{aligned} \tag{9}$$

3. Trajectory Optimization Algorithm Based on Convex Optimization

In this section, we will elaborate on the trajectory optimization algorithm based on convex optimization and embedded in the subsequent guidance system. This includes the use of convexification and discretization methods to convert Problem0 into a convex problem, as well as iteratively solving the converted problem afterward.

3.1. Convexification. The thrust magnitude constraint in Problem0 is nonconvex. Its nonconvex feasible region can be relaxed through lossless convexification to form a high-dimensional convex feasible region, transforming the nonconvex constraint into a relaxed convex constraint. Using the above method, we introduce a slack variable Γ to transform Eq. (2) into

$$\|\boldsymbol{T}(t)\| \le \Gamma(t), \tag{10}$$

$$T_{\min} \le \Gamma(t) \le T_{\max}. \tag{11}$$

By replacing the nonconvex constraint (Eq. (2)) in Problem0 with the convex constraints (Eqs. (10) and (11)), the nonconvex problem is converted into a relaxed problem. Studies have shown that the optimal solution to the relaxed problem is also the optimal solution to Problem0 [8–10]. Taking $\boldsymbol{T}$ and Γ as control variables, that is, the control variables are redefined as $\boldsymbol{u} = [\Gamma, \boldsymbol{T}^T]^T$. The dynamics are transformed into

$$\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) = \begin{bmatrix} \dot{\boldsymbol{r}}(t) = \boldsymbol{v}(t) \\ \dot{\boldsymbol{v}}(t) = \dfrac{\boldsymbol{T}(t) + \boldsymbol{D}(t)}{m(t)} + \boldsymbol{g} \\ \dot{m}(t) = \dfrac{-\Gamma}{(I_{sp} g_0)} \end{bmatrix}. \tag{12}$$

The nonlinearity of Eq. (12) is caused by the free terminal time, aerodynamic drag $\boldsymbol{D}(t)$, and the denominator $m(t)$. It can be linearized through successive convexification. In this paper, the terminal time is added to the optimization variables, which will be optimized together with the state and control variables in the subsequent optimization process. t_f in Eq. (12) is a hidden variable. Define $\tau \triangleq t/t_f$. Apply the chain rule to make t_f explicit:

$$\frac{d\boldsymbol{x}}{dt} = \frac{d\boldsymbol{x}}{d\tau}\frac{d\tau}{dt}. \tag{13}$$

Eq. (12) is transformed into

$$\frac{d\boldsymbol{x}}{d\tau} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) t_f. \tag{14}$$

We adopt successive convexification to iteratively solve Eq. (14). The first-order Taylor expansion is performed at the kth iteration to transform Eq. (14) into

$$\begin{aligned} \frac{d\boldsymbol{x}}{d\tau} = {} & \boldsymbol{f}\left(\boldsymbol{x}^k, \boldsymbol{u}^k\right) t_f^k + \boldsymbol{A}\left(\boldsymbol{x}^k, \boldsymbol{u}^k\right)\left(\boldsymbol{x} - \boldsymbol{x}^k\right) t_f^k \\ & + \boldsymbol{B}\left(\boldsymbol{x}^k, \boldsymbol{u}^k\right)\left(\boldsymbol{u} - \boldsymbol{u}^k\right) t_f^k + \boldsymbol{f}\left(\boldsymbol{x}^k, \boldsymbol{u}^k\right)\left(t_f - t_f^k\right), \end{aligned} \tag{15}$$

where $\boldsymbol{A}(\boldsymbol{x}^k, \boldsymbol{u}^k)$ and $\boldsymbol{B}(\boldsymbol{x}^k, \boldsymbol{u}^k)$ are the gradients of $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})$ with respect to the state variables $\boldsymbol{x} = [\boldsymbol{r}^T, \boldsymbol{v}^T, m]^T$ and control variables $\boldsymbol{u} = [\Gamma, \boldsymbol{T}^T]^T$, respectively.

3.2. Discretization. The problem is still infinite-dimensional after convexification and needs to be discretized into a finite-dimensional problem. We adopt the trapezoidal discretization to discretize the problem. This requires the time of flight to be evenly divided into N discrete intervals, i.e., $N + 1$ discrete points (the value of N in each guidance cycle will be discussed later). The subscript $i (i = 1, \cdots, N + 1)$ represents the ith discrete point of the state or control variable, then, Eq. (15) is discretized as

$$\begin{aligned} \boldsymbol{x}_{i+1} = {} & \boldsymbol{x}_i + \frac{1}{2N}\left(\boldsymbol{A}_i^k\left(\boldsymbol{x}_i - \boldsymbol{x}_i^k\right) t_f^k + \boldsymbol{B}_i^k\left(\boldsymbol{u}_i - \boldsymbol{u}_i^k\right) t_f^k + \boldsymbol{f}_i^k t_f\right) \\ & + \frac{1}{2N}\left(\boldsymbol{A}_{i+1}^k\left(\boldsymbol{x}_{i+1} - \boldsymbol{x}_{i+1}^k\right) t_f^k + \boldsymbol{B}_{i+1}^k\left(\boldsymbol{u}_{i+1} - \boldsymbol{u}_{i+1}^k\right) t_f^k + \boldsymbol{f}_{i+1}^k t_f\right), \end{aligned} \tag{16}$$

where $\boldsymbol{A}_i^k = \boldsymbol{A}(\boldsymbol{x}_i^k, \boldsymbol{u}_i^k)$, $\boldsymbol{B}_i^k = \boldsymbol{B}(\boldsymbol{x}_i^k, \boldsymbol{u}_i^k)$, and $\boldsymbol{f}_i^k = \boldsymbol{f}(\boldsymbol{x}_i^k, \boldsymbol{u}_i^k)$.

