JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 42, No. 5, May 2019

Optimal Rocket Landing Guidance Using Convex Optimization and Model Predictive Control

Jinbo Wang,* Naigang Cui,† and Changzhu Wei‡
Harbin Institute of Technology, 150001 Harbin, People's Republic of China

DOI: 10.2514/1.G003518

In this paper, a novel guidance algorithm based on convex optimization, pseudospectral discretization, and a model predictive control (MPC) framework is proposed to solve the highly nonlinear and constrained fuel-optimal rocket landing problem. The main strategy is to solve the guidance problem by implementing online trajectory optimization in a receding-horizon manner and feeding the rocket with the most recently updated optimal control commands. A pseudospectral-improved successive convexification (PISC) algorithm is adopted to solve the trajectory optimization problem due to its high solution accuracy and computation speed. The PISC algorithm is then embedded in the MPC framework to construct the guidance algorithm. The recursive feasibility of the MPC-based guidance algorithm is guaranteed by executing the original and relaxed trajectory optimization algorithms in parallel. Additionally, the boundedness of the guidance error is proved. Therefore, the guidance algorithm is optimal, robust, and practically stable under constraints and disturbances. Numerical experiments are provided to demonstrate the effectiveness of the proposed algorithm.

Nomenclature

A, B, T	=	matrices describing the linearized dynamics
C_D, C_L	=	aerodynamic drag and lift force coefficients
D	=	flipped Radau pseudospectral differentiation matrix
$\bar{D}, L$	=	aerodynamic drag and lift forces, N
$\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})$	=	right-hand side of the dynamical equation
$h_{\Omega x,y,z}$	=	Earth self-rotation terms
I_{sp}	=	specific impulse of the rocket engine, s
m	=	rocket mass, kg
N	=	number of discretization points
$\boldsymbol{p}, \boldsymbol{p}_0$	=	rocket's actual and nominal parameters
$r_{x,y,z}$	=	position components, m
T	=	thrust magnitude, N
$\mathcal{T}_i$	=	terminal flight time in the ith model predictive control implementation, s
$T_{x,y,z}$	=	thrust components, N
t_0, t_f	=	engine ignition time and terminal flight time, s
V	=	velocity, m/s
$\boldsymbol{x}, \boldsymbol{u}$	=	state and control variables
$\bar{\boldsymbol{x}}, \underline{\boldsymbol{x}}$	=	nominal and actual states used in the model predictive control implementation
α	=	angle of attack, deg
Γ	=	slack variable that relaxes the thrust magnitude constraint, N
δ_Δ	=	deviation between the value of variable Δ and the expected value
$\eta_{y,z}$	=	pointing direction angles of the thrust vector, deg
θ	=	flight-path angle, deg
$\boldsymbol{\xi}$	=	external disturbance
$\boldsymbol{\pi}_x$	=	estimate of the upper bound of the deviation between actual and nominal states
σ	=	heading angle, deg
τ_i	=	ith collocation point
χ	=	adjustment parameter of the decrease rate of the number of discretization points
ω	=	penalty parameter in the optimization problem

Received 17 December 2017; revision received 2 November 2018; accepted for publication 29 November 2018; published online 24 January 2019.

*Ph.D. Candidate, Department of Aerospace Engineering. Member AIAA.
†Professor, Department of Aerospace Engineering.
‡Associate Professor, Department of Aerospace Engineering.

I. Introduction

REUSABLE launch vehicle (RLV) technology has been a popular topic in the aerospace field for decades. In recent years, with the successful landings and reuse of SpaceX's Falcon-9 and Blue Origin's New Shepard rockets, vertical takeoff/vertical landing (VTVL) RLVs (referred to as VTVL-RLVs) have received increasing attention [1,2]. Landing guidance is one of the most challenging aspects of rocket landing. Guiding a rocket booster to return to and land on a planet with an atmosphere is not a trivial task. A large range of deceleration by limited control ability is required, various path constraints and narrow terminal constraints should be followed, and the fuel cost should be minimized to enhance carrying capacity and emergency responsiveness. Addressing this problem with traditional reentry or ascent guidance methods is difficult due to the need to consider optimality under stringent constraints. Furthermore, online and onboard operational abilities of the algorithm are required to enhance the autonomy of the rocket. Motivated by the computational guidance and control philosophy [3], a novel numerical optimal rocket landing guidance algorithm based on convex optimization theory, a pseudospectral (PS) discretization method, and a model predictive control (MPC) framework is proposed.

With the development of computing technologies, optimization theories, and algorithms, online trajectory optimization is becoming an effective approach for solving aerospace guidance problems [4]. Among the many trajectory optimization methods that focus on online operational ability [5–8], methods based on convex optimization have great advantages in mathematical principles and calculation due to their rapid and deterministic convergence properties [8,9]. However, most aerospace applications are dominated by nonconvex dynamics and constraints. Thus, the greatest challenge to applying convex optimization lies in convexifying the problem. Lossless convexification and successive convexification are two effective techniques for handling nonconvexity in optimization. The concept of lossless convexification was proposed to solve the Mars landing guidance problem [10,11], and it has been generalized to solve several kinds of general optimal control problems [12,13]. Successive convexification (also referred to as sequential convex programming or sequential convex optimization) was proposed in Refs. [8,14–17] for convexifying nonlinear terms in system dynamics and constraints; it is an iterative technique that repeatedly solves convex subproblems until the solution converges. Convex-optimization-based methods have been very popular in VTVL rocket applications in recent years.

Liu analyzed the cooperative effects of thrust and aerodynamic forces on the fuel cost of the landing flight by using a successive convexification algorithm [2]. Szmuk and Açıkmeşe studied the fuel-optimal landing problem by combining lossless and successive convexification techniques [16]. Scharf et al. reported successful onboard implementation experiments with the lossless-convexification-based guidance algorithm [18].

In addition to the solution algorithms, in terms of problem discretization, methods based on PS discretization have been reported to have the potential to be used for online trajectory optimization [4–7]. Compared with other discretization methods, PS methods can achieve the same accuracy by using fewer discretization points [19,20]. Moreover, PS methods converge faster for smooth problems; this is referred to as the "spectral accuracy" [21]. In previous work by Wang and Cui [22] and Sagliano [23], the idea of combining the advantages of the PS method and convex optimization to rapidly obtain accurate optimal solutions was proposed; the rapidness and high accuracy of the proposed algorithms are due to the properties of the convex optimization method and PS discretization, respectively.

Rapid online trajectory optimization is a promising approach for achieving optimal guidance, and it is the cornerstone of the landing guidance method studied in this paper. However, trajectory optimization alone is not robust enough to overcome the uncertainties and disturbances encountered during the flight. MPC is a form of control based on online optimization [24]. Because MPC recursively solves the problem with updated system states, it has a certain degree of disturbance rejection ability. The vast applications of MPC in the aerospace guidance field were described in the latest survey paper [25]. In many recent MPC works, convex optimization was embedded into the MPC framework to provide better performance [14,26–28]. One of the major advantages that convex optimization provides is rapid optimization computation in each MPC update, through which the update frequency of the MPC algorithm can increase. As shown in Sec. IV, the guidance performance is closely related to the trajectory update frequency. In addition to computation speed, recursive feasibility is an important issue in MPC implementation [27]. If recursive feasibility cannot be guaranteed, then the actual trajectory update frequency would decrease.

This paper is devoted to the rocket landing guidance application, which requires an efficient online algorithm that can maneuver the rocket to the landing site with minimal fuel cost while simultaneously ensuring safety under constraints and disturbances. To fulfill these requirements, we propose an optimal guidance algorithm based on online trajectory optimization and the MPC framework. Particularly, with a novel parallel implementation strategy, the guidance algorithm is recursively feasible and practically stable.

First, a high-fidelity optimization model of the VTVL-RLV is built, and unique features of the PS method are exploited to provide a more accurate solution. Then, to take advantage of the attractive properties of convex optimization, a previously developed pseudospectral-improved successive convexification (PISC) algorithm is adopted to convexify and solve the nonlinear optimization problem. The modeling/discretization procedures and the development of the PISC algorithm are closely linked, and they support one of the main contributions of the paper: solving the fuel-optimal rocket landing problem with improved solution accuracy and convergence speed.

Next, we wrap the PISC algorithm with the MPC framework, and a parallel feasibility-guaranteed model predictive guidance (PFGMPG) algorithm is proposed. In each MPC update, a relaxed-PISC algorithm and a standard-PISC algorithm are executed in parallel with the most current state information, and updated optimal guidance commands are fed to the rocket in a feedforward manner in real time. With such a parallel implementation strategy, the recursive feasibility and guidance error boundedness are proven; thus, the practical stability of PFGMPG is guaranteed. Numerical experiments are performed to comprehensively evaluate the guidance algorithm, and the effectiveness of the algorithm is demonstrated. The development of the PFGMPG algorithm constitutes another main contribution of the paper.

II. Problem Formulation

In this section, the system dynamics and constraints to be followed by the VTVL-RLV (subsequently referred to as the "rocket") during the landing flight are formulated. Then, we formulate and discuss the fuel-optimal trajectory optimization problem.

In this paper, because the guidance commands are generated directly by trajectory optimization according to system dynamics, the fidelity of the dynamics model is of crucial importance to the guidance performance. To enhance model fidelity, the aerodynamic lift and drag forces are considered, and we design the angle of attack (AOA) as a control freedom together with the engine thrust. Additionally, the "coast phase" before engine ignition is considered in the modeling because the designs of the AOA profile of the unpowered flight and the engine ignition time greatly influence the trajectory.

A. Dynamics and Constraints

The flight of the rocket begins before the final engine ignition and ends at the terminal touchdown. Thus, from the perspective of the working states of the rocket engine, the flight is divided into a coast phase and a "powered phase." During the coast phase, the rocket is unpowered, and the optimal AOA profile and the optimal engine ignition time are designed by online trajectory optimization to meet the path constraints. The powered phase starts at the engine ignition time. Thereafter, the AOA and the thrust control the rocket in a cooperative manner.

The AOA command and the pointing direction of the thrust vector (which is aligned with the rocket body) generated by the guidance algorithm are realized by the attitude control system, which is equipped with grid fins and reaction control systems. The attitude control logic is out of the scope of the current research, and the control performance is assumed to be ideal. The sideslip and bank angles are assumed to be zero or negligible. Thus, the aerodynamic lateral force is omitted. We adopt the spherical rotating Earth model and the isothermal-barotropic atmosphere model [6]. The three-degree-of-freedom dynamics model of the rocket can be expressed as follows:

$$\begin{cases} \dot{V} = -\frac{\cos\alpha T_x + \sin\alpha T_y + \bar{D}}{m} - \frac{\mu\sin\theta\cos\sigma}{r^2} + h_{\Omega x} \\ \dot{\theta} = \frac{-\sin\alpha T_x + \cos\alpha T_y + L}{mV\cos\sigma} - \frac{\mu\cos\theta}{Vr^2\cos\sigma} + h_{\Omega y} \\ \dot{\sigma} = -\frac{T_z}{mV} + \frac{\mu\sin\theta\sin\sigma}{Vr^2} + h_{\Omega z} \\ \dot{r}_x = V\cos\theta\cos\sigma \\ \dot{r}_y = V\sin\theta\cos\sigma \\ \dot{r}_z = -V\sin\sigma \\ \dot{m} = -\frac{T}{I_{\mathrm{sp}}g_0} \end{cases} \tag{1}$$

where V is the velocity, θ is the flight-path angle, σ is the heading angle, $r_{x,y,z}$ are the position components, m is the rocket mass, and the state vector of the system is defined as $\boldsymbol{x} := [V, \theta, \sigma, r_x, r_y, r_z, m]^T \in \mathbb{R}^7$. $T_{x,y,z}$ are the engine thrust components, $T = \sqrt{T_x^2 + T_y^2 + T_z^2}$, α is the AOA, and the control vector is defined as $\boldsymbol{u} := [T_x, T_y, T_z, \alpha]^T \in \mathbb{R}^4$. Although the state and control variables are time dependent, the notation of (t) is omitted in Eq. (1) and throughout the paper for simplicity [e.g., V represents $V(t)$]. The state variables are defined with respect to the "landing reference frame" $Oxyz$, which is illustrated in Fig. 1. The origin of $Oxyz$ is at the landing target, the Oy axis points up, and the Ox axis is aligned with the heading direction of the projection of the velocity vector at initial time onto the xOz plane. Also, r is the radial distance from the Earth center to the rocket, and μ is the gravitational parameter. $L = qS_{\mathrm{ref}}C_L$ and $\bar{D} = qS_{\mathrm{ref}}C_D$ are the lift and drag forces, where $q = \rho V^2/2$ is the dynamic pressure, $\rho = \rho_0 e^{-r_y/h_0}$ is the air density, ρ_0 is the reference density, h_0 is the reference height, S_{ref} is the reference area of the rocket, and C_L and C_D are the lift and drag coefficients, respectively. I_{sp} is the specific impulse of the rocket engine, and g_0 is the gravitational acceleration at sea level. Also, $h_{\Omega x,y,z}$ are the Earth rotation-related terms, and their detailed

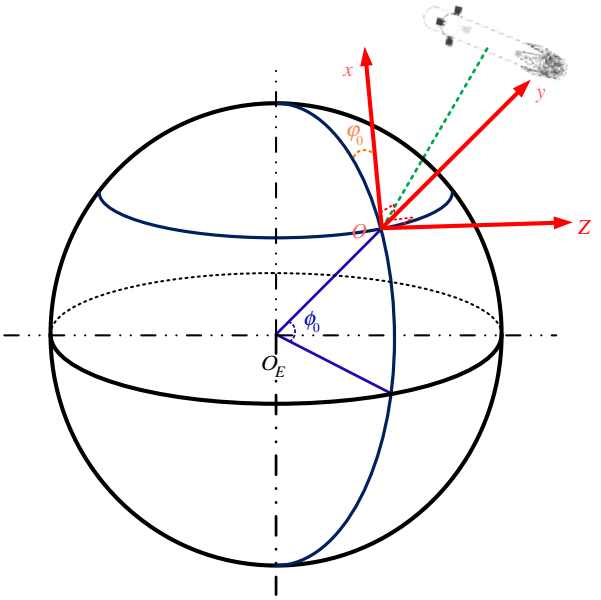

Fig. 1 Landing reference frame.

expressions can be found in the Appendix. The values of constant parameters ρ_0, h_0, and μ can be found in Ref. [6].

To simplify the successive convexification procedure, which will be discussed in Sec. III, we model the thrust vector as thrust components $T_{x,y,z}$ rather than thrust magnitude T and pointing direction angles (i.e., the attitude of the rocket body) $\eta_{y,z}$. Although the magnitude/pointing angle form is more straightforward when analyzing the guidance results, there is a simple conversion relationship between the two forms:

$$\begin{cases} T = \sqrt{T_x^2 + T_y^2 + T_z^2} \\ \eta_y = \tan^{-1}(T_y/T_x) \\ \eta_z = \sin^{-1}(T_z/T) \end{cases} \tag{2}$$

The initial time of the coast phase is set to zero. The initial and terminal times of the powered phase are defined as t_0 and t_f. During the coast phase, the engine thrust is zero; during the powered phase, there is a minimum throttle level of the rocket engine for reliability reasons. Thus, the thrust is bounded as follows:

$$T_x = T_y = T_z = 0, \qquad \forall\, t \in [0, t_0) \tag{3}$$

$$T_{\min} \le \sqrt{T_x^2 + T_y^2 + T_z^2} \le T_{\max}, \qquad \forall\, t \in [t_0, t_f] \tag{4}$$

where $T_{\min}$ and $T_{\max}$ are the minimum and maximum values of the thrust. It should be noted that the constraint in Eq. (4) is nonconvex because the minimum value is nonzero. The rates of change of the thrust components are constrained according to the physical limits of the actuators:

$$-\dot{T}_i^{\max} \le \dot{T}_i \le \dot{T}_i^{\max}, \qquad i = x, y, z, \qquad \forall\, t \in [t_0, t_f] \tag{5}$$

where $\dot{T}_{x,y,z}^{\max}$ are the maximum values of the rates of change. Additionally, according to Eq. (2), the constraints on pointing direction angles are equivalent to the constraints on thrust components:

$$\begin{cases} -T_x \tan \eta_y^{\max} \le T_y \le T_x \tan \eta_y^{\max} \\ -T \sin \eta_z^{\max} \le T_z \le T \sin \eta_z^{\max} \end{cases}, \qquad \forall\, t \in [t_0, t_f] \tag{6}$$

where $\eta_y^{\max}$ and $\eta_z^{\max}$ are the maximum pointing direction angles.

