Hindawi
International Journal of Aerospace Engineering
Volume 2021, Article ID 2734230, 13 pages
https://doi.org/10.1155/2021/2734230

Research Article

Fuel-Optimal Ascent Trajectory Problem for Launch Vehicle via Theory of Functional Connections

Shiyao Li,[1] **Yushen Yan**,[1] **Kun Zhang**,[2] **and Xinguo Li**[1,3]

[1]*School of Astronautics, Northwestern Polytechnical University, Xi'an 710072, China*
[2]*Xi'an Modern Control Technology Research Institute, Xi'an 710065, China*
[3]*National Key Laboratory of Aerospace Flight Dynamics, Northwestern Polytechnical University, Xi'an 710072, China*

Correspondence should be addressed to Shiyao Li; lishiyao@mail.nwpu.edu.cn

Received 3 June 2021; Revised 16 August 2021; Accepted 25 August 2021; Published 13 September 2021

Academic Editor: Erdal Kayacan

In this study, we develop a method based on the Theory of Functional Connections (TFC) to solve the fuel-optimal problem in the ascending phase of the launch vehicle. The problem is first transformed into a nonlinear two-point boundary value problem (TPBVP) using the indirect method. Then, using the function interpolation technique called the TFC, the problem's constraints are analytically embedded into a functional, and the TPBVP is transformed into an unconstrained optimization problem that includes orthogonal polynomials with unknown coefficients. This process effectively reduces the search space of the solution because the original constrained problem transformed into an unconstrained problem, and thus, the unknown coefficients of the unconstrained expression can be solved using simple numerical methods. Finally, the proposed algorithm is validated by comparing to a general nonlinear optimal control software GPOPS-II and the traditional indirect numerical method. The results demonstrated that the proposed algorithm is robust to poor initial values, and solutions can be solved in less than 300 ms within the MATLAB implementation. Consequently, the proposed method has the potential to generate optimal trajectories on-board in real time.

1. Introduction

With the recent development in space exploration, launch vehicles are very important as they are the only means for humans to explore space from the earth. In general, a launch vehicle mission has been planned over a long period, and the trajectory was designed in advance, and it cannot be updated during flight, which means it is not robust or flexible. The fast launch and trajectory reconstruction are the main research of the guidance system, and both need rapid trajectory planning technology. Rapid trajectory planning can shorten the launch mission cycle and quickly update the trajectory in case of thrust failure during the flight of the vehicle to ensure the success of the mission.

The primary aim of the trajectory planning algorithm is to solve the optimal control problem that is generally based on nonlinear dynamics, which achieves specific performance indicators under the constraints of state and control variables. The solution of such problems is mainly achieved using the indirect method [1–3] and the direct method [4–6]. The direct method transforms the optimal control problem of continuous space into a nonlinear programming problem and uses a numerical method to directly optimize the performance index [7–9]. Although the direct method, represented by a sequential quadratic programming algorithm with the pseudospectral discrete, has advanced a lot over a period of time, it still has considerable issues between the on-board application. The algorithm for the general nonlinear programming, for example, the famous sequential quadratic programming algorithm, had low algorithm efficiency and low sensitivity to the initial value and was unable to guarantee convergence in the past, but now, these issues have easily resolved by using a convex optimization method. Ralph Rockafellar, a renowned mathematician, pointed out that the key determinant of the performance of a numerical optimization algorithm is neither the linearity nor nonlinearity of the problem, but the convexity or nonconvexity of the problem [10]. In 2007, JPL proposed lossless convexity technology for the dynamic descent guidance of the Mars lander [11]. After that, a systematic summary of the research and development of lossless convexity technology is presented in [12]. The applicability of the lossless convexity method is extended to general linear systems with multiple state constraints using the concepts of control theory. Unfortunately, only a few nonconvex constraints can be used for lossless convexification, and there is no analytical convexification method for the system dynamic constraints in the trajectory optimization problem of aircraft. A method based on the Newton–Kantorovich/pseudospectral and sequence convexification is used for the ascending phase of the launch vehicle [13]. However, the sequential convex optimization algorithm is a convexification method based on linearization, which increases the dependence on the reference trajectory. This not only offers higher requirements for the reference trajectory but also annihilates the advantage of the convex optimization algorithm that does not rely on the good initial value. Nevertheless, considering the rapidity of the convex optimization algorithm in solving convex problems, in recent years, trajectory planning based on the convex optimization algorithm, such as planetary landing [14–16], rocket ascent guidance [17], and entry guidance [18], has been widely studied.