After discretization, constraints (3), (5), (6), (7), (10), and (11) are transformed into

$$\|\boldsymbol{T}_i\| \cos \eta_{\max} \le \boldsymbol{e}_x \boldsymbol{T}_i, \tag{17}$$

$$\|\boldsymbol{r}_i\| \cos \theta_{\max} \le \boldsymbol{e}_x \boldsymbol{r}_i, \tag{18}$$

$$\boldsymbol{r}_1 = \boldsymbol{r}_0, \boldsymbol{v}_1 = \boldsymbol{v}_0, m_1 = m_0, \tag{19}$$

$$\boldsymbol{r}_{N+1} = 0, \boldsymbol{v}_{N+1} = 0, \boldsymbol{T}_{N+1} = \|\boldsymbol{T}_{N+1}\| \boldsymbol{e}_x, m_{N+1} \ge m_{\text{dry}}, \tag{20}$$

$$\|\boldsymbol{T}_i\| \le \Gamma_i, \tag{21}$$

$$T_{\min} \le \Gamma_i \le T_{\max}. \tag{22}$$

3.3. Trajectory Optimization Algorithm. Define $H_i = I + t_f^k/(2N)\boldsymbol{A}_i^k$, $H_{i+1} = -I + t_f^k/(2N)\boldsymbol{A}_{i+1}^k$, $G_i = t_f^k/(2N)\boldsymbol{B}_i^k$, $G_{i+1} = t_f^k/(2N)\boldsymbol{B}_{i+1}^k$, $F_i = (\boldsymbol{f}_i^k + \boldsymbol{f}_{i+1}^k)/(2N)$, $C_i = \boldsymbol{A}_i^k \boldsymbol{x}_i^k t_f^k + \boldsymbol{B}_i^k \boldsymbol{u}_i^k t_f^k$. Equation (16) can be expressed as

$$H_i \boldsymbol{x}_i + H_{i+1} \boldsymbol{x}_{i+1} + G_i \boldsymbol{u}_i + G_{i+1} \boldsymbol{u}_{i+1} + F_i = \frac{1}{2N}(C_i + C_{i+1}). \tag{23}$$

The terminal time t_f is added to the control variables for optimization. Combine the discretized forms of state variables $\boldsymbol{x} = [\boldsymbol{r}^T, \boldsymbol{v}^T, m]^T$, control variables $\boldsymbol{u} = [\Gamma, \boldsymbol{T}^T]^T$ and t_f into a joint optimization variable Z, that is, define $Z = [\boldsymbol{x}_1^T, \cdots, \boldsymbol{x}_{N+1}^T, \boldsymbol{u}_1^T, \cdots, \boldsymbol{u}_{N+1}^T, t_f]^T$, then, Eq. (23) can be

transformed into the following matrix form:

$$MZ = C, \tag{24}$$

where

$$M = \begin{bmatrix} I & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ H_1 & H_2 & \cdots & 0 & 0 & G_1 & G_2 & \cdots & 0 & 0 & F_1 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & H_N & H_{N+1} & 0 & 0 & \cdots & G_N & G_{N+1} & F_N \end{bmatrix},$$

$$C = 1/(2N) \begin{bmatrix} 2N\boldsymbol{x}_1 \\ C_1 + C_2 \\ \vdots \\ C_N + C_{N+1} \end{bmatrix}.$$

Problem0 is transformed into

$$\begin{aligned} &\text{Problem1} : \min_{\Gamma, \boldsymbol{T}, t_f} J = -m_{N+1}, \\ &\text{subject to } (24)\ (17)(18)(19)(20)(21)(22). \end{aligned} \tag{25}$$

Problem1 is an SOCP problem that can be solved iteratively using the interior-point algorithm. The solution steps are shown in Figure 2.

The convergence condition in Figure 2 is

$$\max_i \left| \boldsymbol{x}_i^{k+1} - \boldsymbol{x}_i^k \right| \le \varepsilon_{\boldsymbol{x}}, \quad \max_i \left| \boldsymbol{u}_i^{k+1} - \boldsymbol{u}_i^k \right| \le \varepsilon_u, \tag{26}$$

which means that if the difference between the values of the optimization variables of two consecutive iterations is within the specified tolerance range, the iteration should be stopped and the optimal solution is obtained.

4. Piecewise Guidance Algorithm Based on MPC Framework

In this section, on the basis of the trajectory optimization algorithm based on convex optimization detailed in the previous section, a piecewise guidance method combining the trajectory optimization algorithm with the MPC framework is proposed. At each sampling time, the real-time state of the rocket fed back by the navigation system is employed as the initial state to start the trajectory optimization algorithm. Then, the optimal values of a series of control variables from the current time to the landing point are obtained. But only the values in the current guidance cycle are applied to control the rocket until the next sampling time. Repeat these steps until the rocket lands at the predetermined point. Since the trajectory optimization algorithm needs to be provided with an initial guess, the optimization result obtained at the previous sampling time can be used as the initial trajectory at the current sampling time. The computational efficiency of the trajectory optimization algorithm based on convex optimization makes it possible for the guidance algorithm to form an effective closed loop. In addition, the practice of recursively introducing the real-time state of the rocket into the trajectory optimization ensures the stability, robustness, and fault tolerance of the closed-loop.