For powered landing flight, estimating the aerodynamic coefficients is challenging. The engine jet plume would have complex effects on the flowfield around the rocket. No open publications provide detailed aerodynamic data. In this paper, we adopt the approximate data estimated by compensating and fitting the computational fluid dynamics results of an unpowered rocket. Estimation errors are treated as parameter uncertainties during the guidance process. The approximated functions of the aerodynamic coefficients are $C_D = C_{D0} + C_D^{\alpha} \cdot \alpha + C_D^{\alpha^2} \cdot \alpha^2$ and $C_L = C_L^{\alpha} \cdot \alpha + C_L^{\alpha^2} \cdot \alpha^2$. During the flight, the AOA and its rate of change are bounded as follows:

$$\begin{cases} -\alpha_{\max} \le \alpha \le \alpha_{\max} \\ -\dot{\alpha}_{\max} \le \dot{\alpha} \le \dot{\alpha}_{\max} \end{cases}, \qquad \forall\, t \in [0, t_f] \tag{7}$$

where $\alpha_{\max}$ and $\dot{\alpha}_{\max}$ are the maximum AOA and the maximum rate of change of the AOA.

The state-control path constraints include those on the normal load, the heating rate, and dynamic pressure:

$$\begin{cases} |(T_y + L\cos\alpha + \bar{D}\sin\alpha)/(mg)| \le n_{\max} \\ k_Q \sqrt{\rho} V^{3.15} \le \dot{Q}_{\max} \\ \rho V^2/2 \le q_{\max} \end{cases}, \qquad \forall\, t \in [0, t_f] \tag{8}$$

where $n_{\max}$, $\dot{Q}_{\max}$, and $q_{\max}$ are the maximum values of the normal load, heating rate, and the dynamic pressure, respectively. The constant k_Q is set as 2.334×10^{-5}, which is based on the rocket's heat shield properties [6].

For precision soft landing, the constraints on the terminal states and control include the following:

$$\begin{cases} \varepsilon_V^l \le V_f \le \varepsilon_V^u, -\varepsilon_\theta \le \theta_f \le \varepsilon_\theta, -\varepsilon_\sigma \le \sigma_f \le \varepsilon_\sigma, \\ \varepsilon_y^l \le r_{yf} \le \varepsilon_y^u, -\varepsilon_x \le r_{xf} \le \varepsilon_x, -\varepsilon_z \le r_{zf} \le \varepsilon_z, \\ m_f \ge m_{\text{dry}}, -\varepsilon_\alpha \le \alpha_f \le \varepsilon_\alpha \end{cases} \tag{9}$$

where $\varepsilon_V^{l,u}, \varepsilon_\theta, \varepsilon_\sigma, \varepsilon_y^{l,u}, \varepsilon_x, \varepsilon_z$, and ε_α are the tolerable error values of the terminal states and the AOA; and m_{dry} is the dry mass of the rocket. Here, we set the terminal height–velocity windows as $r_{yf} \in [\varepsilon_y^l, \varepsilon_y^u]$ and $V_f \in [\varepsilon_V^l, \varepsilon_V^u]$ rather than setting r_{yf} and V_f as zero, and the magnitude of $\varepsilon_y^{l,u}$ and $\varepsilon_V^{l,u}$ are within the order of tens. These values are used because, in actual flight, the terminal touchdown guidance is usually performed in a closed-loop manner with advanced sensors (such as visual or optical sensors) to guarantee the touchdown precision. Moreover, the rocket's velocity and fuel consumption during the terminal touchdown are fairly small, and the constraints will remain satisfied.

Remark 1: The dimensional dynamics model is exhibited here for clarity, whereas the nondimensionalization method described in Ref. [29] has been used in numerical experiments for better robustness and numerical stability.

B. Trajectory Optimization Problem

The optimization objective is to minimize the fuel cost; equivalently, the terminal mass of the rocket is set as the performance index. The trajectory optimization problem can be formulated as follows:

Continuous problem 0 (CP0):

$$\begin{aligned} &\min -m_f \\ &\text{subject to Eqs. (1), (3)-(9)} \end{aligned} \tag{10}$$

Here, the trajectory optimization problem is modeled as a free-time problem, and the engine ignition time and the terminal flight time are determined by the optimization algorithm (rather than fixed offline).

It should be noted that the preceding formulation includes both the coast phase and the powered phase. However, in the MPC-based guidance algorithm, which will be discussed in Sec. IV, the

aforementioned problem is solved repeatedly in a receding-horizon manner. Thus, according to the partitioning of flight phases, if the engine is ignited, only the powered phase will remain in the optimization problem.

Here, we take CP0, which is nonconvex, as the original problem. The sources of nonconvexity are the nonzero minimal thrust constraint and the nonlinear dynamics/path constraints.

For the nonconvex thrust magnitude constraint, Açıkmeşe and Ploen [10], Blackmore et al. [11], Harris and Açıkmeşe [12], and Blackmore et al. [13] proposed the concept of lossless convexification. Using the results of the aforementioned references, a slack variable Γ is introduced to relax Eq. (4). The nonconvex constraint is then transformed into the following convex ones:

$$\sqrt{T_x^2 + T_y^2 + T_z^2} \le \Gamma \tag{11}$$

$$T_{\min} \le \Gamma \le T_{\max} \tag{12}$$

The mass dynamics and the second constraint in Eq. (6) are also transformed as follows:

$$\dot{m} = -\Gamma/(I_{\mathrm{sp}} g_0) \tag{13}$$

$$-\Gamma \sin \eta_z^{\max} \le T_z \le \Gamma \sin \eta_z^{\max} \tag{14}$$

After this transformation, Eq. (11) is a standard second-order cone constraint and Eq. (12) is a linear constraint. The optimization problem that uses the relaxed constraints in Eqs. (11–14) instead of the original ones is referred to as continuous problem 1 (CP1). The slack variable Γ is a new control variable; thus, the control vector is redefined as $\boldsymbol{u} \coloneqq [T_x, T_y, T_z, \alpha, \Gamma]^T \in \mathbb{R}^5$.

The key question of applying the preceding transformation is about the losslessness; that is, whether the solution of CP1 is equivalent to that of CP0. Under an assumption that the path constraints are inactive almost everywhere on $[t_0, t_f]$, the losslessness proof of CP1 can be accomplished by using Pontryagin's principle [30]. Nonetheless, if the path constraints are active in some finite intervals, the proof would be challenging because the available necessary conditions provided by Pontryagin's principle are not sufficiently strong for this particular problem. Fortunately, for the present research, because the rocket is designed to be capable of surviving high-velocity entry, the constraints in Eq. (8) would rarely be active during the landing, in which the velocity of the rocket is usually relatively small. Moreover, when the bound values of the path constraints are set to be very small, such that the constraints are active in some intervals, numerical experiments show that the relaxation remains valid. Thus, the applicability of lossless convexification for CP0 can be ensured. The proof with the inactive-path-constraint assumption and numerical experiments with active constraints are provided in supplementary materials, whereas the proof of losslessness for cases with active constraints will require a dedicated study and is left for future work.

III. Pseudospectral-Improved Successive Convexification Algorithm

The previously developed PISC algorithm [22] is adopted to solve CP1. PISC is based on PS discretization and convex optimization, and it is evolved from classic successive convexification methods that have been studied extensively in the aerospace trajectory optimization and guidance fields [31]. The main contribution of PISC lies in the improvements in solution accuracy and convergence performance, which have crucial effects on the performance of MPC-based guidance.

Because the problem formulated in this paper is more complex than that of the original PISC work, the problem discretization and convexification are briefly discussed in this section. Additionally, a useful strategy of changing the number of discretization points during the guidance process is proposed in Sec. III.A.

A. Problem Discretization

Because CP1 is formulated as a free-time problem for higher fidelity, the engine ignition time and the terminal flight time should be designed coordinately to achieve the optimality. However, the free times would intensify the nonconvexity of the problem; thus, the problem must be properly discretized to facilitate the convexification procedure. Meanwhile, excessive computational burden should be avoided as much as possible. Moreover, as stated earlier, for the method proposed in this paper, the solution accuracy of the trajectory optimization is crucial to the guidance performance.

To meet the aforementioned requirements, we adopt the flipped Radau pseudospectral method to discretize CP1. The PS method enjoys higher solution accuracy as compared with other discretization methods. Additionally, by using the unique time domain mapping of the PS method, the successive convexification method can be easily extended to accommodate free-time problems.

For the fuel-optimal landing problem, the thrust profile usually shows bang–bang characteristics. Different from Ref. [22], we further divide the powered phase into three subphases to accommodate the "bang–arc" and "coast–arc" of the thrust profile so that each subphase of the optimization problem is smooth and the spectral accuracy of the PS method can be used. Moreover, the discretization matrix is sparser for multiphase PS discretization.

According to the aforementioned partitioning of flight phases, the discretized dynamical equation is defined as follows:

$$\sum_{j=0}^{N_p} D_{ij} \boldsymbol{x}_p(\tau_j) - \frac{t_{p\_f} - t_{p\_0}}{2} \boldsymbol{f}(\boldsymbol{x}_p(\tau_i), \boldsymbol{u}_p(\tau_i)) = 0, \quad (i = 1, \ldots, N_p, p = 0, 1, 2, 3) \tag{15}$$

where p is the subphase index; $p = 0$ indicates the coast phase; $p = 1, 2, 3$ indicates the subphases of the powered phase; t_{p_0} and t_{p_f} are the initial and terminal times of each subphase; D is the flipped Radau pseudospectral differentiation matrix [19]; $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})$ is the right-hand side of the dynamical equation; τ_0 is the discretization point at -1; $\tau_i (i = 1, \ldots, N_p)$ are the collocation points in the domain $(-1, 1]$; and N_p is the number of the collocation points. Additionally, the phase-connect (i.e., linkage) constraints are included as $\boldsymbol{x}_{p\_f} = \boldsymbol{x}_{(p+1)\_0}$ and $t_{p\_f} = t_{(p+1)\_0}$ $(p = 0, 1, 2)$.

It can be seen from Eq. (15) that, after discretization, the initial and terminal times of each subphase appear in the discretized algebraic constraints; that is, the time values are now optimal variables.

For the discretization of control constraints, because a time differential is involved in the rates of change of the AOA and thrust components, a factor of $t_{\mathrm{scale}} = (\hat{t}_f - \hat{t}_0)/2$ is included to transform the PS time to the physical time, as shown in Eqs. (16) and (17), where $\hat{t}_0$ and $\hat{t}_f$ are the estimates of the initial and terminal times of the current subphase, respectively. In the successive convexification algorithm, these estimates take the results of the last iteration:

$$-\dot{\alpha}_{\max} \le [\alpha(\tau_{i+1}) - \alpha(\tau_i)]/[t_{\mathrm{scale}}(\tau_{i+1} - \tau_i)] \le \dot{\alpha}_{\max} \tag{16}$$

$$-\dot{T}_m^{\max} \le [T_m(\tau_{i+1}) - T_m(\tau_i)]/[t_{\mathrm{scale}}(\tau_{i+1} - \tau_i)] \le \dot{T}_m^{\max}, \quad m = x, y, z \tag{17}$$

Now, we call the discretized optimization problem discretized problem 1 (DP1). DP1 is still nonconvex; however, the discretization has provided a foundation for the convexification.

It is known that problem size is one of the most important factors affecting the computational complexity of a trajectory optimization algorithm. Thus, to improve the computational efficiency of the MPC-based guidance algorithm, we propose a rule for determining the number of discretization points to be used in optimization updates. During the receding-horizon implementation of the proposed guidance algorithm, the time domain of the optimization becomes shorter in each update; thus, the number of discretization points can be decreased to relieve computational burden without jeopardizing discretization accuracy. The varying pattern of the

number of discretization points to be used in DP1 during the MPC update is designed as follows:

$$N_{i+1} = \left\lceil N_i \left(\frac{\Phi_{i+1}}{\Phi_i} \right)^{\chi} \right\rceil \tag{18}$$

where the ceiling function is defined as $\lceil x \rceil = \min\{n \in \mathbb{Z} | n \geq x\}$, N_i is the number of discretization points of the ith optimization update, Φ_i is the time horizon of the ith optimization, $\Phi_{i+1} = \Phi_i - \Delta t_i$ is the estimate of the time horizon of the $(i+1)$th optimization, and Δt_i is the time consumption of the ith update. χ is the adjustment parameter, and its value is designed to be less than one so that the rate of decrease of the number of discretization points is slower than a linear rate. Consequently, the problem size decreases over time, whereas the discretization accuracy increases.

B. Problem Convexification

After discretization, the nonconvex problem DP1 is convexified and solved by the PISC algorithm. In successive convexification algorithms, a series of linearized subproblems are iteratively solved, and the trust-region constraint is designed to ensure the solution of subproblems converges to that of the original nonlinear problem. For PISC, convergence is accelerated by dynamically updating the trust-region constraint during each iteration.

The development of PISC involves transformations among several equivalent problems, the linearization procedure, and the adjustment of the trust-region parameters during the successive iterations. Detailed discussions and a convergence proof of PISC can be found in Ref. [22]; here, we just list the different linearization results.

We refer to the convexified subproblem in PISC as the linearized subproblem (LP). In the LP, for the problem formulated in this paper, the linearized dynamics is as follows (the subscripts indicating discretization points and subphases are omitted for simplicity):

$$\begin{aligned} &2D \cdot \boldsymbol{x} + \bar{\boldsymbol{f}}(\boldsymbol{x}^k, \boldsymbol{u}^k) + A(\boldsymbol{x}^k, \boldsymbol{u}^k)(\boldsymbol{x} - \boldsymbol{x}^k) + B(\boldsymbol{x}^k, \boldsymbol{u}^k)(\boldsymbol{u} - \boldsymbol{u}^k) \\ &\quad + T(\boldsymbol{x}^k, \boldsymbol{u}^k)(\boldsymbol{t} - \boldsymbol{t}^k) + \bar{\boldsymbol{h}}_\Omega(\boldsymbol{x}^k) = \boldsymbol{0} \end{aligned} \tag{19}$$

where $\boldsymbol{t} \coloneqq [t_0, t_f]^T$; the superscript k indicates the iteration index; $\{\boldsymbol{x}^k, \boldsymbol{u}^k, \boldsymbol{t}^k\}$ represents the reference trajectory of the linearization, and its value is updated with the latest solution during the iteration; $\bar{\boldsymbol{f}}(\boldsymbol{x}, \boldsymbol{u}) = (t_0 - t_f) \cdot [\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) - \boldsymbol{h}_\Omega(\boldsymbol{x})]$; $\bar{\boldsymbol{h}}_\Omega(\boldsymbol{x}) = (t_0 - t_f) \cdot \boldsymbol{h}_\Omega(\boldsymbol{x})$; $\boldsymbol{h}_\Omega(\boldsymbol{x})$ contains the Earth self-rotation terms; and $A = \partial \bar{\boldsymbol{f}} / \partial \boldsymbol{x}$, $B = \partial \bar{\boldsymbol{f}} / \partial \boldsymbol{u}$, and $T = \partial \bar{\boldsymbol{f}} / \partial \boldsymbol{t}$ are the Jacobian matrices with respect to the optimal variables. The detailed forms and expressions of these matrices are provided in the Appendix.