The indirect method solves the optimal control problem by using the classical variational method and the Pontryagin Minimum Principle, derives the first-order necessary conditions of the optimal control, and transforms the optimal control problem into a two-point boundary value problem (TPBVP) [19] that is comprised of initial conditions, Hamiltonian differential equations, optimal conditions, and terminal boundary conditions (including terminal transversal conditions and terminal constraints). However, since the convergence radius of the indirect method is small, and the convergence of the numerical iteration is extremely sensitive to the initial value estimation, which requires a higher accuracy of the initial value estimation, determination of the initial value is highly difficult. To overcome this problem, the deep learning algorithm is used to obtain a higher accuracy of initial value estimation and a better target shooting success rate [20–22]. In general, the higher sensitivity of TPBVP to the initial value makes the problem difficult to solve. Therefore, although the indirect method yields a more accurate solution, it is rarely used in practice.

Recently, a mathematical framework called the Theory of Functional Connections (TFC) has been proposed in [23] to derive the expressions with embedded constraints. The expressions, called constrained expressions, are composed of functionals and functions of functions. The constrained expression is written as

$$y(t) = g(t) + \sum_{n}^{k=1} \Phi_k(t) p_k(t), \tag{1}$$

where $g(t)$ is a free function and $\Phi_k(t)$ are the switch functions composed of the support function $s_k(t)$ with unknown coefficients α_k. The support function is a set of linearly independent functions. If one of the switch functions is equal to 1, the constraint it is referencing is evaluated; otherwise, that is, if it is equal to 0, all other constraints are evaluated. The switch functions can be expressed as$\Phi_k(t) = s_k(t)\alpha_k$. The projection functionals $p_k(t)$ are derived by constraint functions. The constrained optimization problem is transformed into an unconstrained one using the constrained expressions, which reduces the search space of the solution to the admissible solutions that satisfy all constraints. Finally, using common basis functions such as Chebyshev polynomials or Legendre polynomials to express the free function and then using the least-square method to find its unknown coefficient, the solution of the problem can be found. In [24, 25], the TFC algorithm quickly solves the nonlinear differential equations and obtains high-precision solutions. In [26, 27], it is applied to fixed-time asteroid landing and optimal energy landing. The results show that the solution time is basically less than 100 ms, which proves that the algorithm has real-time application potential.

This paper is organized as follows. Section 2 gives a brief description of the TFC mathematical framework to solving TPBVP. In Section 3, the fuel-optimal ascent trajectory problem is described in detail, and the necessary conditions are derived. In Section 4, the fuel-optimal problem in the ascending phase of the launch vehicle is formulated using the TFC framework. Finally, the result and discussion are provided in Section 5.

2. Theory of Functional Connections

In this section, we present an outline of the TFC mathematical framework and a method for solving second-order TPBVP with TFC.

2.1. TFC for TPBVP. In general, trajectory optimization problems are second-order TPBVP [28], which is expressed as

$$F(t, y(t), \dot{y}(t), \ddot{y}(t)) = 0 \quad \text{subject to}: \begin{cases} y(t_0) = y_0, \\ y(t_f) = y_f, \\ \dot{y}(t_0) = \dot{y}_0, \\ \dot{y}(t_f) = \dot{y}_f, \end{cases} \tag{2}$$

where t_0 and t_f represent the initial time and the terminal time, respectively, and $y_0, y_f, \dot{y}_0, \dot{y}_f$ are the initial and terminal constraints, respectively. As mentioned earlier, the constraints are expressed by (1), and then, (2) is simply rewritten as

$$y(t) = g(t) + \Phi_1(t)p_1 + \Phi_2(t)p_2 + \Phi_3(t)p_3 + \Phi_4(t)p_4, \tag{3}$$

where $p_k(t)$ is expressed as

$$\begin{aligned} p_1(t, g_i(t)) &= y_{0_i} - g_i(t_0), \\ p_2(t, g_i(t)) &= y_{f_i} - g_i(t_f), \\ p_3(t, g_i(t)) &= \dot{y}_{0_i} - \dot{g}_i(t_0), \\ p_4(t, g_i(t)) &= \dot{y}_{f_i} - \dot{g}_i(t_f). \end{aligned} \tag{4}$$

The algorithm to derive the term of the constrained expression is given as follows:

(1) Choose $s_k(t)$, which are k linearly independent support functions

(2) Calculate switching functions $\Phi_k(t)$ as a linear combination of the support functions with k unknown coefficients

(3) Formulate a system of equations to solve for the unknown coefficients based on $\Phi_k(t)$

The support functions are defined as $s_k(k) = t^{k-1}$, according to [28]. The switching function is then obtained by solving equations. When the first switch function is activated, the equations are

$$\begin{cases} \Phi_1(t_0) = 1, \\ \Phi_1(t_f) = 0, \\ \Phi_1'(t_0) = 0, \\ \Phi_1'(t_f) = 0. \end{cases} \tag{5}$$

The equations of the first switch functions are combined into the matrix form as

$$\begin{bmatrix} 1 & t_0 & t_0^2 & t_0^3 \\ 1 & t_f & t_f^2 & t_f^3 \\ 0 & 1 & 2t_0 & 3t_0^2 \\ 0 & 1 & 2t_f & 3t_f^2 \end{bmatrix} \begin{bmatrix} \alpha_{11} \\ \alpha_{21} \\ \alpha_{31} \\ \alpha_{41} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{6}$$