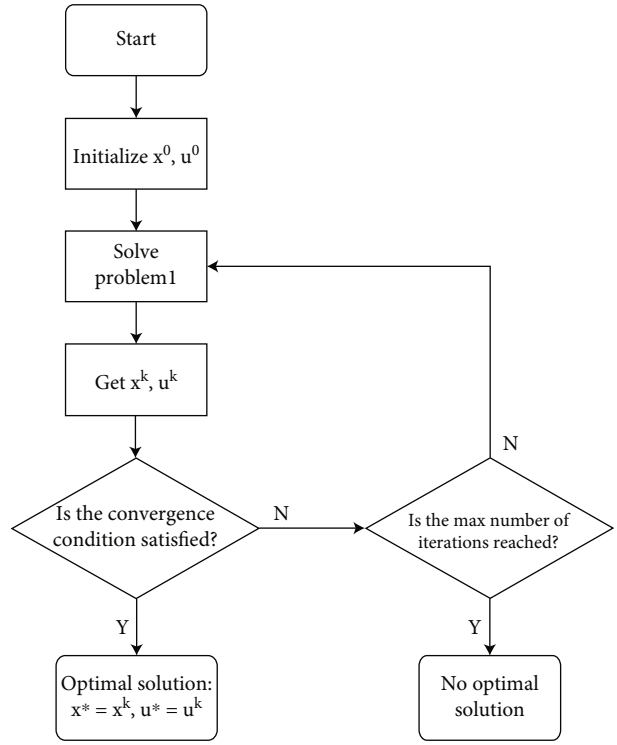

Figure 2: Solution steps for the trajectory optimization problem.

The piecewise method is to cope with the thrust magnitude's bang-bang characteristic. Bang-bang control means the control amount changes dramatically in a short period near the switching point. If the duration of each guidance cycle is too long and a fault occurs near the switching point, the trajectory will deviate from the optimal solution.

The piecewise guidance algorithm based on MPC framework is shown in algorithm 1, where t_i represents the sampling time of the ith trajectory optimization, $\mathscr{T}_i$ represents the terminal time obtained by the ith trajectory optimization, $\boldsymbol{T}$ represents the thrust series from the current time to the landing point, while $\boldsymbol{T}_i$ contains only the portion within the ith guidance cycle, Δt is the duration of each guidance cycle which is set initially, and Δt_i is the actual duration of the ith guidance cycle, $\Delta t_i = t_{i+1} - t_i$.

Over time, the time horizon of the optimization becomes shorter, and the number of discrete points for trajectory optimization within the guidance cycle should also be reduced accordingly. The number of discrete points taken by the ith trajectory optimization algorithm is determined by

$$N_i = \left\lceil N_{i-1} \frac{\mathscr{T}_{i-1} - t_i}{\mathscr{T}_{i-1} - t_{i-1}} \right\rceil, \tag{27}$$

where the ceiling function is defined as $\lceil x \rceil = \min \{n \in \mathbb{Z} | n \ge x\}$. $(\mathscr{T}_{i-1} - t_{i-1})$is the time horizon of the $(i-1)$th

Piecewise guidance algorithm based on MPC framework
Initialization: Save the initial trajectory to the database, set relevant parameters required by the trajectory optimization algorithm, and set the update index $i = 1$.
while$(\mathscr{T}_{i-1} - t_{i-1}) > 0$
  generate guidance commands according to the guidance cycle clock;
  sample the current state $\boldsymbol{x}(t_i)$(i.e. $\boldsymbol{r}(t_i), \boldsymbol{v}(t_i), m(t_i)$) of the rocket;
  employ the optimal trajectory $\boldsymbol{J}_{i-1}$ which was obtained in the previous guidance cycle as the initial guess. Start the trajectory optimization using the current state $\boldsymbol{x}(t_i)$. Obtain a new optimal trajectory $\boldsymbol{J}_i$ and an optimal thrust series $\boldsymbol{T}$. Save $\boldsymbol{J}_i$ to the database, implement $\boldsymbol{T}_i$, and set $i = i + 1$;
  if a fault occurs and t_i approaches the switching point
    shorten the duration of the guidance cycle: $\Delta t_i = \alpha \cdot \Delta t \ (0 < \alpha < 1)$;
  else
    $\Delta t_i = \Delta t$;
  end if
end while

Algorithm 1: Piecewise guidance algorithm based on MPC framework.

optimization, and $(\mathscr{T}_{i-1} - t_i)$ is the estimate of the time horizon of the ith optimization.

5. Integrated GNC System Design

In actual flight, a key technology to realize fault-tolerant guidance is designing a guidance, navigation, and control (GNC) system. The whole GNC system is closed-loop, where the navigation system consists of a variety of sensors for real-time measurement and evaluation of states including the rocket's position, velocity, and mass, the guidance system consists of the guidance algorithm proposed in the previous section to solve the landing trajectory, and the control system consists of the actuators to control the rocket to track the trajectory. The guidance system needs to provide the real-time state of the rocket to the guidance system before the trajectory optimization. And the control commands derived from the guidance algorithm need to be executed by the control system. Thanks to the division of labor and rapid cooperation of each system, the integrated GNC system has strong fault tolerance and robustness against various disturbances and faults including navigation system failures, control system failures, drag coefficient deviations, and atmospheric density deviations.