For the linearized state-control path constraints, we have the following:

$$\begin{cases} |n(V^k, r_y^k, T_y^k, \alpha^k) + c_{11}(V - V^k) + c_{12}(r_y - r_y^k) + c_{13}(T_y - T_y^k) + c_{14}(\alpha - \alpha^k)| - n_{\max} \leq 0 \\ q(V^k, r_y^k) + c_{21}(V - V^k) + c_{22}(r_y - r_y^k) - q_{\max} \leq 0 \\ \dot{Q}(V^k, r_y^k) + c_{31}(V - V^k) + c_{32}(r_y - r_y^k) - \dot{Q}_{\max} \leq 0 \end{cases} \tag{20}$$

where c_{ij} are given in the Appendix.

Now, we discuss the verification of the feasibility and optimality of the PISC solution, which is an important procedure of approving the solution of an optimization software. In PISC, convex subproblems are solved by a mature interior point method (IPM) software [32], and then the solution of a subproblem can be returned or the infeasibility can be detected in polynomial time. If the IPM solver successively outputs feasible solutions (with respect to constraints) and the termination criteria are met, then the physical feasibility of the converged solution can be tested by propagating the system dynamics with the obtained control trajectory [6]. If the propagated and PISC state trajectories are coincident in a numerical sense, then the feasibility can be verified. According to the results in Sec. V.A (and abundant numerical experiments not included in the paper), the feasibility of the PISC solution can be guaranteed. For the optimality, besides theoretical proof, Pontryagin's and Bellman's principles of optimality [30,33] are effective and straightforward approaches to testing an obtained solution [6]. In this paper, the Bellman test is indirectly performed in the nominal guidance simulation in Sec. V.B, and the local optimality of the PISC solution can be verified.

IV. MPC Implementation with Guaranteed Recursive Feasibility

Based on the PISC algorithm, this section studies the landing guidance problem under parameter uncertainties and external disturbances. In contrast to another modern guidance methodology (i.e., feasible or optimal trajectory generation (online or offline) and reference trajectory tracking [34,35]), in this paper, the PISC algorithm is embedded into the MPC framework, and the optimal control variables produced by PISC are directly used as guidance commands. Similar methods have been studied for spacecraft attitude control and reentry guidance problems based on PS methods and nonlinear programming (NLP) algorithms [5,6,36,37]. The current research is evolved from these works, although it differs mainly in two points. First, a pseudospectral-convex algorithm is adopted instead of NLP-based algorithms to enhance the computational performance. Second, and more important, different from the classic serial trajectory optimization implementation manner, a novel parallel implementation strategy is developed. Its purpose is to guarantee the feasibility of MPC updates as well as to improve the performance and robustness of the guidance algorithm, which is a main contribution of the paper.

By executing the trajectory optimization recursively at a relatively high frequency using the most current system states and feeding the rocket the most recently updated optimal guidance commands, an implicit closed-loop feedback is formed. Thus, the guidance commands are optimal and robust under constraints and uncertainties/disturbances. Under such guidance implementation, because the uncertainties and disturbances are implicitly dealt by trajectory optimization updates, the computational tractability and recursive feasibility of the trajectory optimization are premises of exploiting the advantages of MPC and guaranteeing the stability of the landing guidance. The computational tractability is provided by the rapid and deterministic convergence properties of PISC. To guarantee the recursive feasibility, the PFGMPG algorithm is proposed based on a relaxed-PISC algorithm.

In the rest of this section, the PFGMPG algorithm is constructed, the recursive feasibility of the guidance algorithm is proved, and the boundedness of the guidance error is analyzed.

A. Parallel Feasibility-Guaranteed Model Predictive Guidance Algorithm

First, we reformulate CP0 to accommodate the MPC implementation.

MPC problem 0 (MP0):

$$\begin{aligned} \min \quad & J[\bar{\boldsymbol{x}}(\cdot), \boldsymbol{u}(\cdot), \mathcal{T}_i] = -m_f \\ \text{subject to} \quad & \left.\begin{aligned} \dot{\bar{\boldsymbol{x}}}(t) &= \boldsymbol{f}(\bar{\boldsymbol{x}}(t), \boldsymbol{u}(t); \boldsymbol{p}_0) \\ \boldsymbol{c}(\bar{\boldsymbol{x}}(t), \boldsymbol{u}(t)) &\in \boldsymbol{C}, \boldsymbol{u}(t) \in \boldsymbol{U} \end{aligned}\right\} \forall t \in [t_i, \mathcal{T}_i] \\ & \bar{\boldsymbol{x}}(t_i) = \underline{\boldsymbol{x}}(t_i), \bar{\boldsymbol{x}}(\mathcal{T}_i) \in \boldsymbol{X}_f \end{aligned} \tag{21}$$

where $\bar{\boldsymbol{x}}(t)$ is the nominal state obeying the nominal dynamics used in the optimization, $\boldsymbol{p}_0$ is the nominal system parameter, $i \in \mathbb{Z}^+$ is the MPC update index, t_i is the update sampling time, $\mathcal{T}_i$ is the terminal flight time, $\boldsymbol{c}(\bar{\boldsymbol{x}}(t), \boldsymbol{u}(t)) \in \boldsymbol{C}$ denotes the path constraints, $\boldsymbol{u}(t) \in \boldsymbol{U}$ denotes the control constraints, and $\underline{\boldsymbol{x}}(t_i)$ is the actual rocket state sampled at time t_i. As in classic MPC methods, the initial state of the optimization is set as the current sampling state $\bar{\boldsymbol{x}}(t_i) = \underline{\boldsymbol{x}}(t_i)$, and $\bar{\boldsymbol{x}}(\mathcal{T}_i) \in \boldsymbol{X}_f$ denotes the terminal constraints.

There are two types of possibilities that would cause MP0 to be infeasible. One is that the problem is physically infeasible, i.e., it is impossible to reach the target from the current state without violating any constraints. The other is that the problem is physically feasible but an irrational choice of initial trajectory leads to numerical infeasibility. For the initial trajectory, it is natural to use the last updated results as the initial trajectory of the current optimization in MPC. However, due to the effects of parameter uncertainties and disturbances, the actual system state $\underline{\boldsymbol{x}}(t_i)$ may deviate from the nominal state $\bar{\boldsymbol{x}}_{i-1}(t_i)$ of the $(i-1)$th optimization results at the ith sampling time. Thus, if the $(i-1)$th updated results are taken as the initial trajectory of the ith optimization update, the initial constraint $\bar{\boldsymbol{x}}(t_i) = \underline{\boldsymbol{x}}(t_i)$ in MP0 may not be directly satisfied. Many researchers have suggested that the most recent feasible optimal trajectory can be used when the current optimization is infeasible [38]. However, the advantage of MPC in using the latest state information would be lost in this case.

Inspired by the ideas of tube-MPC [27,38] and the "soft constraints/penalty functions" methodology [39], which are widely used in trajectory optimization and MPC practice, we propose a new trajectory optimization problem, called RMP0, to address the previously stated infeasibility issue.

Relaxed MPC problem 0 (RMP0):

$$\begin{aligned} \min \quad & J[\bar{\boldsymbol{x}}(\cdot), \boldsymbol{u}(\cdot), \mathcal{T}_i] = -\omega_1 \cdot m_f + \omega_2 \|\underline{\boldsymbol{x}}(t_i) - \bar{\boldsymbol{x}}(t_i)\| \\ \text{subject to} \quad & \left.\begin{aligned} \dot{\bar{\boldsymbol{x}}}(t) &= \boldsymbol{f}(\bar{\boldsymbol{x}}(t), \boldsymbol{u}(t); \boldsymbol{p}_0) \\ \boldsymbol{c}(\bar{\boldsymbol{x}}(t), \boldsymbol{u}(t)) &\in \boldsymbol{C}, \boldsymbol{u}(t) \in \boldsymbol{U} \end{aligned}\right\} \forall t \in [t_i, \mathcal{T}_i] \\ & |\bar{\boldsymbol{x}}(t_i) - \underline{\boldsymbol{x}}(t_i)| \le \boldsymbol{\pi}_x, \bar{\boldsymbol{x}}(\mathcal{T}_i) \in \boldsymbol{X}_f \end{aligned} \tag{22}$$

where $|\bar{\boldsymbol{x}}(t_i) - \underline{\boldsymbol{x}}(t_i)| \le \boldsymbol{\pi}_x$ is the soft initial constraint, and ω_1 and ω_2 are penalty parameters.

There are two differences between RMP0 and MP0: the initial constraint and the cost function. In RMP0, the initial constraint is relaxed to ensure the feasibility of the optimization when the last updated results are used as the initial trajectory. The initial nominal state can be in a small range around the actual state, where the range $\boldsymbol{\pi}_x$ is an estimate of the upper bound of the deviation between the actual state and the $(i-1)$th updated nominal state at t_i, and its value can be estimated offline (the estimation of $\boldsymbol{\pi}_x$ will be discussed in Sec. IV.C). For the cost function, the term $\omega_2 \|\underline{\boldsymbol{x}}(t_i) - \bar{\boldsymbol{x}}(t_i)\|$ is a penalty for the state deviation, and it ensures that the initial nominal state is as close as possible to the actual trajectory.

The purpose of designing RMP0 is to introduce the most current sampling state information into the trajectory optimization as much as possible under the premise of ensuring feasibility. Thus, the feasibility issue of using the last updated results as the initial trajectory is circumvented. The feasibility proof of RMP0 is provided in Sec. IV.B.

According to numerical experiments, when the problem is not physically infeasible and the penalty parameters are well designed, the initial state $\bar{\boldsymbol{x}}(t_i)$ obtained by RMP0 would converge to a range very close to the actual state and, in many instances, we can get $\bar{\boldsymbol{x}}(t_i) = \underline{\boldsymbol{x}}(t_i)$ in a numerical sense.

In the process of solving MP0 and RMP0, the same measures discussed in Sec. III are adopted to obtain the convex subproblems linearized MPC subproblem (LMP) and linearized and relaxed MPC subproblem (LRMP). The second-order term in the cost function can be addressed by variable substitution [9]. Hereafter, the PISC algorithm using subproblem LMP is referred to as standard-PISC (S-PISC) and the algorithm using LRMP is referred to as relaxed-PISC (R-PISC).

It should be noted that, although R-PISC has guaranteed feasibility, S-PISC has better performance. The additional penalty term in R-PISC increases the complexity of the algorithm; thereby, the time consumption of every iteration will be increased. Moreover, to obtain a satisfactory $\|\bar{\boldsymbol{x}}(t_i) - \underline{\boldsymbol{x}}(t_i)\|$ value, several more iterations are often required for R-PISC than for S-PISC. That is, R-PISC cannot simply replace S-PISC, but the algorithms compensate for each other's shortcomings.

Based on the S-PISC and R-PISC algorithms, we now construct the PFGMPG algorithm within the MPC framework. The basic structure of PFMPG is shown in Fig. 2. During each update, PFGMPG executes the S-PISC and R-PISC algorithms in parallel, with S-PISC as the primary algorithm and R-PISC as the backup algorithm. The computation of S-PISC is usually faster than that of R-PISC. If S-PISC finishes first and the solution is feasible, then the loop is exited, the solution is employed to update the database, and the current update is set as finished. If S-PISC does not finish first or if

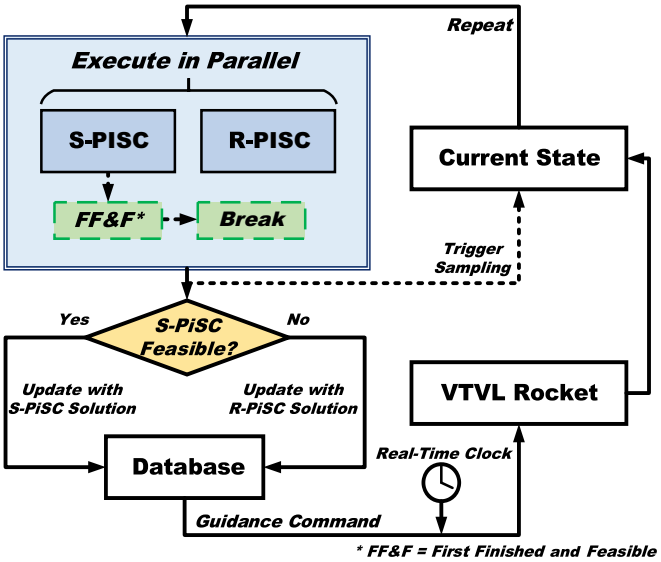

Fig. 2 Algorithm structure of PFGMPG.

S-PISC is infeasible, then the update is finished when the computations of both S-PISC and R-PISC are finished. Then, if S-PISC is feasible, its solution is employed; otherwise, the solution of R-PISC with guaranteed feasibility is employed. Simultaneously, the current state information is sampled and the next optimization computation is started. Meanwhile, during this computation, the guidance commands stored in the database are applied to the rocket according to the timing.

Because there is no data interaction between the S-PISC and R-PISC algorithms, the PFGMPG algorithm is particularly suitable for multicore processors. When taking full advantage of the characteristics of multicore parallel computing, the time consumption of one update cycle of PFGMPG would be nearly equivalent to that of implementing S-PISC alone [40]. At present, parallel computing is already a mature technology, and multicore processors have been successfully used in aerospace trajectory optimization [41] and robot motion planning [42].

Now, we present the PFGMPG algorithm as follows:

Parallel feasibility-guaranteed model predictive guidance algorithm

Initialization. Save initial trajectory to database, set deviation bound $\boldsymbol{\pi}_x$, set penalty parameters ω_1 and ω_2, set parameters needed by PISC, and set update index $i = 1$.

while $(\mathcal{T}_{i-1} - t_{i-1}) > 0$
  Generate guidance commands according to database and real-time clock;
  Sample the current system state $\underline{\boldsymbol{x}}(t_i)$;
  Run **S-PISC** and **R-PISC** in parallel with $\underline{\boldsymbol{x}}(t_i)$;
  if S-PISC obtains solution first and the solution is feasible
    Save the results of **S-PISC** to database, and set $i = i + 1$;
    break;
  else, if the update finished and **S-PISC** is feasible
    Save the results of **S-PISC** to database, and set $i = i + 1$;
  else
    Save the results of **R-PISC** to database, and set $i = i + 1$;
  end if
end while

Remark 2 (stability): With respect to the stability of the actual trajectory, because the final state is bounded in the optimization, the trajectory is guaranteed to converge to a neighborhood of the desired terminal set as long as either S-PISC or R-PISC is feasible in the final update and the deviation between the actual and nominal trajectories is bounded. In the following two subsections, we will prove the recursive feasibility of the guidance algorithm and present an analysis of the deviation bound between the actual and nominal trajectories. With the guarantee of recursive feasibility and the boundedness of guidance error, PFGMPG is practically stable.