Similarly, application of this method to the other three switch functions yields the following matrix:

$$\begin{bmatrix} 1 & t_0 & t_0^2 & t_0^3 \\ 1 & t_f & t_f^2 & t_f^3 \\ 0 & 1 & 2t_0 & 3t_0^2 \\ 0 & 1 & 2t_f & 3t_f^2 \end{bmatrix} \begin{bmatrix} \alpha_{11} & \alpha_{21} & \alpha_{31} & \alpha_{41} \\ \alpha_{21} & \alpha_{22} & \alpha_{32} & \alpha_{42} \\ \alpha_{31} & \alpha_{23} & \alpha_{33} & \alpha_{43} \\ \alpha_{41} & \alpha_{24} & \alpha_{34} & \alpha_{44} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{7}$$

The coefficient α_k is obtained by matrix inversion, and the expression of the switch functions is obtained as

$$\begin{aligned} \Phi_1(t) &= \frac{1}{(t_f - t_0)^3}\left(-t_f^2(3t_0 - t_f) + 6t_0 t_f t - 3(t_0 + t_f)t^2 + 2t^3\right), \\ \Phi_2(t) &= \frac{1}{(t_f - t_0)^3}\left(-t_0^2(t_0 - 3t_f) - 6t_0 t_f t + 3(t_0 + t_f)t^2 - 2t^3\right), \\ \Phi_3(t) &= \frac{1}{(t_f - t_0)^2}\left(-t_0 t_f^2 + t_f(2t_0 + t_f)t - (t_0 + 2t_f)t^2 + t^3\right), \\ \Phi_4(t) &= \frac{1}{(t_f - t_0)^2}\left(-t_0^2 t_f + t_0(t_0 + 2t_f)t - (2t_0 + t_f)t^2 + t^3\right). \end{aligned} \tag{8}$$

The boundary conditions of (2) are effectively embedded within the constrained expression by substituting (4) and (8) into (3). Then, by substituting (3) into the vector differential equation $F(t, y(t), \dot{y}(t), \ddot{y}(t))$, the constrained TPBVPs are transformed into an unconstrained problem. According to (3), $y(t)$ is replaced by $g(t)$; thus, the original vector differential equation $F(t, y(t), \dot{y}(t), \ddot{y}(t))$ is transformed into $\widehat{F}(t, g(t), \dot{g}(t), \ddot{g}(t))$, which is only a function of t, the free functions $g(t)$, and their derivatives

$$\widehat{F}(t, g(t), \dot{g}(t), \ddot{g}(t)) = 0. \tag{9}$$

As mentioned earlier, $\widehat{F}(t, g(t), \dot{g}(t), \ddot{g}(t))$ is unconstrained because the boundary is represented by the switch function $\Phi_k(t)$ and the projection function $p_k(t)$.

After determining the switch function and the projection function, we next discuss the free function $g(t)$.

2.2. Definition of the Free Function. In selecting a free function, we are essentially looking for the best function approximator. A natural choice for the free function is a linear combination of basis functions, as this is capable of spanning the entire function space that the basis spans, as the number of basis functions approaches infinity. The free function is expressed as

$$g(t) = \xi^T h, \tag{10}$$

where ξ are $m \times 1$ unknown coefficients and h are m basis functions.

Next, the problem domain t is mapped to the domain of the basis functions z, and Chebyshev and Legendre polynomials are commonly used, the domains of which are defined in $[-1, 1]$. To implement the basis functions, a map between t and z is defined as

$$z = z_0 + \frac{z_f - z_0}{t_f - t_0}(t - t_0) \leftrightarrow t = t_0 + \frac{t_f - t_0}{z_f - z_0}(z - z_0). \tag{11}$$

By using (11), the derivatives of $g(t)$ are computed as

$$\frac{\mathrm{d}^k g}{\mathrm{d}t^k} = c^k \xi^T \frac{\mathrm{d}^k h(z)}{\mathrm{d}z^k}, \quad \text{where } k \in [0, n], \tag{12}$$

where $c \coloneqq \mathrm{d}z/\mathrm{d}t = (z_f - z_0)/(t_f - t_0)$.