In this section, an integrated GNC system is built, as shown in Figure 3. In a guidance cycle, based on the current state $\boldsymbol{r}(t_i), \boldsymbol{v}(t_i), m(t_i)$ fed back by the navigation system, the guidance system adopts the trajectory optimization algorithm based on convex optimization to generate an optimal thrust series $\boldsymbol{T}$ (where $\boldsymbol{J}_{i-1}$ represents the initial guess which is the optimal trajectory of the last guidance). The control system outputs the thrust command $\boldsymbol{T}_i$ of the current guidance cycle and transmits it to the rocket, and the rocket system executes the command accordingly. In the simulation experiments, we use numerical integration to integrate the state of the rocket after one guidance cycle, which is used to simulate the real-time state of the rocket. Yet in actual flight, the state is measured by the navigation system and transmitted to the guidance system. Here comes the next guidance cycle, repeat the above until the rocket lands. The piecewise guidance algorithm proposed in the previous section and the integrated GNC system designed in this section ensure the fault tolerance of the guidance method.

6. Simulation Experiments

In this section, numerical simulations are provided to verify the reliability, accuracy, fault tolerance, and robustness of the proposed guidance method. All simulation experiments are carried out on MATLAB with the use of the modeling tool CVX [36, 37] to establish the guidance problem and the solver MOSEK [38] to solve it. We employ the fourth-order Runge-Kutta integration with a 0.01 s time step to obtain updated states at each sampling time. The parameters used in the simulations are shown in Table 1.

The number of discrete points for the first trajectory optimization is: $N = 40$. Assuming r_i^0, v_i^0, and m_i^0 vary linearly from the initial value to the terminal value, then, the initial guesses for all optimization parameters are

$$\begin{cases} \boldsymbol{r}_i^0 = \left(\boldsymbol{r}_f - \boldsymbol{r}_0\right)\dfrac{(i-1)}{N} + \boldsymbol{r}_0, \\ \boldsymbol{v}_i^0 = \left(\boldsymbol{v}_f - \boldsymbol{v}_0\right)\dfrac{(i-1)}{N} + \boldsymbol{v}_0, \\ m_i^0 = \left(m_{\text{dry}} - m_0\right)\dfrac{(i-1)}{N} + m_0, \\ \Gamma_i^0 = T_{\min}, \boldsymbol{T}_i^0 = \dfrac{-\Gamma_i^0 \boldsymbol{v}_i^0}{\left\|\boldsymbol{v}_i^0\right\|}, \\ t_f^0 = 40s. \end{cases} \tag{28}$$

The convergence condition is

$$\varepsilon_{\boldsymbol{r}} = 10^{-3}m, \varepsilon_{\boldsymbol{v}} = 10^{-3}m/s, \varepsilon_m = 10^{-3}kg, \varepsilon_\Gamma = 10^{-3}kN. \tag{29}$$

6.1. Reliability and Accuracy. In this subsection, we conduct the simulations in the absence of any faults and deviations. Figures 4 and 5 show the trajectories and control commands

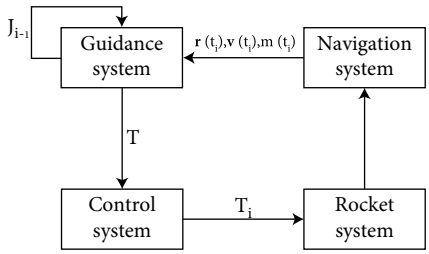

Figure 3: Integrated GNC system.

Table 1: Parameter values.

Parameter	Value	Units
$\boldsymbol{r}_0$	$[3500, 700, 0]^T$	m
$\boldsymbol{v}_0$	$[-200,-90, 0]^T$	m/s
m_0	27000	kg
m_{dry}	20000	kg
$T_{\min}$	100	kN
$T_{\max}$	300	kN
$\theta_{\max}$	80	°
$\eta_{\max}$	15	°
g_0	9.8	m/s2
I_{sp}	270	s
ρ	1.225	kg/m3
S_D	10	m2
C_D	2.2	—
Δt	1	s

of the rocket vertical landing phase obtained by the trajectory optimization algorithm and the guidance method presented in this paper, respectively. The red curves represent the results of the trajectory optimization algorithm, and the black is the guidance method. The differences between the results are small, demonstrating the reliability of the guidance method. We then perform numerical integration using the control commands to obtain the actual terminal states. Tables 2 and 3 show the terminal states and landing errors, respectively. It can be seen that the landing position and velocity errors of the guidance method are smaller than those of the trajectory optimization algorithm, which proves that the guidance method proposed in this paper has higher accuracy.

6.2. Fault Tolerance and Robustness. In this subsection, we assume three scenarios: faults occur in the navigation system, faults occur in the control system, and estimates of parameters (the drag coefficient and the atmospheric density) have deviations. Considering these scenarios, we apply the trajectory optimization algorithm and the guidance method proposed in this paper to conduct simulation experiments, respectively, and compare their results.