B. Recursive Feasibility of PFGMPG

Recursive feasibility is one of the most important properties of PFGMPG, and it signifies that, as long as the trajectory optimization problem is feasible at the initial time, the problem is feasible at every update time during the guidance process. Only when recursive feasibility is guaranteed can the stability condition of PFGMPG be satisfied and the expected performance of the algorithm be achieved. Now, we present the proof of the recursive feasibility of PFGMPG.

Proposition 1 (recursive feasibility of PFGMPG): Suppose that the trajectory optimization problem is feasible at t_0 with terminal flight time $\mathcal{T}_0$, and let t_i for $i \in \mathbb{Z}^+$ be the time at which PFGMPG is updated. The feasibility of PFGMPG is guaranteed at any t_i.

Proof: If S-PISC is feasible at t_i for all $i \in \mathbb{Z}^+$, then PFGMPG is recursively feasible. At any t_i, if S-PISC is not feasible, then we prove that R-PISC is feasible. The idea is to show the solution of the $(i-1)$th optimization is a feasible solution of the ith optimization for R-PISC.

Suppose S-PISC or R-PISC is feasible at t_{i-1} and provides optimal solution $\bar{\boldsymbol{x}}^*_{i-1}(t), \boldsymbol{u}^*_{i-1}(t)$ for $t \in [t_{i-1}, \mathcal{T}^*_{i-1}]$. For the ith optimization, set the initial trajectory as follows:

$$\mathcal{T}_i = \mathcal{T}^*_{i-1}, \quad \boldsymbol{u}_i(t) = \boldsymbol{u}^*_{i-1}(t), \quad \bar{\boldsymbol{x}}_i(t) = \bar{\boldsymbol{x}}^*_{i-1}(t), \quad \forall\, t \in [t_i, \mathcal{T}_i] \tag{23}$$

Then, the path constraints, control constraints, and terminal constraints in the ith optimization are satisfied directly for R-PISC because the solution of the $(i-1)$th optimization is feasible.

For the initial condition, $\boldsymbol{\pi}_x$ is the upper bound between the actual trajectory and the $(i-1)$th nominal trajectory during $t \in [t_{i-1}, \mathcal{T}^*_{i-1}]$, which ensures $|\bar{\boldsymbol{x}}^*_{i-1}(t_i) - \underline{\boldsymbol{x}}(t_i)| \le \boldsymbol{\pi}_x$. Thus, the state $\bar{\boldsymbol{x}}_i(t_i) = \bar{\boldsymbol{x}}^*_{i-1}(t_i)$ is feasible for the initial condition $|\bar{\boldsymbol{x}}(t_i) - \underline{\boldsymbol{x}}(t_i)| \le \boldsymbol{\pi}_x$ of R-PISC.

Therefore, R-PISC is feasible for all $i \in \mathbb{Z}^+$. Because S-PISC and R-PISC are solved in parallel, there is at least one optimization algorithm that can obtain a feasible solution in every update. Hence, PFGMPG is recursively feasible.

C. Guidance Error Under Parameter Uncertainties and External Disturbances

To construct the practical stability and to uncover more features of the guidance algorithm, in this subsection, we analyze the characteristics of the guidance error of PFGMPG.

Benefitting from the MPC implementation and the guarantee of recursive feasibility, for physically feasible guidance applications, a new nominal trajectory is obtained in each update with the initial state being identical or very close to the actual state; thus, the guidance error does not propagate between update cycles. Although, during each update cycle, the trajectory deviation is related to the system dynamical characteristics, the size of the uncertainties/disturbances, and especially the frequency of the optimization update.

Here, we employ the methodology used in Ref. [5] to analyze the guidance error during an update cycle. A Carathéodory-π trajectory-based optimal feedback control method, which is similar to the MPC framework used in this paper, was studied in Ref. [5].

The nominal dynamics model used in the trajectory optimization is formulated in Eq. (21), and the actual system dynamics can be represented by the following:

$$\dot{\underline{\boldsymbol{x}}}(t) = \boldsymbol{f}(\underline{\boldsymbol{x}}(t), \boldsymbol{u}(t); \boldsymbol{p}) + \boldsymbol{\xi}(t) \tag{24}$$

where $\boldsymbol{p}$ is the actual system parameter, and $\boldsymbol{\xi}(t)$ is the disturbance that the system suffers during the flight.

The discretization and linearization errors in the trajectory optimization can also be included in $\boldsymbol{p}$. In practice, the effects of these errors are relatively small as compared with those of the uncertainties and disturbances, but they are nonnegligible; this will be analyzed later using numerical experiments. Here, we only consider the discretization and linearization errors as special types of parameter uncertainties.

In addition to the effects of parameter uncertainties and external disturbances, the computational delay of the optimization affects the guidance performance [5]. That is, the ith updated guidance commands based on the information sampled at t_i can only act on the system at the $(i+1)$th sampling time with a time delay Δt_i, where Δt_i is the time consumption of the ith trajectory optimization.

Similar to Ref. [5], we make some assumptions about the uncertainties, disturbances, and system dynamics as follows:

1) There exists $\varepsilon_{p,\xi} > 0$ such that $\|\boldsymbol{p} - \boldsymbol{p}_0\| \le \varepsilon_p$ and $\|\boldsymbol{\xi}\|_{L\infty} \le \varepsilon_\xi$.

2) Function $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}; \boldsymbol{p})$ is Lipschitz continuous, i.e., there exist positive constants $\mathrm{Lip}f_{x,u,p}$ such that

$$\|\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}; \boldsymbol{p}) - \boldsymbol{f}(\boldsymbol{y}, \boldsymbol{v}; \boldsymbol{q})\| \le \mathrm{Lip}f_x\|\boldsymbol{x} - \boldsymbol{y}\| + \mathrm{Lip}f_u\|\boldsymbol{u} - \boldsymbol{v}\| + \mathrm{Lip}f_p\|\boldsymbol{p} - \boldsymbol{q}\|$$

We first analyze the situation in which S-PISC is feasible. During the ith update cycle, under guidance command $[t_i, t_{i+1}] \to \boldsymbol{u}^*(t, \underline{\boldsymbol{x}}(t_i))$, the nominal state at t_{i+1} is given by the following:

$$\bar{\boldsymbol{x}}(t_{i+1}) = \underline{\boldsymbol{x}}(t_i) + \int_{t_i}^{t_{i+1}} \boldsymbol{f}(\bar{\boldsymbol{x}}(t), \boldsymbol{u}^*(t, \underline{\boldsymbol{x}}(t_i)); \boldsymbol{p}_0)\,\mathrm{d}t \tag{25}$$

whereas the actual state at t_{i+1} is given by the following:

$$\underline{\boldsymbol{x}}(t_{i+1}) = \underline{\boldsymbol{x}}(t_i) + \int_{t_i}^{t_{i+1}} \boldsymbol{f}(\underline{\boldsymbol{x}}(t), \boldsymbol{u}^*(t, \underline{\boldsymbol{x}}(t_{i-1})); \boldsymbol{p})\,\mathrm{d}t + \int_{t_i}^{t_{i+1}} \boldsymbol{\xi}(t)\,\mathrm{d}t \tag{26}$$

Thus, according to the aforementioned assumptions, the state deviation at t_{i+1} can be expressed as follows:

$$\begin{aligned}\|\underline{\boldsymbol{x}}(t_{i+1}) - \bar{\boldsymbol{x}}(t_{i+1})\| \le{}& \mathrm{Lip}f_x \cdot \int_{t_i}^{t_{i+1}} \|\underline{\boldsymbol{x}}(t) - \bar{\boldsymbol{x}}(t)\|\mathrm{d}t \\ &+ \mathrm{Lip}f_u \cdot \int_{t_i}^{t_{i+1}} \|\boldsymbol{u}^*(t, \underline{\boldsymbol{x}}(t_{i-1})) - \boldsymbol{u}^*(t, \underline{\boldsymbol{x}}(t_i))\|\mathrm{d}t \\ &+ \mathrm{Lip}f_p \cdot \|\boldsymbol{p} - \boldsymbol{p}_0\| \cdot \Delta t_i + \|\boldsymbol{\xi}\|_{L\infty} \cdot \Delta t_i\end{aligned} \tag{27}$$

According to Bellman's principle of optimality, if the system is accurately modeled and ideally undisturbed during $[t_{i-1}, t_i]$, then $\boldsymbol{u}^*(t, \underline{\boldsymbol{x}}(t_i)) \equiv \boldsymbol{u}^*(t, \underline{\boldsymbol{x}}(t_{i-1}))$ for $t \in [t_i, \mathcal{T}_i]$. In actual guidance situations, under the preceding assumptions and for limited Δt_{i-1}, we can deduce that $\|\boldsymbol{u}^*(t, \underline{\boldsymbol{x}}(t_i)) - \boldsymbol{u}^*(t, \underline{\boldsymbol{x}}(t_{i-1}))\|_{L_\infty} \le \varepsilon_{u,i}$, where $\varepsilon_{u,i}$ is a measure of the effects of parameter uncertainties and disturbances on the system during $[t_{i-1}, t_i]$, i.e., $\varepsilon_{u,i} = d(\boldsymbol{p}, \boldsymbol{d}, \Delta t_{i-1})$. Then, we have the following:

$$\begin{aligned}\|\underline{\boldsymbol{x}}(t_{i+1}) - \bar{\boldsymbol{x}}(t_{i+1})\| \le{}& \mathrm{Lip}f_x \cdot \int_{t_i}^{t_{i+1}} \|\underline{\boldsymbol{x}}(t) - \bar{\boldsymbol{x}}(t)\|\,\mathrm{d}t \\ &+ \mathrm{Lip}f_u \cdot \varepsilon_{u,i} \cdot \Delta t_i + \mathrm{Lip}f_p \cdot \varepsilon_p \cdot \Delta t_i + \varepsilon_\xi \cdot \Delta t_i\end{aligned} \tag{28}$$

From Gronwall's lemma [5,43], Eq. (28) can be reduced to the following:

$$\begin{aligned}\|\underline{\boldsymbol{x}}(t_{i+1}) - \bar{\boldsymbol{x}}(t_{i+1})\| \le{}& (\mathrm{Lip}f_u \cdot \varepsilon_{u,i} \cdot \Delta t_i + \mathrm{Lip}f_p \cdot \varepsilon_p \cdot \Delta t_i + \varepsilon_\xi \cdot \Delta t_i) \\ &\cdot \exp(\mathrm{Lip}f_x \cdot \Delta t_i)\end{aligned} \tag{29}$$

Set $K = \mathrm{Lip}f_u \cdot \varepsilon_{u,i} + \mathrm{Lip}f_p \cdot \varepsilon_p + \varepsilon_\xi$. Then, we have:

$$\|\underline{\boldsymbol{x}}(t_{i+1}) - \bar{\boldsymbol{x}}(t_{i+1})\| \le K \cdot \Delta t_i \cdot \exp(\mathrm{Lip}f_x \cdot \Delta t_i) \tag{30}$$

According to Eq. (30), the state deviation between the actual and nominal trajectories is bounded, and the upper bound is related to the characteristics of the system dynamics, the magnitude of the uncertainties/disturbances, and the frequency of the optimization update. As long as the assumptions are satisfied, we have $\|\underline{\boldsymbol{x}}(t_{i+1}) - \bar{\boldsymbol{x}}(t_{i+1})\| \to 0$ as $\Delta t_i \to 0$; and the deviation bound decreases faster than an exponential rate as Δt_i decreases. Thus, the robustness and accuracy of the guidance algorithm can be significantly improved as the computation of trajectory optimization accelerates, which in turn supports the necessities of adopting the PISC algorithm and designing the varying pattern of the number of discretization points.

Equation (30) can be used to estimate the upper bound of the deviation between the actual and nominal states during an update cycle. The Lipschitz constants $\mathrm{Lip}f_{x,u,p}$, the maximum computation consumption $\Delta t_{\max}$, and the maximum effect of disturbances on the control trajectory $\varepsilon_{u,\max}$ can be estimated by offline numerical simulations. For given ε_p and ε_ξ, we can set

$$\boldsymbol{\pi}_x = (\mathrm{Lip}f_u \cdot \varepsilon_{u,\max} + \ \mathrm{Lip}f_p \cdot \varepsilon_p + \varepsilon_\xi) \cdot \Delta t_{\max} \cdot \exp(\mathrm{Lip}f_x \cdot \Delta t_{\max})$$

Next, we analyze the situation in which S-PISC is infeasible and the solution of R-PISC is adopted. Let $\boldsymbol{\delta}_x(t_i)$ be the initial deviation between the actual trajectory and the ith updated nominal trajectory obtained by R-PISC. Then, during the ith update cycle, under guidance command $[t_i, t_{i+1}] \to \boldsymbol{u}^*(t, \underline{\boldsymbol{x}}(t_i))$, the nominal state at t_{i+1} is given by the following:

$$\bar{\boldsymbol{x}}(t_{i+1}) = \underline{\boldsymbol{x}}(t_i) + \boldsymbol{\delta}_x(t_i) + \int_{t_i}^{t_{i+1}} \boldsymbol{f}(\bar{\boldsymbol{x}}(t), \boldsymbol{u}^*(t, \underline{\boldsymbol{x}}(t_i)); \boldsymbol{p}_0)\,\mathrm{d}t \tag{31}$$

The actual state at t_{i+1} is the same as that in Eq. (26). By the same derivation process as described previously, we have the following:

$$\|\underline{\boldsymbol{x}}(t_{i+1}) - \bar{\boldsymbol{x}}(t_{i+1})\| \le (\|\boldsymbol{\delta}_x(t_i)\| + K \cdot \Delta t_i) \cdot \exp(\mathrm{Lip}f_x \cdot \Delta t_i) \tag{32}$$

As stated earlier, $\|\boldsymbol{\delta}_x(t_i)\|$ is generally very close to zero and $\|\boldsymbol{\delta}_x(t_i)\| \ll \|\boldsymbol{\pi}_x(t_i)\|$ for physically feasible applications. Thus, the guidance error of PFGMPG is bounded even when S-PISC is infeasible.

For the guidance accuracy, as stated earlier, the guidance error does not propagate between update cycles for physically feasible applications. Additionally, by using the varying pattern of the number of discretization points in Eq. (18), with the shortening of the time domain of optimization, the optimization computation accelerates in each update. Thus, according to Eqs. (30) and (32), the guidance accuracy is gradually improved during the flight.

V. Numerical Demonstrations and Analyses

In this section, numerical results are presented to demonstrate the algorithms developed in Secs. III and IV, and the performance of the algorithms are analyzed in detail. First, nominal trajectory optimizations are performed to verify the effectiveness of the PISC algorithm. Then, basic properties of the PFGMPG algorithm are demonstrated by nominal guidance simulations. Finally, guidance simulations under parameter uncertainties and wind disturbances are performed. The rocket parameters used in this paper are as follows: $m_0 = 35{,}000$ kg, $m_{\mathrm{dry}} = 25{,}000$ kg, $T_{\min} = 300$ kN, $T_{\max} = 860$ kN, $I_{\mathrm{sp}} = 400$ s, $S_{\mathrm{ref}} = 10.75$ m^2, $C_{D0} = 0.35$, $C_D^\alpha = 0.02376$, $C_D^{\alpha^2} = 0.01125$, $C_L^\alpha = 0.01877$, and $C_L^{\alpha^2} = 0.00968$. The bound values of the path and control constraints are as follows: $q_{\max} = 55{,}000$ N/m^2, $\dot{Q}_{\max} = 3500$ kW/m^2, $n_{\max} = 1$, $\alpha_{\max} = 10$ deg, $\dot{\alpha}_{\max} = 1$ deg/s, $\dot{T}_x^{\max} = 3000$ kN/s, $\dot{T}_{y,z}^{\max} = 500$ kN/s, and $\eta_{y,z}^{\max} = 20$ deg. The initial values and terminal constraints of the states are presented in Table 1.