2.3. Domain Discretization. For solving TPBVPs numerically, the domain $t \in [t_0, t_f]$ must be discretized by $N + 1$ points. The common method is uniform distribution, but the advantage of Chebyshev–Gauss–Lobatto collocation points is that when the number of basis functions increases, the condition number should also increase slowly, which is useful for improving computational efficiency. The Chebyshev–Gauss–Lobatto collocation points are defined as

$$z_k = -\cos\left(\frac{k\pi}{N}\right) \quad \text{for } k = 1, 2, \cdots, N. \tag{13}$$

Thus, the new vector differential equation $\widehat{F}(t, g(t), \dot{g}(t), \ddot{g}(t))$ becomes $\tilde{F}(z, \xi)$, where the unknown coefficient ξ is the variable that needs to be solved. $\tilde{F}(z, \xi)$ is expressed in the form of loss functions at each discrete point

$$\mathscr{L}_i(\xi) = \left\{ \begin{array}{c} \tilde{F}_i(z_0, \xi) \\ \vdots \\ \tilde{F}_i(z_d, \xi) \\ \vdots \\ \tilde{F}_i(z_N, \xi) \end{array} \right\}. \tag{14}$$

By setting $\mathscr{L}_i = 0$, the unknown coefficient ξ is solved using optimization schemes such as iterative least-squares.

To solve the nonlinear least-square problem, we need the Jacobian matrix of the loss function, which is written as

$$J(\xi) = \begin{bmatrix} \dfrac{\partial \tilde{F}_i(z_0, \xi)}{\partial \xi_1} & \dfrac{\partial \tilde{F}_i(z_0, \xi)}{\partial \xi_2} & \cdots \dfrac{\partial \tilde{F}_i(z_0, \xi)}{\partial \xi_j} \\ \vdots & \vdots & \vdots \\ \dfrac{\partial \tilde{F}_i(z_d, \xi)}{\partial \xi_1} & \dfrac{\partial \tilde{F}_i(z_d, \xi)}{\partial \xi_2} & \cdots \dfrac{\partial \tilde{F}_i(z_d, \xi)}{\partial \xi_j} \\ \vdots & \vdots & \vdots \\ \dfrac{\partial \tilde{F}_i(z_N, \xi)}{\partial \xi_1} & \dfrac{\partial \tilde{F}_i(z_N, \xi)}{\partial \xi_2} & \cdots \dfrac{\partial \tilde{F}_i(z_N, \xi)}{\partial \xi_j} \end{bmatrix}. \tag{15}$$

The estimation is updated by

$$\xi^{k+1} = \xi^k - \Delta\xi, \tag{16}$$

where $\Delta\xi = \left(J(\xi)^T J(\xi)\right)^{-1} J(\xi)^T \mathscr{L}(\xi)$.

The iterative process stops when the convergence tolerance is met:

$$L_2[\mathscr{L}(\xi)] < \varepsilon, \tag{17}$$

where ε is the stopping criterion that is defined by the user.

Figure 1 shows the outline of the TFC framework.

Recently, the position of numerical calculation in the current guidance and control field is emphasized in [29], which, based on numerical calculation, is called as the Computational Guidance and Control (CG&C). The CG&C replaces offline planning and closed-loop guidance with on-board computing, which is more robust, more accurate, and more flexible and can adapt to more complex environments and missions, but offers high requirements for computational efficiency. As mentioned earlier, the algorithms used in the CG&C are basically divided into two: direct and indirect. In this study, we used the indirect method, because the optimal control problem is transformed into TPBVP, and then, the TFC method is used to transform and solve the problem.

We found that the TFC method is quite similar to the collocation method in which all three methods using orthogonal polynomials over the global domain, according to the different method. In the collocation method, the states and costates are expanded by using orthogonal polynomials, and the boundary conditions are considered as part of the optimization scheme [30]. It is similar that the pseudospectral method used orthogonal polynomials like Chebyshev or Legendre polynomials to approximate the states and costates, and the boundary conditions are also considered as part of the optimization scheme. The TFC method may use a similar operation, but the fundamental difference between the TFC method and the other two methods lies in handling the constraints of the problem. The TFC method uses orthogonal polynomials to expand the free function $g(t)$ in a constrained expression and then expresses the problem constraints analytically step by step as mentioned above, which can reduce the search space of the solution and thus improve the computational efficiency. In fact, the advantages of the TFC method are presented in [27]; the results in [27] show that the TFC method is two orders of magnitude faster than the pseudospectral method in a fixed time optimal control problem and one order of magnitude faster than the pseudospectral method in a free time optimal control problem.

3. Fuel-Optimal Problem in the Ascending Phase of the Launch Vehicle

In this section, the problem is transformed into a TPBVP, and the first-order necessary conditions and transversal conditions of the problem are derived using the Pontryagin Minimum Principle.