6.2.1. Faults Occur in the Navigation System. It is assumed that the navigation system has faults. Rather, the measurement of the real-time state of the rocket has the following errors:

$$\begin{cases} \|\boldsymbol{r}_{\text{measure}}\| = \|\boldsymbol{r}_{\text{true}}\| \pm \min\left(10, \|\boldsymbol{r}_{\text{true}}\| \times \varepsilon\%\right) \times \text{random}(0, 1), \\ \|\boldsymbol{v}_{\text{measure}}\| = \|\boldsymbol{v}_{\text{true}}\| \pm \min\left(1, \|\boldsymbol{v}_{\text{true}}\| \times \varepsilon\%\right) \times \text{random}(0, 1), \end{cases} \tag{30}$$

where 10 and 1 represent the absolute errors of the rocket's position and velocity, ε represents the maximum range of the relative error, $\text{random}(0, 1)$ represents a random number generated from [0,1], $\boldsymbol{r}_{\text{true}}, \boldsymbol{v}_{\text{true}}$ represents the actual state of the rocket, and $\boldsymbol{r}_{\text{measure}}, \boldsymbol{v}_{\text{measure}}$ represents the state measured by the navigation system and fed back to the guidance system. We take the value of ε as 5, 10, and 15 for experiments, and the landing errors under these fault conditions are shown in Table 4.

It can be seen from Table 4 that as the relative error of the measured state increases, the landing errors of the rocket also increase, but they remain within a small range. This experiment proves that the guidance method proposed in this paper is fault-tolerant and robust to navigation system faults.

6.2.2. Faults Occur in the Control System. It is assumed that the control system has two fault conditions: thrust cannot change continuously (that is, thrust is constant during each guidance cycle), and thrust magnitude has deviations. The trajectory optimization algorithm and the guidance method proposed in this paper are, respectively, applied to conduct simulations under the above two fault conditions, and their results are compared.

(1) Thrust Cannot Change Continuously. For trajectory optimization, we assume that thrust is constant per second, and for guidance, thrust is constant during each guidance cycle. The landing errors and landing masses are shown in Table 5.

(2) Thrust Magnitude Has Deviations. It is assumed that there are deviations within ±5% of the thrust magnitude. In the experiment, random deviations within ±5% are applied to the thrust magnitude obtained by the trajectory optimization algorithm and the guidance method. Then, the deviated thrust is used to control the rocket. The landing errors and landing masses are shown in Table 6.

It can be seen from Tables 5 and 6 that when the thrust cannot be continuously changed or the thrust magnitude has deviations, the landing position error and velocity error obtained by trajectory optimization are significantly larger than those obtained by guidance. The landing position errors of the guidance method under the two fault conditions remain at the same order of magnitude as that under the no-fault condition in Subsection 6.1, which demonstrates that the two fault conditions have little effect on the landing position. As for the landing velocity error, the guidance method can make it at the same order of magnitude as the no-fault condition when the thrust magnitude has deviations within ±5%. The error is one order of magnitude

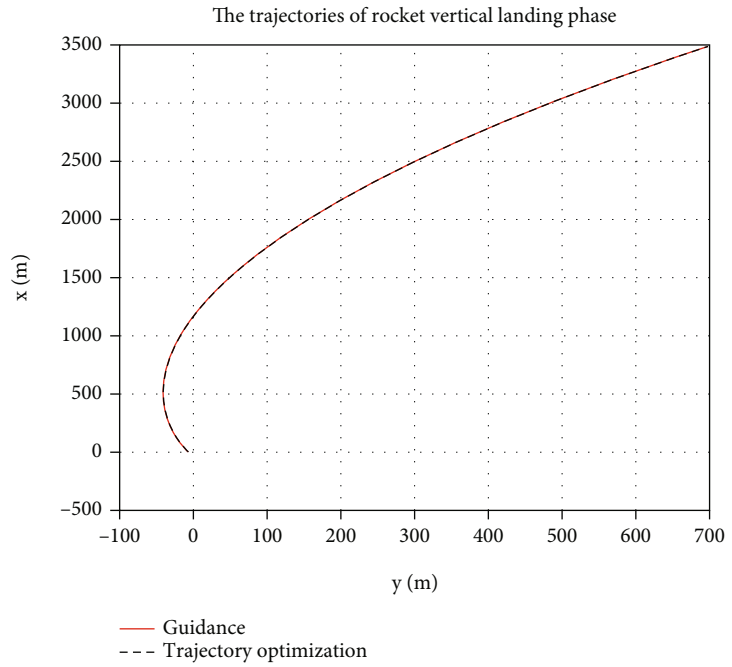

Figure 4: The trajectories of rocket vertical landing phase obtained by the guidance method and the trajectory optimization algorithm.

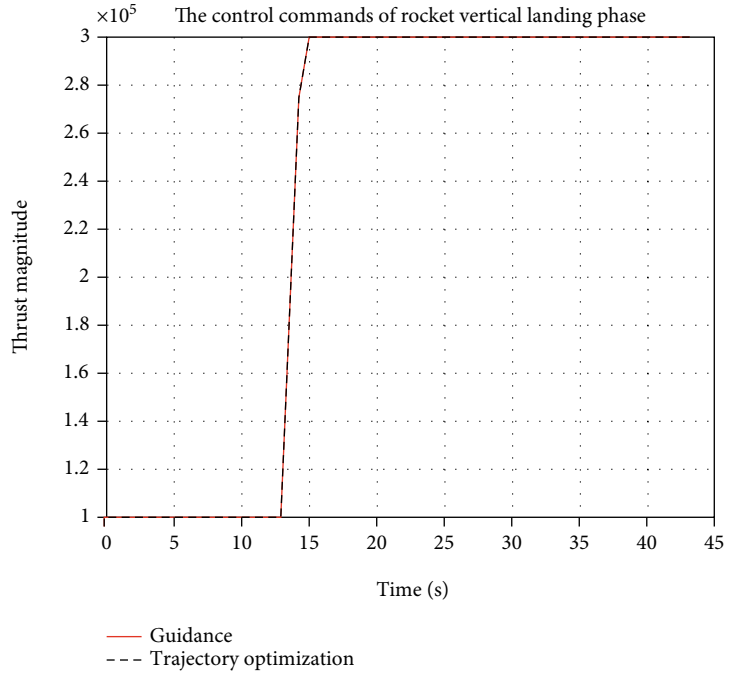

Figure 5: The control commands of rocket vertical landing phase obtained by the guidance method and the trajectory optimization algorithm.