All numerical optimizations and simulations in this paper are performed on a laptop with an Intel Core i7-7700HQ 2.80 GHz CPU. The results in Sec. V.A are obtained using MATLAB 2012a (for reasonable comparison between PISC and GPOPS-II [44]), and the results in Secs. V.B and V.C are obtained using Visual Studio 2015 with C++ language (for higher efficiency and convenience of invoking multicore processing). The convex subproblems produced by the algorithms are solved using the MOSEK software (MATLAB API and C API, respectively) [32].

A. Performance of the PISC Algorithm

In this subsection, the performance of the PISC algorithm is analyzed, and two advantages of this algorithm (i.e., high computation speed and high solution accuracy) are demonstrated via comparison. Optimization experiments are performed using PISC, GPOPS-II, and a trapezoidal-discretization-based improved successive convexification (TISC) algorithm. GPOPS-II is an optimization software package of the hp-adaptive Radau pseudospectral discretization method and general NLP methods. The results obtained by GPOPS-II can be seen as benchmarks to evaluate PISC. TISC is the same as PISC except that trapezoidal discretization is adopted as in Refs. [15,17], and it is used to demonstrate the benefit of using the PS method from the perspective of discretization accuracy.

In PISC, the total number of discretization points is set to 100: there are 20 points in the coast phase and {30, 20, 30} points in the subphases of the powered phase. In GPOPS-II, hp-adaptive mesh

Table 1 Initial values and terminal constraints of the states

Variable	Initial	Terminal
V, m/s	400	[20.0, 22.0]
θ, deg	−55	[−90.5, −89.5]
σ, deg	0	[−0.5, 0.5]
r_x, m	−6500	[−1.0, 1.0]
r_y, m	16500	[50.0, 52.0]
r_z, m	−50	[−1.0, 1.0]

refinement is carried out, and the SNOPT solver is used. The final number of discretization points of GPOPS-II after automatic mesh refinement is 88 (which is fewer than, but comparable to, that of PISC). In TISC, for simplicity, the ignition and terminal times are fixed, the time values are set to the results obtained by GPOPS-II, and the number of discretization points is also set to 100. The convergence tolerance is set to 10^{-5}, and the same coarse initial trajectory is used in the three algorithms.

The optimized trajectories are shown in Figs. 3–5. Here, the thrust vector is displayed in the thrust magnitude/pointing angle form. It can be seen that the thrust magnitude has a significant bang–bang characteristic. For the subphases of PISC, one subphase is in the coast–arc and the other two are successively in the bang–arc. The terminal masses are 29,397.8 kg for PISC, 29,523.3 kg for GPOPS-II, and 29,288.5 kg for TISC. All constraints are satisfied using the three algorithms.

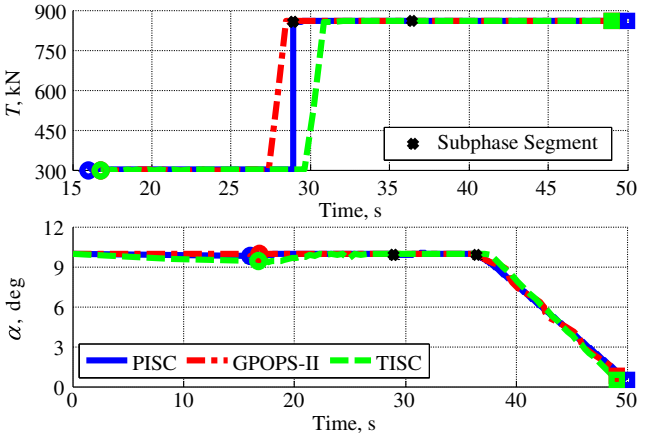

Fig. 3 Thrust and AOA histories of the three algorithms.

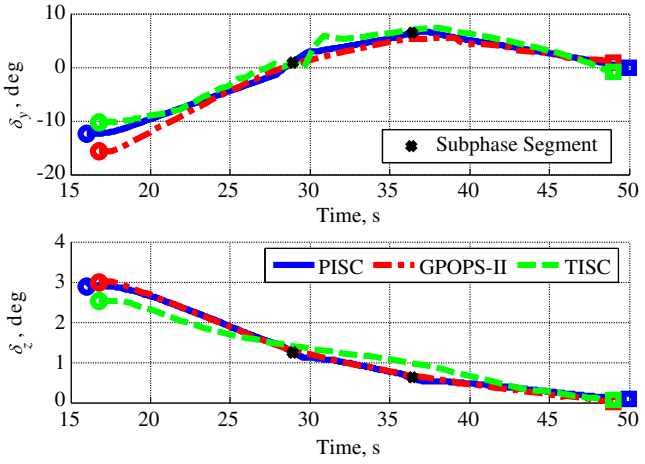

Fig. 4 Pointing angle histories of the three algorithms.

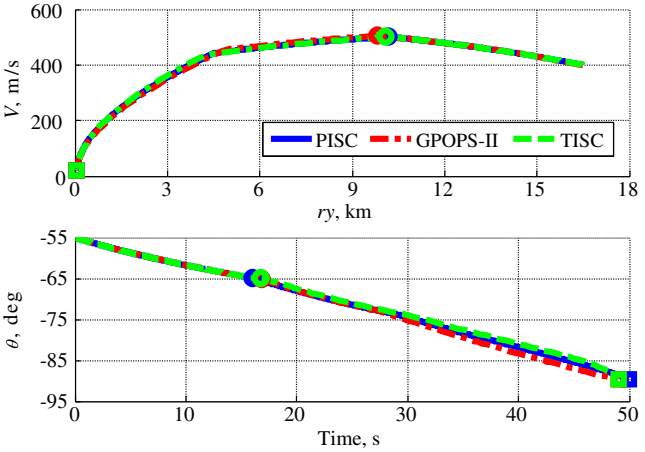

Fig. 5 Height–velocity profiles and flight-path angle histories of the three algorithms.

Table 2 Comparison of the propagation results

Algorithm	δ_{V_f}, m/s	δ_{θ_f}, deg	δ_{σ_f}, deg	$\delta_{r_{xf}}$, m	$\delta_{r_{zf}}$, m	δ_{m_f}, kg
PISC	2.443	0.285	0.021	15.349	0.992	12.677
GPOPS-II	1.925	0.263	0.014	11.114	0.634	10.343
TISC	46.794	1.916	0.355	132.682	15.341	86.852

As shown in the figures, the solutions of the three algorithms are very similar, whereas the average deviation of the control variables between PISC and GPOPS-II is 3.22%, and that between TISC and GPOPS-II is 6.87%. The CPU times are 4.42 s for PISC, 86.51 s for GPOPS-II, and 6.03 s for TISC. The time consumption of PISC is only 5.11% of that of GPOPS-II, even though there are fewer discretization points used in GPOPS-II. Similar result patterns can be obtained using different parameter settings. That is, for the rocket landing problem, the general effectiveness of PISC is comparable to that of GPOPS-II. Moreover, the well-tuned PISC algorithm is more efficient than general NLP-based methods due to the advantages of convex optimization theory.

To demonstrate the feasibility and accuracy of the PISC solution, we extract the optimized control trajectories obtained by the three algorithms and use them to propagate the nonlinear dynamical equations in Eq. (1). The fourth-order Runge–Kutta method and a 0.01 s time step are adopted. The termination condition of propagation is the propagated height reaching 50 m. The deviations between the optimized terminal state and the propagated terminal state are shown in Table 2. It can be seen that the solution accuracies of PISC and GPOPS-II are acceptable and comparable, and GPOPS-II outperforms PISC slightly. In contrast, the accuracy of TISC is considerably lower than those of the other two algorithms. Because the major difference between PISC and TISC is the discretization method, we can conclude that, for the same problem size, solution accuracy can be significantly improved by using the PS method.

We further analyze the results of GPOPS-II and PISC. The deviation of GPOPS-II is due to the discretization error, and the deviation of PISC is due to the discretization error and the linearization error. Because the numbers of discretization points used in PISC and GPOPS-II are comparable (100 vs 88), it is deduced from Table 2 that the linearization error is an order of magnitude lower than the discretization error in PISC under the current parameter settings. When the convergence tolerance is set to smaller values, the effect of the linearization error on PISC is weakened at the expense of more iterations and a longer CPU time.

B. Performance of the PFGMPG Algorithm

In this subsection, the basic performance of the PFGMPG algorithm is demonstrated and analyzed under ideal conditions: that is, without parameter uncertainties or disturbances. Simulations with uncertainties and disturbances are discussed in the next subsection. We use the OpenMP API [45] and Visual Studio IDE to develop the parallel algorithm. In each MPC update, two threads are set to perform the S-PISC and R-PISC algorithms on different CPU cores. The termination condition of guidance is the flight height reaching 50 m. The χ value in Eq. (18) is set to 0.7. The values of $\boldsymbol{\pi}_x$ used in R-PISC are designed according to trajectory simulations and rocket dynamics, and they are listed in Table 3.

From Sec. IV and the preceding subsection, even when there is no parameter uncertainty or disturbance, there exist inherent methodological errors in the PFGMPG algorithm: that is, the discretization error and the linearization error introduced by the PISC algorithm. In spite of this, the error magnitudes are sufficiently small (as shown in Table 2) because PISC is well designed so that they can be easily handled by the PFGMPG algorithm. The guidance results

Table 3 Estimated deviation bound values

Maximum deviation	Estimated value
π_V, m/s	8.5
π_θ, deg	2.0
π_σ, deg	0.8
π_{r_x}, m	15.0
π_{r_y}, m	25.0
π_{r_z}, m	7.0
π_m, kg	150.0

are shown in Figs. 6–13. All constraints are satisfied, and the terminal states are within the terminal feasible set (TFS) of PISC, which is quite narrow. Moreover, the performance index (also known as the the terminal mass) is 29,489.7 kg, which is enhanced when compared with the result of single PISC trajectory optimization.

Figures 6 and 7 illustrate the CPU time consumption and the number of PISC iterations in each trajectory optimization update during the guidance process. There are 46 updates in total: four in the coast phase, and 42 in the powered phase. S-PISC is feasible in every update because the effects of the discretization and linearization errors are small. Because S-PISC and R-PISC are performed in parallel, the CPU time is slightly longer than the cases in which only S-PISC is performed. In comparison, if R-PISC is set as the primary algorithm, approximately $9 \sim 35\%$ more CPU time and $1 \sim 3$ more iterations are needed in each update, according to different initial deviation ranges.

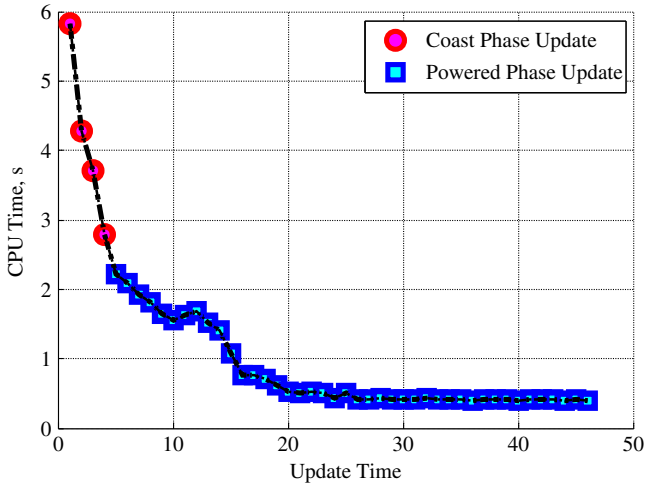

Fig. 6 CPU time consumption of every update.

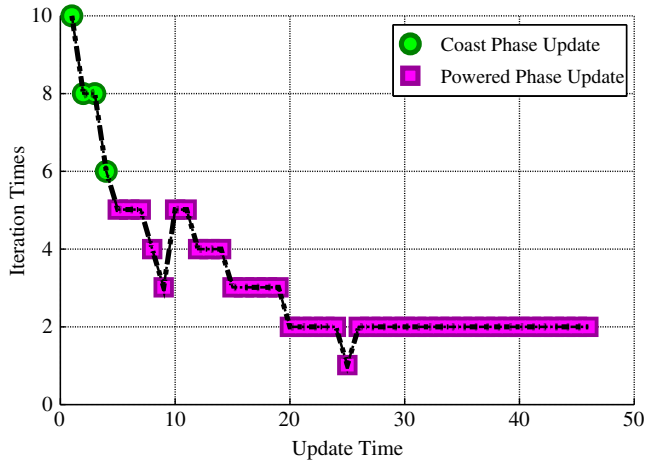

Fig. 7 Number of PISC iterations of every update.

As the guidance process progresses, the CPU time and the number of PISC iterations decrease significantly because the varying pattern of the number of discretization points shown in Eq. (18) is adopted and the quality of the initial trajectory improves during each update. Moreover, as the update interval shortens, the effect of the computational delay decreases, which further benefits the subsequent updates.

As shown in Fig. 6, the update interval remains at approximately 0.4 s since the 28th update and does not further shorten afterward. The update interval remains unchanged because the time consumption of transaction processing (such as memory reading and control command interpolation) cannot be avoided. Additionally, because we use a general IPM software, the specific structure and sparsity pattern of the problem are not fully used. Code optimization, dedicated numerical scaling and balancing [30], and the customized IPM [46] can significantly improve this situation; thus, better computational efficiency can be expected. Besides, because the initial trajectory is satisfactory, PISC requires only two iterations to converge in the final several updates, as shown in Fig. 7.

It can be noticed in Fig. 6 that the CPU time does not monotonically decrease, and a "wave crest" is observed during the 10th–15th updates. The reason is that, starting from the 10th update, the dynamic pressure suffered by the rocket gradually reaches its peak value, and the effect of the aerodynamic forces is the most significant during this period. Thus, the nonlinear effect of system dynamics is the most significant, and the CPU time of PISC is accordingly longer.

Figures 8–11 illustrate the actual control trajectories used in the guidance process and the optimal control commands obtained by every trajectory optimization update. The subplot pn the right side of

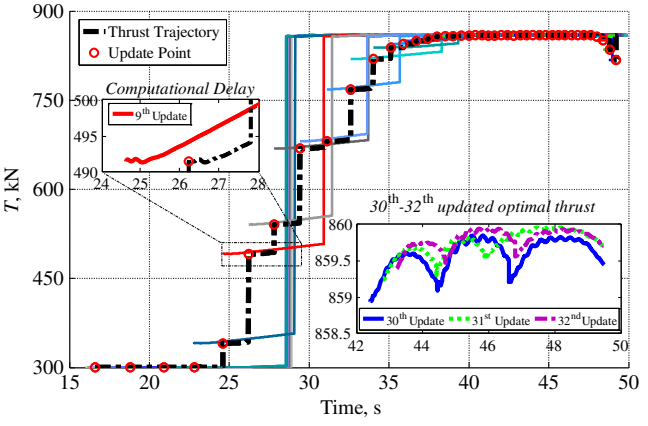

Fig. 8 Actual guidance thrust trajectory and all updated thrust commands of PFGMPG.