3.1. Dynamical Model. In this section, the last stage of the launching vehicle is studied. The dimensionless equations of motion of a three-dimensional (3-D) launch vehicle can be expressed in the Earth Center Inertial Coordinate System as follows:

$$\begin{aligned} \dot{r} &= v, \\ \dot{v} &= \frac{T}{m} + a_g, \\ \dot{m} &= \frac{T}{I_{sp} g_0}, \end{aligned} \tag{18}$$

where r is the inertial position, which is normalized by the radius of the Earth $R_0 = 6378145$ m. v is the velocity, which is normalized by $\sqrt{R_0 g_0}$, in which $g_0 = 9.81\ \mathrm{m/s^2}$ represents the gravitational acceleration magnitude on the surface of the Earth. The mass of the launch vehicle is denoted by m. The thrust is denoted by $T = TI_b$, where I_b is the unit vector of the body axis satisfying

$$\|I_b\| \equiv 1. \tag{19}$$

For most launch vehicles, the mass flow is uncontrollable; thus, the thrust magnitude $T = \|T\|$ is constant and uncontrollable during the same flight phase. The gravity acceleration

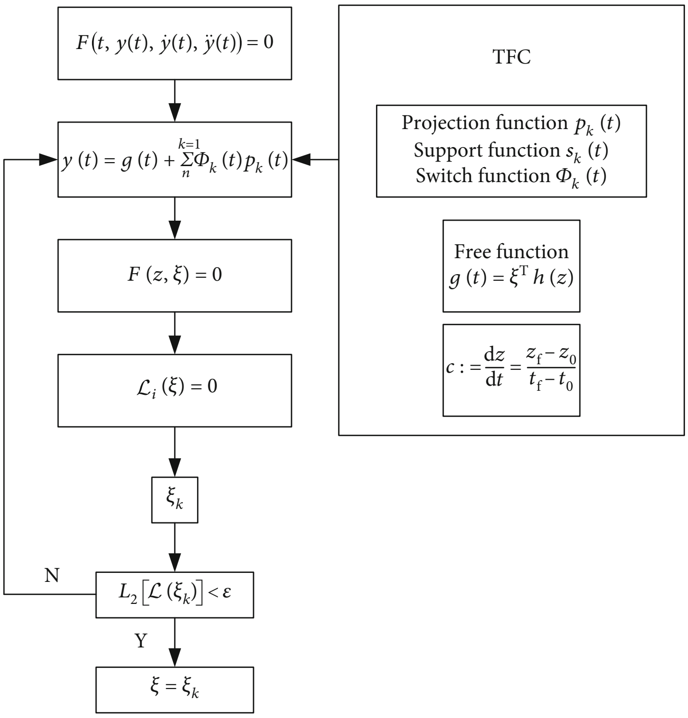
$F(t, y(t), \dot{y}(t), \ddot{y}(t)) = 0$
$y(t) = g(t) + \sum_{n}^{k=1} \Phi_k(t) p_k(t)$
$F(z, \xi) = 0$
$\mathcal{L}_i(\xi) = 0$
ξ_k
N
$L_2[\mathcal{L}(\xi_k)] < \varepsilon$
Y
$\xi = \xi_k$
TFC
Projection function $p_k(t)$
Support function $s_k(t)$
Switch function $\Phi_k(t)$
Free function
$g(t) = \xi^{\mathrm{T}} h(z)$
$c := \frac{\mathrm{d}z}{\mathrm{d}t} = \frac{z_f - z_0}{t_f - t_0}$

Figure 1: TFC framework.

$a_g = -r/r^3$, where $r = \|r\|$. The specific impulse of the engine is denoted by I_{sp}. The differentiation of the equations in (18) is with respect to dimensionless time normalized by $\sqrt{R_0/g_0}$.

As the mass flow is constant, the optimal ascent problem is treated as a minimum-time problem:

$$\min \ J = t_f, \tag{20}$$

subject to

$$\begin{gathered}\dot{r} = v, \quad \dot{v} = \frac{T}{m} + a_g, \quad \dot{m} = \frac{T}{I_{sp} g_0}, \\ r(0) = r_0, \quad v(0) = v_0, \quad m(0) = m_0, \\ r(t_f) = r_f, \quad v(t_f) = v_f.\end{gathered} \tag{21}$$

3.2. First-Order Necessary Conditions. On the basis of the Pontryagin Minimum Principle, the mass of the vehicle is treated as a prescribed function of time instead of a state variable. The Hamiltonian function is written as

$$H = 1 + \lambda_r^T v + \lambda_v^T \left(\frac{T}{m} I_b - \frac{r}{r^3} \right), \tag{22}$$

where $I_b = -\lambda_v / \lambda_v$ is called the primer's vector, according to Lawden's theory [31]. Thus, (22) is rewritten as

$$H = 1 + \lambda_r^T v - \frac{T}{m} \|\lambda_v\| - \frac{\lambda_v^T r}{r^3}. \tag{23}$$

The first-order necessary conditions for optimality then give the differential equations of the costate variables:

$$\begin{aligned}\dot{\lambda}_r &= -\frac{\partial H}{\partial r} = \frac{\lambda_v^T r}{r^5}, \\ \dot{\lambda}_v &= -\frac{\partial H}{\partial v} = -\lambda_r.\end{aligned} \tag{24}$$