Table 2: Terminal states.

	$t_f\,(s)$	$r_x(t_f)\,(m)$	$r_y(t_f)\,(m)$	$v_x(t_f)\,(m/s)$	$v_y(t_f)\,(m/s)$	$m(t_f)\,(kg)$
Trajectory optimization	43.18	-2.0085	-1.1821	-0.0646	-0.0332	23137.09
Guidance	43.18	-0.0276	-0.0241	-0.0210	-0.0043	23135.79

Table 3: Landing errors.

	Landing position error/m	Landing velocity error/(m/s)
Trajectory optimization	2.3305	0.0726
Guidance	0.0366	0.0214

Table 4: Landing errors and masses under the condition that the navigation system has faults.

	Landing position error/m	Landing velocity error/(m/s)	Landing mass/kg
$\varepsilon = 5$	2.4665	5.7779	23331.19
$\varepsilon = 10$	3.5750	5.8301	23329.48
$\varepsilon = 15$	4.0624	6.5514	23355.94

Table 5: Landing errors and masses under the condition that thrust cannot change continuously.

	Landing position error/m	Landing velocity error/(m/s)	Landing mass/kg
Trajectory optimization	407.6401	4.8062	23081.57
Guidance	0.0914	0.3145	23136.54

Table 6: Landing errors and masses under the condition that thrust magnitude has deviations.

	Landing position error/m	Landing velocity error/(m/s)	Landing mass/kg
Trajectory optimization	359.1367	6.3939	23137.13
Guidance	0.0185	0.0209	23133.57

Table 7: Landing errors and masses of the trajectory optimization algorithm.

Deviation	Landing position error/m	Landing velocity error/(m/s)	Landing mass/kg
$C_D + 15\%$	289.5933	4.9790	23135.95
$C_D - 15\%$	356.5689	6.2598	23135.95
$\rho + 15\%$	289.9584	4.9790	23135.95
$\rho - 15\%$	356.5689	6.2598	23135.95

Table 8: Landing errors and masses of the guidance method.

Deviation	Landing position error/m	Landing velocity error/(m/s)	Landing mass/kg
$C_D + 15\%$	0.0035	0.0239	23230.63
$C_D - 15\%$	4.1627	29.2456	23957.32
$\rho + 15\%$	0.0035	0.0239	23230.63
$\rho - 15\%$	4.1627	29.2456	23957.32

greater when the thrust cannot be continuously changed, but it still does not exceed 1 m/s. The simulation experiments demonstrate that the guidance method proposed in this paper is fault-tolerant and robust to control system faults.

6.2.3. Estimates of Parameters Have Deviations. It is assumed that the actual drag coefficient and atmospheric density differ by ±15% from the estimated values. We apply the trajectory optimization algorithm and the guidance method proposed in this paper to conduct simulation experiments using the estimated parameters. Then, we perform numerical integration using the actual parameters to obtain real terminal states. The landing errors and landing masses of the two algorithms under the fault condition are shown in Tables 7 and 8, respectively, (where $C_D + 15\%$ indicates that the actual drag coefficient has a value of $1.15 \times C_D$).

It can be seen from Tables 7 and 8 that the guidance method proposed in this paper can significantly improve the landing position accuracy in the case of the drag coefficient or the atmospheric density having deviations. When the actual values of the two parameters are larger than the estimated values, the accuracy of the landing velocity is also significantly improved. This experiment demonstrates the fault tolerance and robustness of the guidance method to the landing position in the presence of parameter deviations. However, when the actual values of the two parameters are smaller than the estimated values, the landing velocity errors obtained by the guidance method will become larger. We refer to the analysis of Ref. [19] and learn that when the actual values of the two parameters are decreased, part of the mechanical energy of the rocket cannot be dissipated as expected; and when the thrust magnitude saturates, there is no additional energy to compensate for the undissipated mechanical energy, leading to the error of the optimal trajectory becomes larger. Subsequent research on guidance methods should attempt to address this issue.

6.2.4. Combination of Various Faults. Finally, we consider navigation system failures, control system failures (thrust magnitude has deviations), drag coefficient deviations, and atmospheric density deviations at the same time and carry out Monte Carlo simulations (200 cases). For the navigation system and control system, the failures are added in the same way as the previous experiments. Drag coefficient deviations and atmospheric density deviations are considered to be normally distributed, and their means are set to zero. The 3σ values are, respectively, taken as $C_D \sim 15\%$, $\rho \sim 15\%$.