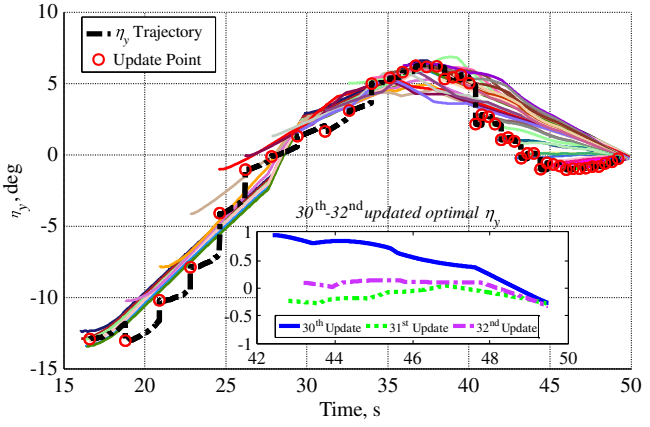

Fig. 9 Actual guidance η_y trajectory and all updated η_y commands of PFGMPG.

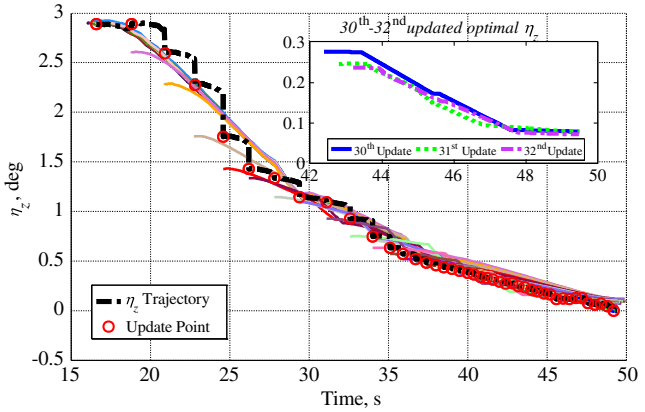

Fig. 10 Actual guidance η_z trajectory and all updated η_z commands of PFGMPG.

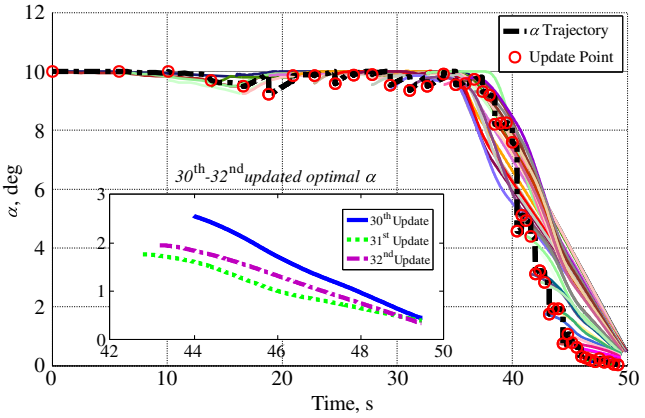

Fig. 11 Actual guidance AOA trajectory and all updated AOA coSmmands of PFGMPG.

Fig. 8 is an enlarged view of the optimized thrust commands of the 30th–32nd updates, and the left subplot shows the computational delay effect, which was discussed in Sec. IV. Figures 12 and 13 illustrate the state and dynamic pressure trajectories.

According to Figs. 8–11, under the effects of the discretization error, the linearization error, and the computational delay, the results of each update are very close but not coincident, and Bellman's principle of optimality is met in a practical sense. The coincidence of the updated control commands improves when more discretization points and smaller convergence tolerance values are used, and the actual control trajectories become smoother, at the expense of increased computational burden. Similar to the wave crest of the CPU time, the noncoincidence of the updated control commands is

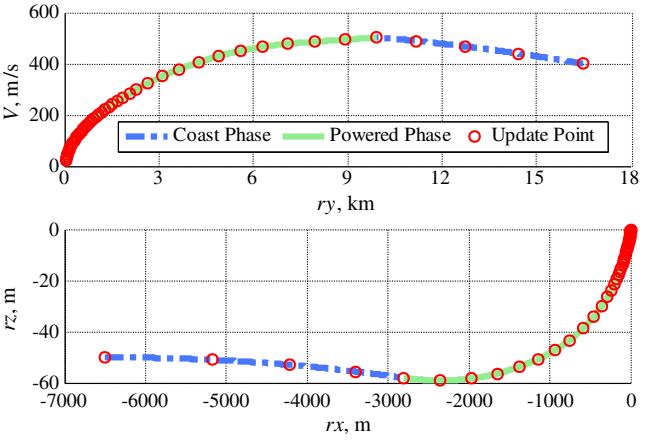

Fig. 12 Height–velocity and X-Z position profiles of PFGMPG.

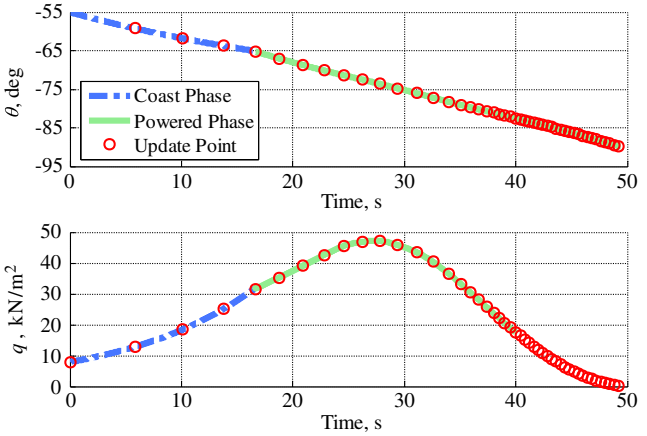

Fig. 13 Flight-path angle and dynamic pressure histories of PFGMPG.

particularly obvious during the large-dynamic-pressure phase, when the effect of nonlinearity is most significant.

Although the preceding results and discussions are obtained under ideal simulation conditions, they still apply to the scenarios in which uncertainties and disturbances exist.

Now, to further demonstrate the necessity of adopting the PS discretization method for the guidance problem, we perform a version of PFGMPG based on the TISC algorithm, wherein the optimization problem is discretized by the trapezoidal method as stated earlier. Likewise, the ignition and terminal times are set as fixed, and the time values are taken from the results of the aforementioned PISC-based guidance simulation. The values of $\boldsymbol{\pi}_x$ are set as double those shown in Table 3. Other settings are identical to the PISC-based guidance simulation.

There are 40 updates in total: four in the coast phase, and 36 in the powered phase. Because the effect of the discretization error is no longer small, S-TISC is infeasible in seven updates. The terminal mass is 29,195.3 kg, which is lower than that of the single TISC trajectory optimization. Moreover, the terminal velocity and position are without the terminal feasible set of TISC, where $\delta_{V_f} = 3.21$ m/s and $\delta_{r_{xf}} = 4.77$ m.

As verified by the preceding results, because the guidance commands are directly taken from the results of the trajectory optimization, the solution accuracy of the optimization significantly impacts the guidance performance. The larger the discretization error, the larger the initial deviation between the nominal and actual states, thus causing a poorer initial trajectory in the next update. Then, more CPU time is required for S-TISC to obtain a solution if it is feasible. As discussed in Sec. IV, during each update cycle, the guidance error increases when the update frequency decreases, which further affects the subsequent updates. As a result, the guidance accuracy is declined. Moreover, as stated in Sec. IV, the discretization error can be seen as a special parameter uncertainty, and more fuel will be consumed to compensate for this "uncertainty" effect. The guidance performance will further deteriorate when there are actual parameter uncertainties or external disturbances.

In contrast, for PISC-based PFGMPG, because the solution accuracy is significantly improved by the PS method, the aforementioned problems are greatly alleviated and the guidance performance is accordingly satisfactory.

C. Numerical Simulations of PFGMPG Under Parameter Uncertainties and Disturbances

In this subsection, we demonstrate the performance of the PFGMPG algorithm under parameter uncertainties and disturbances. First, we study the effects of aerodynamic coefficient uncertainties, i.e., the deviations in the drag coefficient, the lift coefficient, and the atmospheric density. The extreme coefficient deviation values and the guidance simulation results are shown in Table 4, where $C_D + 15\%$ indicates that the actual drag coefficient has a value of $1.15 \times C_D$ during the guidance process, and the same notations are used for the other parameter uncertainties.

Table 4 Results of PFGMPG under parameter uncertainties

Uncertainty	Terminal states	Terminal mass, kg	Number of updates	Number of times that S-PISC is infeasible
$C_D + 15\%$	$V_f = 18.3$ m/s, $r_{xf} = -2.4$ m, others within TFS	29,421.3	44	3
$C_D - 15\%$	$V_f = 23.6$ m/s, $r_{xf} = 4.1$ m, others within TFS	28,901.6	43	9
$C_L + 20\%$	All within TFS	29,492.2	45	0
$C_L - 20\%$	All within TFS	29,445.7	46	2
$\rho + 15\%$	$V_f = 17.1$ m/s, $r_{xf} = -2.9$ m, others within TFS	29,399.3	44	3
$\rho - 15\%$	$V_f = 24.6$ m/s, $r_{xf} = 5.6$ m, others within TFS	28,877.8	43	10

Table 5 Results of PFGMPG under wind disturbance

V_W, α_W, β_W (m/s, deg, deg)	Terminal states	Terminal mass, kg	Number of updates	Number of times that S-PISC is infeasible
10, 5, 5	$V_f = 19.51$ m/s, others within TFS	29432.6	46	5
−10, −5, −5	$V_f = 22.9$ m/s, $r_{xf} = 1.6$ m, All within TFS	29247.1	45	9
15, 10, 10	$V_f = 18.42$ m/s, $r_{xf} = -1.9$ m, others within TFS	29211.4	44	8
−15, −10, −10	$V_f = 24.2$ m/s, $r_{xf} = 3.4$ m, others within TFS	28734.6	45	12
20, 15, 15	$V_f = 16.3$ m/s, $r_{xf} = -3.2$ m, $\sigma_f = -0.6$ deg, others within TFS	28924.3	43	11
20, −15, −15	$V_f = 17.7$ m/s, $r_{xf} = -2.1$ m, others within TFS	28545.9	44	15
−20, −15, −15	*Infeasible since the 11th update*			
−20, 15, 15	$V_f = 38.2$ m/s, $\theta_f = -87.5$ deg, $\sigma_f = -0.72$ deg, $r_{xf} = 157.2$ m, $r_{zf} = 21.2$ m	28,289.1	40	19

Based on these results, we analyze the guidance performance under parameter uncertainties as follows:

1) The aerodynamic coefficient deviations have greater influence on guidance performance during the large-dynamic-pressure phase and the latter stage of the flight when the thrust magnitude reaches its maximum value.

During the large-dynamic-pressure phase, as the nonlinear effect of aerodynamic forces already influences the PISC algorithm, a large aerodynamic force deviation (because q is large) will enhance this effect, likely resulting in the infeasibility of S-PISC. In addition, because the thrust trajectory of the fuel-optimal landing flight has a significant bang–bang characteristic, the ability to resist parameter uncertainties by relying on thrust adjustment decreases during the latter stage of the flight, when the thrust magnitude reaches saturation. In particular, when the drag coefficient is decreased (or the atmospheric density is decreased), a portion of the mechanical energy cannot be dissipated as expected. However, when the thrust has reached saturation, no extra power can be provided to compensate for the undissipated mechanical energy. The following S-PISC is thus likely to be infeasible, and the initial deviation between the actual velocity and the nominal velocity obtained by R-PISC has difficulty converging to a value with a small magnitude. By contrast, when the drag coefficient is increased, the dissipation of the mechanical energy is faster than expected, and the state deviations can be compensated by reducing the thrust magnitude; thus, the effect on the feasibility of S-PISC is rather slight.

Besides, because the pointing direction angle commands will not be saturated during the flight, there is sufficient normal control ability to compensate for the effect of lift coefficient deviation.

2) The guidance error does not propagate between update cycles. This characteristic is discussed in Sec. IV, and it is demonstrated here via simulation. According to Table 4, although there are several updates in which S-PISC is infeasible and the values of initial deviations in R-PISC do not always converge to zero, the terminal state can always reach the neighborhood of the terminal feasible set.

3) The guidance performance is directly related to the PFGMPG update frequency. Increasing the update frequency weakens the effect of computational delay, thereby decreasing the state deviation and the number of updates in which S-PISC is infeasible. It also introduces more actual state information into the guidance algorithm, thereby increasing robustness. In summary, the higher the update frequency, the more the PFGMPG algorithm is "closed loop".

Next, we study the effect of wind disturbance on the PFGMPG algorithm. The wind does not affect the rocket ground speed and position directly, and thus the form of the system dynamics remains unchanged. However, the wind influences the rocket air speed, and thus the aerodynamic forces are reformulated as follows:

$$\begin{aligned}\bar{D} &= 0.5 \cdot \rho(V + V_w)^2 S_{\text{ref}}[C_{D0} + C_D^{\alpha}(\alpha + \widehat{\alpha}_w) + C_D^{\alpha^2}(\alpha + \widehat{\alpha}_w)^2] \\ L &= 0.5 \cdot \rho(V + V_w)^2 S_{\text{ref}}[C_L^{\alpha}(\alpha + \alpha_w) + C_L^{\alpha^2}(\alpha + \alpha_w)^2] \\ S &= 0.5 \cdot \rho(V + V_w)^2 S_{\text{ref}}[C_S^{\beta}\beta_w + C_S^{\beta^2}\beta_w^2]\end{aligned} \tag{33}$$

where V_w is the wind speed projection on the rocket ground speed vector, $\widehat{\alpha}_w = \sqrt{\alpha_w^2 + \beta_w^2}$, and α_w and β_w are the attached AOA and angle of sideslip, respectively. S is the attached lateral force, which is caused by β_w. We set $C_S^{\beta} = C_L^{\alpha}$ and $C_S^{\beta^2} = C_L^{\alpha^2}$. Here, the extreme values of V_w, α_w, and β_w are set as 20 m/s, 15 deg, and 15 deg, respectively. Table 5 shows the guidance results with various V_w, α_w, and β_w values.

Because the essence of wind disturbance lies in its effects on the aerodynamic forces, the result patterns reflected by Table 5 are basically the same as those in the preceding aerodynamic coefficient deviation scenarios. That is, the actual drag force has a greater effect on the guidance performance, the guidance error does not propagate between update cycles, and the guidance performance is directly related to the update frequency. The difference is that the effect of wind disturbance is more significant during the small-ground-speed

Table 6 Mean values and deviations of terminal states in Monte Carlo simulations

Terminal state	Mean	Standard deviation	Maximum deviation from TFS
V_f, m/s	20.5265	1.2512	4.0759
θ_f, deg	−89.7181	0.4747	1.3620
σ_f, deg	−0.2003	0.3008	0.7273
r_{xf}, m	0.5561	1.7237	6.2084
r_{zf}, m	−0.3096	0.4882	1.1180

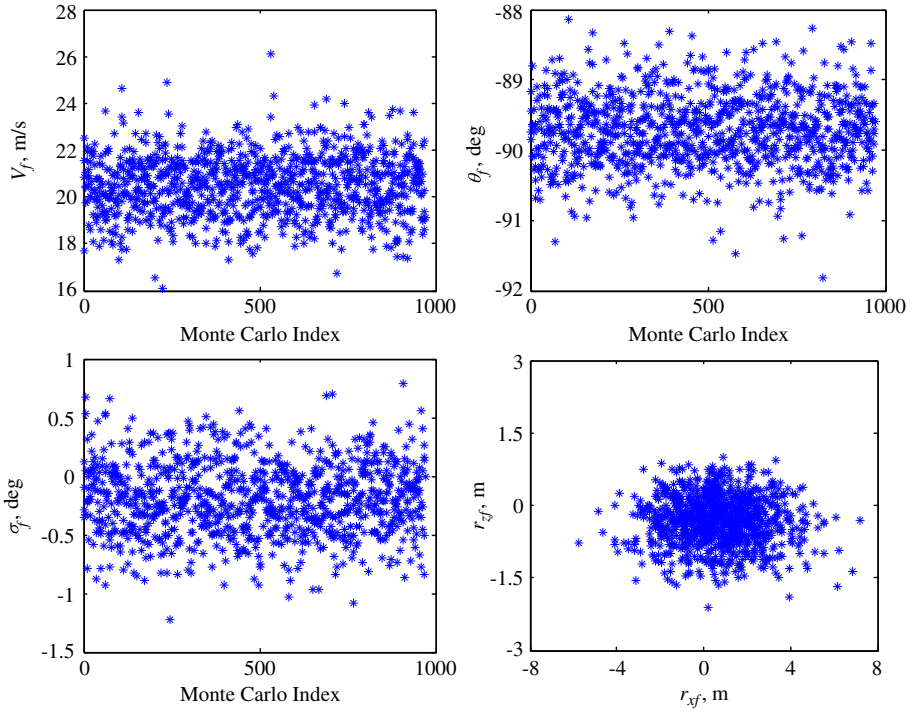

Fig. 14 Monte Carlo results.

phase because the wind speed affects the dynamic pressure $\rho(V+V_w)^2/2$ more when V is small. Similarly, because the normal and lateral control abilities of the rocket are sufficient, the effects of α_w and β_w on the normal and lateral motions are relatively small.