The transversality condition is expressed as

$$H(t_f) = 0. \tag{25}$$

4. Solution via TFC

4.1. TPBVP in TFC Framework. As mentioned in the previous section, to find the optimal state, the following nonlinear TPBVPs must be solved:

$$\dot{r} = v, \tag{26}$$

$$\dot{v} = -\frac{T}{m} \frac{\lambda_v}{\lambda_v} - \frac{r}{r^3}, \tag{27}$$

$$\dot{m} = \frac{T}{I_{sp} g_0}, \tag{28}$$

$$\dot{\lambda}_r = -\frac{\partial H}{\partial r} = \frac{\lambda_v^T r}{r^5}, \tag{29}$$

$$\dot{\lambda}_v = -\frac{\partial H}{\partial v} = -\lambda_r, \tag{30}$$

$$H\left(t_f\right) = 1 + \lambda_r^T\left(t_f\right)v\left(t_f\right) - \frac{T}{m}\left\|\lambda_v\left(t_f\right)\right\| - \frac{\lambda_v^T\left(t_f\right)r\left(t_f\right)}{r^3\left(t_f\right)} = 0, \tag{31}$$

where (26) and (27) are subject to

$$\begin{aligned} r(0) &= r_0, \quad v(0) = v_0, \quad m(0) = m_0, \\ r\left(t_f\right) &= r_f, \quad v\left(t_f\right) = v_f. \end{aligned} \tag{32}$$

Additionally, there is a redundant equation and can be removed by the TFC constraints. The derivative of $r(t)$ is exactly the function $\mathrm{v}(t)$ because TFC constraints are analytical expressions; thus, (26) can be disregarded. The problem is now reduced. The new equations are expressed as

$$\begin{aligned} \mathscr{L}_a(t) &= a - \dot{\lambda}_r - \frac{\lambda_v^T r}{r^5}, \\ \mathscr{L}_r(t) &= \dot{\lambda}_r - \frac{\lambda_v^T r}{r^5}, \\ \mathscr{L}_v(t) &= \dot{\lambda}_v + \lambda_r, \\ \mathscr{L}_H &= 1 + \lambda_r^T v - \frac{T}{m}\left\|\lambda_v\right\| - \frac{\lambda_v^T r}{r^3}. \end{aligned} \tag{33}$$

To solve the above equations, the TFC constraints with $r(t), \lambda_r(t), \lambda_v(t)$ need to be constructed by the TFC method. The unknown coefficients in the TFC constraint expressions are expressed as $\xi_a, \xi_{\lambda_r}, \xi_{\lambda_v}$, and $v(t), a(t), \dot{\lambda}_r(t), \dot{\lambda}_v(t)$ can be obtained by taking the derivative of the TFC constraint expression $r(t), \lambda_r(t), \lambda_v(t)$, respectively.

The initial and terminal constraints of the problem discussed in this paper are position and velocity constraints, respectively. The TFC constraint expressions of $r(t), v(t), a(t)$ are written as

$$\begin{aligned} r(t) &= g(t) + \Phi_1(t)p_1 + \Phi_2(t)p_2 + \Phi_3(t)p_3 + \Phi_4(t)p_4, \\ v(t) &= g(t) + \dot{\Phi}_1(t)p_1 + \dot{\Phi}_2(t)p_2 + \dot{\Phi}_3(t)p_3 + \dot{\Phi}_4(t)p_4, \\ a(t) &= g(t) + \ddot{\Phi}_1(t)p_1 + \ddot{\Phi}_2(t)p_2 + \ddot{\Phi}_3(t)p_3 + \ddot{\Phi}_4(t)p_4, \end{aligned} \tag{34}$$

where the projection function is written as

$$\begin{aligned} p_1(t) &= r_0 - g(t_0), \\ p_2(t) &= r_f - g\left(t_f\right), \\ p_3(t) &= v_0 - \dot{g}(t_0), \\ p_4(t) &= v_f - \dot{g}\left(t_f\right). \end{aligned} \tag{35}$$

Next, consider constructing a free function. According to (10)–(12), the time domain is mapped to the Chebyshev domain. However, it should be noted that in the time-free TPBVP, the parameter c is a function of t_f. Combined with the TFC method, the new unknown variable ξ_t is used to represent the parameter c, and the optimal time is obtained by solving ξ_t. In addition, to ensure the solved final time is positive, b is used instead of b, where $b^2 = c$. Then, $r(t)$ is rewritten as