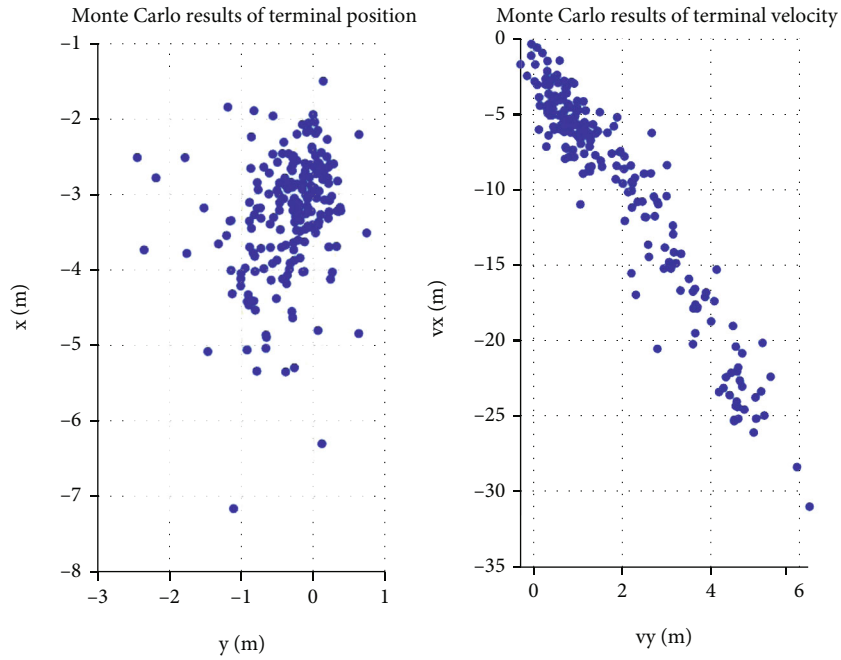

Figure 6: Monte Carlo results of terminal position and terminal velocity.

Results of terminal position and terminal velocity are shown in Figure 6. The maximum errors are (7.2551 m, 31.6108 m/s). This experiment proves that the guidance method can still keep the terminal states within acceptable limits under the combination of various faults.

7. Conclusions

In this paper, a fault-tolerant guidance method is proposed to realize online guidance of rocket vertical landing. The main contribution of this paper is that we propose a piecewise guidance algorithm. We first embed a trajectory optimization algorithm based on convex optimization in the MPC framework and then put forward a piecewise method to cope with the bang-bang characteristic of the thrust magnitude. An integrated GNC system is designed to enhance the fault-tolerance and robustness of the guidance method, which constitutes another contribution of this paper. Simulation experiments are conducted under conditions of no faults and deviations, navigation system failures, control system failures, drag coefficient deviations, and atmospheric density deviations, respectively, proving the reliability and fault-tolerance of the guidance method. The proposed method shows great potential for online use.

Data Availability

Data are available on request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This study was cosupported in part by the National Natural Science Foundation of China (nos. 61903146 and 61873319).

References

[1] H. Zhao, H. Pan, C. Wang, X. Yi, and H. Hu, "Vertical landing guidance navigation and control of reusable launch vehicle," *Missiles and Space Vehicles*, vol. 1, pp. 76–81, 2021.

[2] Z. Song, C. Wang, S. Theil et al., "Survey of autonomous guidance methods for powered planetary landing," *Frontiers of Information Technology & Electronic Engineering*, vol. 21, no. 5, pp. 652–674, 2020.

[3] J. Meditch, "On the problem of optimal thrust programming for a lunar soft landing," *IEEE Transactions on Automatic Control*, vol. 9, no. 4, pp. 477–484, 1964.

[4] N. Cui, D. Guo, K. Li, and C. Z. Wei, "A survey of numerical methods for aircraft trajectory optimization," *Tactical Missile Technology*, vol. 5, pp. 37–51, 2020.

[5] S. Y. Chen and Q. L. Xia, "A multiconstrained ascent guidance method for solid rocket-powered launch vehicles," *International Journal of Aerospace Engineering*, vol. 2016, Article ID 6346742, 11 pages, 2016.

[6] L. Mu, J. Yu, Y. Zhang et al., "Landing trajectory generation using gauss pseudo-spectral method," in *2021 40th Chinese control conference (CCC)*, pp. 3751–3755, Shanghai, China, 2021.

[7] Z. Song and C. Wang, "Development of online trajectory planning technology for launch vehicle return and landing. astronautical systems," *Engineering Technology*, vol. 6, 2019.

[8] B. Acikmese and S. R. Ploen, "Convex programming approach to powered descent guidance for mars landing," *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 5, pp. 1353–1366, 2007.

[9] L. Blackmore, B. Açikmeşe, and D. P. Scharf, "Minimum-landing-error powered-descent guidance for Mars landing using convex optimization," *Journal of Guidance, Control, and Dynamics*, vol. 33, no. 4, pp. 1161–1171, 2010.

[10] B. Açıkmeşe and L. Blackmore, "Lossless convexification of a class of optimal control problems with non- convex control constraints," *Automatica*, vol. 47, no. 2, pp. 341–347, 2011.

[11] X. Liu and P. Lu, "Robust trajectory optimization for highly constrained rendezvous and proximity operations," in *AIAA guidance Navigation, and Control (GNC) Conference*, p. 4720, Boston, MA, USA, 2013.

[12] X. Liu, Z. Shen, and P. Lu, "Entry trajectory optimization by second-order cone programming," *Journal of Guidance, Control, and Dynamics*, vol. 39, no. 2, pp. 227–241, 2016.

[13] X. Liu, "Fuel-optimal rocket landing with aerodynamic controls," *Journal of Guidance, Control, and Dynamics*, vol. 42, no. 1, pp. 65–77, 2019.

[14] Z. Wang and M. J. Grant, "Minimum-fuel low-thrust transfers for spacecraft: a convex approach," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 54, no. 5, pp. 2274–2290, 2018.