Finally, we consider the parameter uncertainties and wind disturbance simultaneously and perform Monte Carlo simulations of the PFGMPG algorithm. Here, we assume all parameter uncertainties and disturbances are normally distributed. The mean values of the uncertainties and disturbances are set to zero. The 3σ values are set as 15% for C_D and ρ, 20% for C_L, 20 m/s for V_w, and 15 deg for α_w and β_w. The guidance simulation is performed 1000 times. The results of 971 simulations are satisfactory. However, in the other 29 simulations, the problem is physically infeasible because there are multiple extreme parameter deviations and disturbances (e.g., in one of these cases, $C_D=-14.7\%$, $\rho=-6.9\%$, and $V_w=-19.5$ m/s), thus S-PISC becomes successively infeasible after a certain update. Although, with the guaranteed feasibility, R-PISC can still obtain solutions. However, the deviations between the actual and nominal states are no longer capable of converging to small values, and the actual state cannot reach an acceptable neighborhood of the terminal feasible set by using these solutions. The results of the 971 completed Monte Carlo simulations are shown in Table 6 and Fig. 14.

According to Table 6, under the combined effect of various parameter uncertainties and wind disturbance, the performance of PFGMPG is satisfactory. Only when multiple extreme parameter deviations and disturbances act on the rocket does guidance failure occur. Most of the guidance results fall into the terminal feasible set of the trajectory optimization, or within an acceptable neighborhood of the terminal feasible set.

VI. Conclusions

In this paper, a parallel feasibility-guaranteed model predictive guidance algorithm is proposed to solve the optimal landing guidance problem of vertical takeoff/vertical landing rockets. The main contribution of this paper is to combine the convex-optimization-based rapid trajectory optimization with a novel parallel model predictive control implementation strategy. During each guidance cycle, a high computational efficiency is guaranteed by the convex optimization theory and parallel computing technology. This enables PFGMPG to simultaneously solve a standard and a feasibility-guaranteed relaxed trajectory optimization problem. Based on such a parallel implementation, the recursive feasibility and the guidance error boundedness of the MPC-based algorithm are proven. Therefore, the proposed guidance algorithm is practically stable. Exhaustive numerical experiments are performed to evaluate the PFGMPG algorithm; and it is demonstrated that the algorithm is optimal, robust, and precise. The developed method shows the potential for online use, and it provides an option for future rocket landing guidance applications.

Appendix: Expressions of Some Variables Used in the Paper

In Eq. (1), the expression of $h_{\Omega x,y,z}$ is as follows:

$$\begin{bmatrix} h_{\Omega x} \\ h_{\Omega y}\cdot V\cos\sigma \\ -h_{\Omega z}\cdot V \end{bmatrix} = -V_G\cdot\begin{bmatrix} \Omega_x^2-\Omega^2 & \Omega_x\Omega_y & \Omega_x\Omega_z \\ \Omega_x\Omega_y & \Omega_y^2-\Omega^2 & \Omega_z\Omega_y \\ \Omega_x\Omega_z & \Omega_z\Omega_y & \Omega_z^2-\Omega^2 \end{bmatrix}\cdot\begin{bmatrix} r_x \\ r_y+R_0 \\ r_z \end{bmatrix}$$
$$-\boldsymbol{V}_G\cdot\begin{bmatrix} 0 & -2\Omega_z & 2\Omega_y \\ 2\Omega_z & 0 & -2\Omega_x \\ -2\Omega_y & 2\Omega_x & 0 \end{bmatrix}\cdot\begin{bmatrix} \dot{r}_x \\ \dot{r}_y \\ \dot{r}_z \end{bmatrix}$$
$$\boldsymbol{V}_G=\begin{bmatrix} \cos\theta\cos\sigma & \sin\theta\cos\sigma & -\sin\sigma \\ -\sin\theta & \cos\theta & 0 \\ \cos\theta sin\sigma & \sin\theta\sin\sigma & \cos\sigma \end{bmatrix},\begin{bmatrix} \Omega_x \\ \Omega_y \\ \Omega_z \end{bmatrix}=\Omega\cdot\begin{bmatrix} \cos\phi_0\cos\varphi_0 \\ \sin\phi_0 \\ -\cos\phi_0 sin\varphi_0 \end{bmatrix}$$

where Ω is the Earth self-rotation rate, ϕ_0 is the latitude of the landing site; and φ_0 is the angle between north and the projection of the velocity vector at initial time onto the xOz plane, as shown in Fig. 1.

The forms of matrices A, B, and T are as follows:

$$A=\left.\begin{bmatrix} a_{11} & a_{12} & a_{13} & 0 & a_{15} & 0 & a_{17} \\ a_{21} & a_{22} & a_{23} & 0 & a_{25} & 0 & a_{27} \\ a_{31} & a_{32} & a_{33} & 0 & a_{35} & 0 & a_{37} \\ a_{41} & a_{42} & a_{43} & 0 & 0 & 0 & 0 \\ a_{51} & a_{52} & a_{53} & 0 & 0 & 0 & 0 \\ a_{61} & 0 & a_{63} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}\right|_k, \quad B=\left.\begin{bmatrix} b_{11} & b_{12} & 0 & b_{14} & 0 \\ b_{21} & b_{22} & 0 & b_{24} & 0 \\ 0 & 0 & b_{33} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & b_{75} \end{bmatrix}\right|_k,$$

$$\text{and} \quad T=\left.\begin{bmatrix} t_{11} & t_{12} \\ \vdots & \vdots \\ t_{71} & t_{72} \end{bmatrix}\right|_k$$

The expressions of the preceding matrices and the expressions of c_{ij} in the powered phase are given as follows: The elements of the coast phase can be obtained by setting $t_0 = 0$, $t_f = t_0$, and $\Gamma = T_x = T_y = T_z = 0$. Define $\Phi = t_f - t_0$, $c\Delta$ and $s\Delta$ indicate $\cos\Delta$ and $sin\Delta$, $\bar{T}_x = c\alpha \cdot T_x + s\alpha \cdot T_y$, $\bar{T}_y = c\alpha \cdot T_y - s\alpha \cdot T_x$, $s_c\sigma = s\sigma/c^2\sigma$, $\bar{D}_\alpha = \partial\bar{D}/\partial\alpha$, $L_\alpha = \partial L/\partial\alpha$, and f^Δ indicates the right-hand-side function of Δ dynamics. Then, we have the following:

$$a_{11} = 2\Phi\bar{D}/(mV), a_{12} = \Phi\mu c\theta c\sigma/r^2, a_{12} = -\Phi\mu s\theta s\sigma/r^2,$$
$$a_{15} = -\Phi(\bar{D}/mh_0 + 2\mu s\theta c\sigma/r^3), a_{17} = -\Phi(\bar{T}_x/m^2 + \bar{D}/m^2),$$
$$a_{21} = \Phi[(\bar{T}_y + L)/(mV^2c\sigma) - \mu c\theta/(V^2r^2c\sigma)],$$
$$a_{22} = -\Phi\mu s\theta/(Vr^2c\sigma),$$
$$a_{23} = -\Phi[\bar{T}_y s_c\sigma/(mV) + Ls_c\sigma/(mV) - \mu c\theta s_c\sigma/(Vr^2)],$$
$$a_{25} = \Phi[L/(h_0mVc\sigma) - 2\mu c\theta/(Vr^3c\sigma)],$$
$$a_{27} = \Phi[\bar{T}_y/(m^2Vc\sigma) + L/(m^2Vc\sigma)],$$
$$a_{31} = -\Phi[T_z/(mV^2) - \mu s\sigma s\theta/(V^2r^2)], a_{32} = -\Phi\mu s\sigma c\theta/(Vr^2),$$
$$a_{33} = -\Phi\mu c\sigma s\theta/(Vr^2), a_{35} = 2\Phi\mu s\sigma s\theta/(Vr^3),$$
$$a_{37} = -\Phi T_z/(m^2V), a_{41} = -\Phi c\theta c\sigma, a_{42} = \Phi Vs\theta c\sigma,$$
$$a_{43} = \Phi Vc\theta s\sigma, a_{51} = -\Phi s\theta c\sigma, a_{52} = -\Phi Vc\theta c\sigma,$$
$$a_{53} = \Phi Vs\theta s\sigma.a_{61} = \Phi s\sigma, a_{63} = \Phi Vc\sigma, b_{11} = \Phi c\alpha/m,$$
$$b_{12} = \Phi s\alpha/m, b_{14} = \Phi(\bar{T}_y/m + \bar{D}_\alpha/m),$$
$$b_{21} = \Phi s\alpha/(mVc\sigma), b_{22} = -\Phi c\alpha/(mVc\sigma),$$
$$b_{24} = \Phi[\bar{T}_x/(mVc\sigma) - L_\alpha/(mVc\sigma)], b_{33} = \Phi/(mV),$$
$$b_{75} = \Phi(I_{\text{sp}}g_0).t_{11} = f^V - h_{\Omega x}, t_{21} = f^\theta - h_{\Omega y}, t_{31} = f^\sigma - h_{\Omega z},$$
$$t_{41} = f^x, t_{51} = f^y, t_{61} = f^z, t_{71} = f^m, t_{i2} = -t_{i1},$$
$$i = 1, \ldots, 7; c_{11} = \rho VS_{\text{ref}}(c\alpha C_L + s\alpha C_D)/(mg),$$
$$c_{12} = \kappa\rho V^2S_{\text{ref}}(c\alpha C_L + s\alpha C_D)/(2mg), c_{13} = 1/(mg),$$
$$c_{14} = -\rho V^2S_{\text{ref}}[s\alpha(C_L^\alpha + 2C_L^{\alpha^2}\alpha) - c\alpha(C_D^\alpha + 2C_D^{\alpha^2}\alpha)]/(2mg),$$
$$c_{21} = \rho V, c_{22} = -\rho V^2/(2h_0), c_{31} = 3.15k_Q\rho^{0.5}V^{2.15},$$
$$c_{32} = -k_Q\rho^{0.5}V^{3.15}/(2h_0)$$

Acknowledgments

We acknowledge the support to this research by the National Natural Science Foundation of China (grant no. 61403100). We would also like to thank the anonymous reviewers, whose comments were very useful in improving the quality of the paper.

References

[1] Blackmore, L., "Autonomous Precision Landing of Space Rockets," *Bridge*, Vol. 4, No. 46, 2016, pp. 15–20.
[2] Liu, X., "Fuel-Optimal Rocket Landing with Aerodynamic Controls," *AIAA Guidance, Navigation, and Control Conference*, AIAA Paper 2017-1732, Jan. 2017. doi:10.2514/6.2017-1732
[3] Lu, P., "Introducing Computational Guidance and Control," *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 2, 2017, pp. 193–193. doi:10.2514/1.G002745
[4] Ross, I. M., and Fahroo, F., "Issues in the Real-Time Computation of Optimal Control," *Mathematical and Computer Modelling*, Vol. 43, Nos. 9–10, 2006, pp. 1172–1188. doi:10.1016/j.mcm.2005.05.021
[5] Ross, I. M., Sekhavat, P., Fleming, A., and Gong, Q., "Optimal Feedback Control: Foundations, Examples, and Experimental Results for a new Approach," *Journal of Guidance, Control, and Dynamics*, Vol. 31, No. 2, 2008, pp. 307–321. doi:10.2514/1.29532
[6] Bollino, K. P., "High-Fidelity Real-Time Trajectory Optimization for Reusable Launch Vehicles," Ph.D. Thesis, Naval Postgraduate School, Monterey, CA, 2006.
[7] Sagliano, M., Mooij, E., and Theil, S., "Onboard Trajectory Generation for Entry Vehicles via Adaptive Multivariate Pseudospectral Interpolation," *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 2, 2017, pp. 466–476. doi:10.2514/1.G001817
[8] Liu, X., and Lu, P., "Solving Nonconvex Optimal Control Problems by Convex Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 3, 2014, pp. 750–765. doi:10.2514/1.62110
[9] Boyd, S., and Vandenberghe, L., *Convex Optimization*, Cambridge Univ. Press, Cambridge, England, U.K., 2004, Chaps. 1–4.
[10] Açıkmeşe, B., and Ploen, S. R., "Convex Programming Approach to Powered Descent Guidance for Mars Landing," *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 5, 2007, pp. 1353–1366. doi:10.2514/1.27553
[11] Blackmore, L., Acikmese, B., and Scharf, D. P., "Minimum-Landing-Error Powered-Descent Guidance for Mars Landing Using Convex Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 33, No. 4, 2010, pp. 1161–1171. doi:10.2514/1.47202
[12] Harris, M. W., and Açıkmeşe, B., "Lossless Convexification of Non-Convex Optimal Control Problems for State Constrained Linear Systems," *Automatica*, Vol. 50, No. 9, 2014, pp. 2304–2311. doi:10.1016/j.automatica.2014.06.008
[13] Blackmore, L., Açıkmeşe, B., and Carson, J. M., III, "Lossless Convexification of Control Constraints for a Class of Nonlinear Optimal Control Problems," *Systems Control Letters*, Vol. 61, No. 8, 2012, pp. 863–870. doi:10.1016/j.sysconle.2012.04.010
[14] Morgan, D., Chung, S.-J., and Hadaegh, F. Y., "Model Predictive Control of Swarms of Spacecraft Using Sequential Convex Programming," *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 6, 2014, pp. 1725–1740. doi:10.2514/1.G000218
[15] Liu, X., Shen, Z., and Lu, P., "Entry Trajectory Optimization by Second Order Cone Programming," *Journal of Guidance, Control, and Dynamics*, Vol. 39, No. 2, 2016, pp. 227–241. doi:10.2514/1.G001210
[16] Szmuk, M., and Açıkmeşe, B., "Successive Convexification for Fuel Optimal Powered Landing with Aerodynamic Drag and Non-Convex Constraints," *AIAA Guidance, Navigation, and Control Conference*, AIAA Paper 2016-0378, Jan. 2016. doi:10.2514/6.2016-0378
[17] Wang, Z., and Grant, M. J., "Constrained Trajectory Optimization for Planetary Entry via Sequential Convex Programming," *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 10, 2017, pp. 2603–2615. doi:10.2514/1.G002150
[18] Scharf, D. P., Açıkmeşe, B., Dueri, D., Benito, J., and Casoliva, J., "Implementation and Experimental Demonstration of Onboard Powered Descent Guidance," *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 2, 2017, pp. 213–229. doi:10.2514/1.G000399
[19] Garg, D., "Advances in Global Pseudospectral Methods for Optimal Control," Ph.D. Thesis, Univ. of Florida, Gainesville, FL, 2011.