$$\begin{aligned} r(t) = r(z) &= \Phi_1(z)r_0 + \Phi_2(z)r_f + \Phi_3(z)\frac{v_0}{b^2} + \Phi_4(z)\frac{v_f}{b^2} \\ &\quad + \left(h(z) - \Phi_1(z)h_0 - \Phi_2(z)h_f - \Phi_3(z)h_0' - \Phi_4(z)h_f'\right)^T \xi_a, \\ v(t) = b^2 v(z) &= b^2\left[\Phi_1'(z)r_0 + \Phi_2'(z)r_f + \Phi_3'(z)\frac{v_0}{b^2} + \Phi_4'(z)\frac{v_f}{b^2}\right. \\ &\quad \left. + \left(h'(z) - \Phi_1'(z)h_0 - \Phi_2'(z)h_f - \Phi_3'(z)h_0' - \Phi_4'(z)h_f'\right)^T \xi_a\right], \\ a(t) = b^4 r(z) &= b^4\left[\Phi_1''(z)r_0 + \Phi_2''(z)r_f + \Phi_3''(z)\frac{v_0}{b^2} + \Phi_4''(z)\frac{v_f}{b^2}\right. \\ &\quad \left. + \left(h''(z) - \Phi_1''(z)_0 - \Phi_2''(z)h_f - \Phi_3''(z)h_0' - \Phi_4''(z)h_f'\right)^T \xi_a\right], \end{aligned} \tag{36}$$

where $h_0 = h(z_0)$ and $h_f = h(z_f)$.

The expression of the switch function is similar to (8):

$$\begin{aligned} \Phi_1(z) &= \frac{1}{\left(z_f - z_0\right)^3}\left(-z_f^2\left(3z_0 - z_f\right) + 6z_0 z_f z - 3\left(z_0 + z_f\right)z^2 + 2z^3\right), \\ \Phi_2(z) &= \frac{1}{\left(z_f - z_0\right)^3}\left(-z_0^2\left(z_0 - 3z_f\right) - 6z_0 z_f z + 3\left(z_0 + z_f\right)z^2 - 2z^3\right), \\ \Phi_3(z) &= \frac{1}{\left(z_f - z_0\right)^2}\left(-z_0 z_f^2 + z_f\left(2z_0 + z_f\right)z - \left(z_0 + 2z_f\right)z^2 + z^3\right), \\ \Phi_4(z) &= \frac{1}{\left(z_f - z_0\right)^2}\left(-z_0^2 z_f + z_0\left(z_0 + 2z_f\right)z - \left(2z_0 + z_f\right)z^2 + z^3\right). \end{aligned} \tag{37}$$

Next, the TFC constraint expressions of the costates are constructed by the same steps as above:

$$\begin{aligned} \lambda_r(t) &= \lambda_r(z) = h(z)^T \xi_r, \\ \dot{\lambda}_r(t) &= b^2 \lambda_r'(z) = b^2 h'(z)^T \xi_r, \\ \lambda_v(t) &= \lambda_v(z) = h(z)^T \xi_v, \\ \dot{\lambda}_v(t) &= b^2 \lambda_v'(z) = b^2 h'(z)^T \xi_v. \end{aligned} \tag{38}$$

Substitution of the above TFC constraint expression into the loss function yields the loss function with respect to unknown coefficient ξ. Then, the solution of the problem is obtained using the nonlinear least-square method. The unknown coefficient ξ is expressed as

$$\xi = \left\{ \xi_{a_1}^T \, \xi_{a_2}^T \, \xi_{a_3}^T \, \xi_{\lambda_{r1}}^T \, \xi_{\lambda_{r2}}^T \, \xi_{\lambda_{r3}}^T \, \xi_{\lambda_{v1}}^T \, \xi_{\lambda_{v2}}^T \, \xi_{\lambda_{v3}}^T \, \xi_t \right\}^T \in \mathbb{R}^{9m+1}. \tag{39}$$

The loss function can be expressed as

$$\mathbb{L} = \left\{ \mathscr{L}_{a_1}^T \, \mathscr{L}_{a_2}^T \, \mathscr{L}_{a_3}^T \, \mathscr{L}_{\lambda_{r1}}^T \, \mathscr{L}_{\lambda_{r2}}^T \, \mathscr{L}_{\lambda_{r3}}^T \, \mathscr{L}_{\lambda_{v1}}^T \, \mathscr{L}_{\lambda_{v2}}^T \, \mathscr{L}_{\lambda_{v3}}^T \, \mathscr{L}_H \right\}^T. \tag{40}$$

To use the nonlinear least-square method, the partial derivative of the loss function needs to be calculated as

$$a(t) = b^4 r(t) = b^4 \left[\Phi_1'(z) r_0 + \Phi_2'(z) r_f + \Phi_3'(z) \frac{v_0}{b^2} + \Phi_4'(z) \frac{v_f}{b^2} + \left(h''(z) - \Phi_1'(z) h_0 - \Phi_2'(z) h_f - \Phi_3'(z) h_0' - \Phi_4'(z) h_f' \right)^T \xi_a \right]. \tag{41}$$