[15] X. Lan, W. Xu, Z. Zhao, and G. Liu, "Autonomous control strategy of a swarm system under attack based on projected view and light transmittance," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 3, pp. 648–655, 2021.

[16] X. Liu, "Fuel-optimal rocket landing with aerodynamic controls," in *AIAA Guidance, Navigation, and Control Conference*, pp. 1732–1747, Grapevine, TX, USA, 2017.

[17] J. Zhao, Y. Huang, H. Li, and X. He, "Uncertainty optimization for return trajectory of vertical takeoff and vertical landing launch vehicle," *Acta Aeronautica et Astronautica Sinica*, , pp. 255–269, 2021.

[18] J. Wang, N. Cui, J. Guo, and D. Xu, "High precision rapid trajectory optimization algorithm for launch vehicle landing," *Control Theory & Applications*, vol. 35, no. 3, pp. 389–398, 2018.

[19] J. Wang, N. Cui, and C. Wei, "Optimal rocket landing guidance using convex optimization and model predictive control," *Journal of Guidance, Control, and Dynamics*, vol. 42, no. 5, pp. 1078–1092, 2019.

[20] Z. An, F. Xiong, and Z. Liang, "Landing-phase guidance of rocket using bias proportional guidance and convex optimization," *Acta Aeronautica et Astronautica Sinica*, vol. 41, no. 5, pp. 242–255, 2020.

[21] M. Szmuk, B. Acikmese, and A. W. Berning, "Successive convexification for fuel-optimal powered landing with aerodynamic drag and non-convex constraints," in *AIAA Guidance, Navigation, and Control Conference*, San Diego, California, USA, 2016.

[22] Z. Zhang, Y. Ma, G. Y. Geng, and M. L. Yu, "Convex optimization method used in the landing-phase on-line guidance of rocket vertical recovery," *Journal of Ballistics*, vol. 29, no. 1, 2017.

[23] N. Shao and X. Yan, "Multi-stage trajectory optimization for vertical pin-point landing of a reusable launch vehicle," *Journal of Astronautics*, vol. 35, 2019.

[24] L. Xie, H. Zhang, X. Zhou, and G. Tang, "A convex programming method for rocket powered landing with angle of attack constraint," *Access*, vol. 8, pp. 100485–100496, 2020.

[25] M. Yang and G. You, "A two-stage successive convexification method for the powered descent guidance problem," *Scientia Sinica Mathematica*, vol. 50, no. 9, pp. 234–247, 2020.

[26] D. Malyuta, T. Reynolds, M. Szmuk, M. Mesbahi, B. Acikmese, and J. M. Carson, "Discretization performance and accuracy analysis for the rocket powered descent guidance problem," in *AIAA Scitech 2019 Forum*, p. 925, San Diego, California, USA, 2019.

[27] S. Yuan, T. Liu, and Y. Huang, "Switched adaptive resilient control of missile autopilot systems," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 57, no. 6, pp. 4227–4237, 2021.

[28] Y. Song, W. Zhang, X. Miao, Z. Zhang, and S. Gong, "Onboard guidance algorithm for the powered landing phase of a reusable rocket," *Journal of Tsinghua University (Science and Technology)*, vol. 61, no. 3, pp. 230–239, 2021.

[29] A. Botelho, M. Martinez, C. Recupero, A. Fabrizi, and G. de Zaiacomo, "Design of the landing guidance for the retropropulsive vertical landing of a reusable rocket stage," *CEAS Space Journal*, vol. 14, pp. 551–564, 2022.

[30] D. Morgan, S. J. Chung, and F. Y. Hadaegh, "Model predictive control of swarms of spacecraft using sequential convex programming," *Journal of Guidance, Control, and Dynamics*, vol. 37, no. 6, pp. 1725–1740, 2014.

[31] W. Dunham, C. Petersen, and I. Kolmanovsky, "Constrained control for soft landing on an asteroid with gravity model uncertainty," in *2016 American Control Conference (ACC)*, pp. 5842–5847, Boston, MA, USA, 2016.

[32] U. Eren, A. Prach, B. B. Koçer, S. V. Raković, E. Kayacan, and B. Açıkmeşe, "Model predictive control in aerospace systems: current state and opportunities," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 7, pp. 1541–1566, 2017.

[33] C. Wang and Z. Song, "Convex model predictive control for rocket vertical landing," in *2018 37th Chinese control conference (CCC)*, pp. 9837–9842, Wuhan, China, 2018.

[34] K. Cui, W. Han, Y. Liu, X. Wang, X. Su, and J. Liu, "Model predictive control for automatic carrier landing with time delay," *International Journal of Aerospace Engineering*, vol. 2021, Article ID 8613498, 19 pages, 2021.

[35] J. Guadagnini, M. Lavagna, and P. Rosa, "Model predictive control for reusable space launcher guidance improvement," *Acta Astronautica*, vol. 193, pp. 767–778, 2022.

[36] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.0 beta," 2013, http://cvxr.com/cvx.

[37] M. Grant and S. Boyd, "Graph implementations for nonsmooth convex programs, Recent Advances in Learning and Control (a Tribute to M. Vidyasagar)," in *Lecture Notes in Control and Information Sciences*, V. Blondel, S. Boyd, and H. Kimura, Eds., pp. 95–110, Springer, 2008.

[38] E. D. Andersen, C. Roos, and T. Terlaky, "On implementing a primal-dual interior-point method for conic quadratic optimization," *Mathematical Programming*, vol. 95, no. 2, pp. 249–277, 2003.