[20] Huntington, G. T., "Advancement and Analysis of a Gauss Pseudospectral Transcription for Optimal Control," Ph.D. Thesis, Massachusetts Inst.of Technology, Cambridge, MA, 2007.
[21] Trefethen, L. N., *Spectral Methods in MATLAB*, Society for Industrial and Applied Mathematics, Philadelphia, 2000, Chaps. 1 and 4.
[22] Wang, J., and Cui, N., "A Pseudospectral-Convex Optimization Algorithm for Rocket Landing Guidance," *AIAA Guidance, Navigation, and Control Conference*, AIAA Paper 2018-1871, Jan. 2018. doi:10.2514/6.2018-1871
[23] Sagliano, M., "Pseudospectral Convex Optimization for Powered Descent and Landing," *Journal of Guidance, Control, and Dynamics*, Vol. 41, No. 2, 2018, pp. 320–334. doi:10.2514/1.G002818
[24] Mayne, D. Q., Rawlings, J. B., Rao, C. V., and Scokaert, P. O. M., "Constrained Model Predictive Control: Stability and Optimality," *Automatica*, Vol. 36, No. 6, 2000, pp. 789–814. doi:10.1016/S0005-1098(99)00214-9
[25] Eren, U., Prach, A., Koçer, B. B., Raković, S. V., Kayacan, E., and Açıkmeşe, B., "Model Predictive Control in Aerospace Systems: Current State and Opportunities," *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 7, 2017, pp. 1541–1566. doi:10.2514/1.G002507
[26] Mayne, D. Q., "Model Predictive Control: Recent Developments and Future Promise," *Automatica*, Vol. 50, No. 12, 2014, pp. 2967–2986. doi:10.1016/j.automatica.2014.10.128
[27] Carson, J. M., Açikmeşe, B., Murray, R. M., and MacMartin, D. G., "A Robust Model Predictive Control Algorithm Augmented with a Reactive Safety Mode," *Automatica*, Vol. 49, No. 5, 2013, pp. 1251–1260. doi:10.1016/j.automatica.2013.02.025
[28] Zeilinger, M. N., Raimondo, D. M., Domahidi, A., Morari, M., and Jones, C. N., "On Real-Time Robust Model Predictive Control," *Automatica*, Vol. 50, No. 3, 2014, pp. 683–694. doi:10.1016/j.automatica.2013.11.019
[29] Lu, P., "Entry Guidance: A Unified Method," *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 3, 2014, pp. 713–728. doi:10.2514/1.62605
[30] Ross, I. M., *A Primer in Pontryagin's Principle in Optimal Control*, Collegiate Publ., Carmel, CA, 2015.
[31] Liu, X., Lu, P., and Pan, B., "Survey of Convex Optimization for Aerospace Applications," *Astrodynamics*, Vol. 1, No. 1, 2017, pp. 23–40. doi:10.1007/s42064-017-0003-8
[32] Andersen, E. D., Roos, C., and Terlaky, T., "On Implementing a Primal-Dual Interior-Point Method for Conic Quadratic Optimization," *Mathematical Programming*, Vol. 95, No. 2, 2003, pp. 249–277. doi:10.1007/s10107-002-0349-3
[33] Bellman, R. E., and Dreyfus, S. E., *Applied Dynamic Programming*, Princeton Univ. Press, Princeton, NJ, 1962.
[34] Shen, Z., and Lu, P., "Onboard Generation of Three-Dimensional Constrained Entry Trajectories," *Journal of Guidance, Control, and Dynamics*, Vol. 26, No. 1, 2003, pp. 111–121. doi:10.2514/2.5021
[35] Bollino, K., Oppenheimer, M., and Doman, D., "Optimal Guidance Command Generation and Tracking for Reusable Launch Vehicle Reentry," *AIAA Guidance, Navigation, and Control Conference*, AIAA Paper 2006-6691, Aug. 2006. doi:10.2514/6.2006-6691
[36] Bollino, K., and Ross, I. M., "A Pseudospectral Feedback Method for Real-Time Optimal Guidance of Reentry Vehicles," *Proceedings of the IEEE American Control Conference*, IEEE, New York, July 2007, pp. 3861–3867. doi:10.1109/ACC.2007.4282500
[37] Bollino, K., Ross, I. M., and Doman, D. B., "Optimal Nonlinear Feedback Guidance for Reentry Vehicles," *AIAA Guidance, Navigation, and Control Conference*, AIAA Paper 2006-6074, Aug. 2006. doi:10.2514/6.2006-6074
[38] Carson, J. M., "Robust Model Predictive Control with a Reactive Safety Mode," Ph.D. Thesis, California Inst. of Technology, Pasadena, CA, 2011.
[39] Kerrigan, E. C., and Maciejowski, J. M., "Soft Constraints and Exact Penalty Functions in Model Predictive Control," *Proceedings of the UKACC International Conference on Control*, Institution of Engineering and Technology, U.K., Sept. 2000.
[40] Golub, G. H., and Ortega, J. M., *Scientific Computing: An Introduction with Parallel Computing*, Elsevier, New York, 2014.
[41] Antony, T., and Grant, M. J., "Rapid Indirect Trajectory Optimization on Highly Parallel Computing Architectures," *Journal of Spacecraft and Rockets*, Vol. 54, No. 5, 2017, pp. 1081–1091. doi:10.2514/1.A33755
[42] Pan, J., and Manocha, D., "GPU-Based Parallel Collision Detection for Fast Motion Planning," *International Journal of Robotics Research*, Vol. 31, No. 2, 2012, pp. 187–200. doi:10.1177/0278364911429335
[43] Zabczyk, J., *Mathematical Control Theory: An Introduction*, Birkhäuser, Boston, Cambridge, MA, 1992, pp. 92–93.
[44] Patterson, M. A., and Rao, A. V., "GPOPS-II: A Matlab Software for Solving Multiple-Phase Optimal Control Problems Using hp-Adaptive Gaussian Quadrature Collocation Methods and Sparse Nonlinear Programming," *ACM Transactions on Mathematical Software*, Vol. 41, No. 1, 2014, pp. 1–37. doi:10.1145/2684421
[45] Chapman, B., Jost, G., and Van der Pas, R., *Using OpenMP: Portable Shared Memory Parallel Programming*, MIT Press, Cambridge, MA, 2008.
[46] Dueri, D., Acikmese, B., Scharf, D., and Harris, M., "Customized Real Time Interior-Point Methods for Onboard Powered-Descent Guidance," *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 2, 2017, pp. 197–212. doi:10.2514/1.G001480

This article has been cited by:

1. Yang Li, Wanchun Chen, Liang Yang. 2022. Successive Chebyshev pseudospectral convex optimization method for nonlinear optimal control problems. *International Journal of Robust and Nonlinear Control* **32**:1, 326-343. [Crossref]
2. Runqiu Yang, Xinfu Liu. Gravity-Turn-Based Precise Landing Guidance for Reusable Rockets 3423-3434. [Crossref]
3. Yangyang Ma, Binfeng Pan, Chuanchuan Hao, Shuo Tang. 2022. Improved sequential convex programming using modified Chebyshev–Picard iteration for ascent trajectory optimization. *Aerospace Science and Technology* **120**, 107234. [Crossref]
4. Jiawei Wang, Ran Zhang, Huifeng Li. 2022. Onboard optimization of multi-arc trajectories with constraints on duration of arcs. *Acta Astronautica* **109**. . [Crossref]
5. Zhengyu Song, Cong Wang. 2021. Powered soft landing guidance method for launchers with non-cluster configured engines. *Acta Astronautica* **189**, 379-390. [Crossref]
6. Yangyang Ma, Binfeng Pan, Shuo Tang. Improved Parallel-Structured Newton-Type Guidance for Launch Vehicles Under Thrust Drop Fault. *Journal of Spacecraft and Rockets*, ahead of print1-15. [Abstract] [Full Text] [PDF] [PDF Plus]
7. Biao Yang, Wuxing Jing, Changsheng Gao. 2021. Online midcourse guidance method for boost phase interception via adaptive convex programming. *Aerospace Science and Technology* **118**, 107037. [Crossref]
8. Shaozhao Lu, Yao Zhang, Xingang Li, Quan Hu. 2021. Decentralized closed-loop optimization for 6-DOF self-assembly satellites. *Acta Astronautica* **114**. . [Crossref]
9. Henry Stoldt, Declan Quinn, Jake Kavanagh, Craig T. Johansen. MAPLEAF: A Compact, Extensible, Open-Source, 6-Degrees-of-Freedom Rocket Flight Simulation Framework . [Abstract] [PDF] [PDF Plus]
10. Francesco Marchetti, Edmondo Minisci. 2021. Genetic Programming Guidance Control System for a Reentry Vehicle under Uncertainties. *Mathematics* **9**:16, 1868. [Crossref]
11. Dawei Yang. 2021. Target Tracking Algorithm Based on Adaptive Scale Detection Learning. *Complexity* **2021**, 1-11. [Crossref]
12. Jian Zhao, Xiangyue He, Haiyang Li, Lin Lu. 2021. An adaptive optimization algorithm based on clustering analysis for return multi-flight-phase of VTVL reusable launch vehicle. *Acta Astronautica* **183**, 112-125. [Crossref]
13. Shengwei Jia, Fugui Li, Mingbo Tong. 2021. Instantaneous Impact Point Guidance with Coast Arcs for Solid Rockets. *International Journal of Aerospace Engineering* **2021**, 1-12. [Crossref]
14. Sait SOVUKLUK, Mert ANKARALI. 2021. Giriş Kısıtlı Ters Sarkaç Mekanizmasının Optimal Kontrolü. *European Journal of Science and Technology* . [Crossref]
15. Guangtong Xu, Yan Cao, Jingliang Sun, Zhexuan Zhang, Teng Long. Real-Time Path Generation for UAV Swarms Using Receding Planning Framework and Priority Decoupling Mechanism 4338-4343. [Crossref]
16. Yu Song, Xinyuan Miao, Shengping Gong. 2021. Adaptive powered descent guidance based on multi-phase pseudospectral convex optimization. *Acta Astronautica* **180**, 386-397. [Crossref]
17. Huan Zhang, Zhihua Zhao, Gexue Ren, Pengxiang Hu, Yunfei Yang, Zhongwen Pan, Francesco Sanfedino. 2021. Arresting-Cable System for Robust Terminal Landing of Reusable Rockets. *Journal of Spacecraft and Rockets* **58**:2, 425-443. [Abstract] [Full Text] [PDF] [PDF Plus] [Supplementary Material]
18. Byeong-Un Jo, Jaemyung Ahn. 2021. Optimal staging of reusable launch vehicles considering velocity losses. *Aerospace Science and Technology* **109**, 106431. [Crossref]
19. Sergio Perez-Roca, Julien Marzat, Helene Piet-Lahanier, Nicolas Langlois, Marco Galeotta, Francois Farago, Serge Le Gonidec. 2021. Model-Based Robust Transient Control of Reusable Liquid-Propellant Rocket Engines. *IEEE Transactions on Aerospace and Electronic Systems* **57**:1, 129-144. [Crossref]
20. Danylo Malyuta, Yue Yu, Purnanand Elango, Behçet Açıkmeşe. 2021. Advances in trajectory optimization for space vehicle control. *Annual Reviews in Control* **52**, 282-315. [Crossref]
21. Jinbo Wang, Huixu Li, Hongbo Chen. 2020. An Iterative Convex Programming Method for Rocket Landing Trajectory Optimization. *The Journal of the Astronautical Sciences* **67**:4, 1553-1574. [Crossref]
22. Boris Benedikter, Alessandro Zavoli, Guido Colasurdo, Simone Pizzurro, Enrico Cavallini. Autonomous Upper Stage Guidance Using Convex Optimization and Model Predictive Control . [Abstract] [PDF] [PDF Plus]
23. Zhenbo Wang, Ye Lu. 2020. Improved Sequential Convex Programming Algorithms for Entry Trajectory Optimization. *Journal of Spacecraft and Rockets* **57**:6, 1373-1386. [Abstract] [Full Text] [PDF] [PDF Plus]

24. Bin Li, Hongbo Zhang, Wei Zheng, Lei Wang. 2020. Spacecraft close-range trajectory planning via convex optimization and multi-resolution technique. *Acta Astronautica* **175**, 421-437. [Crossref]
25. Cong-ying Cai, Xiao-lan Yao. 2020. Trajectory optimization with constraints for alpine skiers based on multi-phase nonlinear optimal control. *Frontiers of Information Technology & Electronic Engineering* **21**:10, 1521-1534. [Crossref]
26. Qingqing Dang, Haichao Gui, Kun Liu, Bo Zhu. 2020. Relaxed-Constraint Pinpoint Lunar Landing Using Geometric Mechanics and Model Predictive Control. *Journal of Guidance, Control, and Dynamics* **43**:9, 1617-1630. [Abstract] [Full Text] [PDF] [PDF Plus]
27. Weibo Xia, Weihong Wang, Wenjing Zhai, Ziqian Gao, Chuan Gao. A Fast Trajectory Optimization of Pinpoint Landing by Chebyshev Pseudospectral and SOCP 3773-3778. [Crossref]
28. Jian Zhao, Haiyang Li, Xiangyue He, Yuechen Huang, Jianghui Liu. 2020. Uncertainty Analysis for Return Trajectory of Vertical Takeoff and Vertical Landing Reusable Launch Vehicle. *Mathematical Problems in Engineering* **2020**, 1-18. [Crossref]
29. Xiang Zhou, Hong-Bo Zhang, Lei Xie, Guo-Jian Tang, Wei-Min Bao. 2020. An improved solution method via the pole-transformation process for the maximum-crossrange problem. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering* **234**:9, 1491-1506. [Crossref]
30. Yuan Li, Baojun Pang, Changzhu Wei, Naigang Cui, Yongbei Liu. 2020. Online trajectory optimization for power system fault of launch vehicles via convex programming. *Aerospace Science and Technology* **98**, 105682. [Crossref]
31. Lei Xie, Hongbo Zhang, Xiang Zhou, Guojian Tang. 2020. A Convex Programming Method for Rocket Powered Landing With Angle of Attack Constraint. *IEEE Access* **8**, 100485-100496. [Crossref]
32. Lei He, Xiaodong Yan, Shuo Tang. 2019. Spiral-diving trajectory optimization for hypersonic vehicles by second-order cone programming. *Aerospace Science and Technology* **95**, 105427. [Crossref]
33. Jinbo Wang, Naigang Cui, Changzhu Wei. 2019. Rapid trajectory optimization for hypersonic entry using a pseudospectral-convex algorithm. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering* **233**:14, 5227-5238. [Crossref]
34. Yuan Li, Yingzi Guan, Yongzhi Shan, Yanfeng He. Rapid Trajectory Re-plan for Power System Fault of Ballistic Missiles via Convex Optimization 655-660. [Crossref]
35. Shaul Gutman. Rendezvous and Soft Landing in Closed Form via LQ Optimization 536-540. [Crossref]
36. Yuan Li, Yingzi Guan, Changzhu Wei, Renyi Hu. 2019. Optimal Control of Ascent Trajectory for Launch Vehicles: A Convex Approach. *IEEE Access* **7**, 186491-186498. [Crossref]