The terms of (41) are defined by

$$J_{a,\xi_a} = \begin{bmatrix} \frac{\partial \mathscr{L}_{a_1}}{\partial \xi_{a_1}} & \frac{\partial \mathscr{L}_{a_1}}{\partial \xi_{a_2}} & \frac{\partial \mathscr{L}_{a_1}}{\partial \xi_{a_3}} \\ \frac{\partial \mathscr{L}_{a_2}}{\partial \xi_{a_1}} & \frac{\partial \mathscr{L}_{a_2}}{\partial \xi_{a_2}} & \frac{\partial \mathscr{L}_{a_2}}{\partial \xi_{a_3}} \\ \frac{\partial \mathscr{L}_{a_3}}{\partial \xi_{a_1}} & \frac{\partial \mathscr{L}_{a_3}}{\partial \xi_{a_2}} & \frac{\partial \mathscr{L}_{a_3}}{\partial \xi_{a_3}} \end{bmatrix}, \quad J_{a,\xi_v} = \begin{bmatrix} \frac{\partial \mathscr{L}_{a_1}}{\partial \xi_{v_1}} & \frac{\partial \mathscr{L}_{a_1}}{\partial \xi_{v_2}} & \frac{\partial \mathscr{L}_{a_1}}{\partial \xi_{v_3}} \\ \frac{\partial \mathscr{L}_{a_2}}{\partial \xi_{v_1}} & \frac{\partial \mathscr{L}_{a_2}}{\partial \xi_{v_2}} & \frac{\partial \mathscr{L}_{a_2}}{\partial \xi_{v_3}} \\ \frac{\partial \mathscr{L}_{a_3}}{\partial \xi_{v_1}} & \frac{\partial \mathscr{L}_{a_3}}{\partial \xi_{v_2}} & \frac{\partial \mathscr{L}_{a_3}}{\partial \xi_{v_3}} \end{bmatrix},$$

$$J_{a,\xi_t} = \begin{bmatrix} \frac{\partial \mathscr{L}_{a_1}}{\partial \xi_t} & \frac{\partial \mathscr{L}_{a_2}}{\partial \xi_t} & \frac{\partial \mathscr{L}_{a_3}}{\partial \xi_t} \end{bmatrix}^T,$$

$$J_{\lambda_r,\xi_a} = \begin{bmatrix} \frac{\partial \mathscr{L}_{r_1}}{\partial \xi_{a_1}} & \frac{\partial \mathscr{L}_{r_1}}{\partial \xi_{a_2}} & \frac{\partial \mathscr{L}_{r_1}}{\partial \xi_{a_3}} \\ \frac{\partial \mathscr{L}_{r_2}}{\partial \xi_{a_1}} & \frac{\partial \mathscr{L}_{r_2}}{\partial \xi_{a_2}} & \frac{\partial \mathscr{L}_{r_2}}{\partial \xi_{a_3}} \\ \frac{\partial \mathscr{L}_{r_3}}{\partial \xi_{a_1}} & \frac{\partial \mathscr{L}_{r_3}}{\partial \xi_{a_2}} & \frac{\partial \mathscr{L}_{r_3}}{\partial \xi_{a_3}} \end{bmatrix}, \quad J_{\lambda_r,\xi_r} = \begin{bmatrix} \frac{\partial \mathscr{L}_{r_1}}{\partial \xi_{r_1}} & 0 & 0 \\ 0 & \frac{\partial \mathscr{L}_{r_2}}{\partial \xi_{r_2}} & 0 \\ 0 & 0 & \frac{\partial \mathscr{L}_{r_3}}{\partial \xi_{r_3}} \end{bmatrix},$$

$$J_{\lambda_r,\xi_v} = \begin{bmatrix} \frac{\partial \mathscr{L}_{r_1}}{\partial \xi_{v_1}} & 0 & 0 \\ 0 & \frac{\partial \mathscr{L}_{r_2}}{\partial \xi_{v_2}} & 0 \\ 0 & 0 & \frac{\partial \mathscr{L}_{r_3}}{\partial \xi_{v_3}} \end{bmatrix}, \quad J_{\lambda_r,\xi_b} = \begin{bmatrix} \frac{\partial \mathscr{L}_{r_1}}{\partial \xi_t} \\ \frac{\partial \mathscr{L}_{r_2}}{\partial \xi_t} \\ \frac{\partial \mathscr{L}_{r_3}}{\partial \xi_t} \end{bmatrix},$$

$$J_{\lambda_v,\xi_r} = \begin{bmatrix} \frac{\partial \mathscr{L}_{v_1}}{\partial \xi_{r_1}} & 0 & 0 \\ 0 & \frac{\partial \mathscr{L}_{v_2}}{\partial \xi_{r_2}} & 0 \\ 0 & 0 & \frac{\partial \mathscr{L}_{v_3}}{\partial \xi_{r_3}} \end{bmatrix}, \quad J_{\lambda_v,\xi_v} = \begin{bmatrix} \frac{\partial \mathscr{L}_{v_1}}{\partial \xi_{v_1}} & 0 & 0 \\ 0 & \frac{\partial \mathscr{L}_{v_2}}{\partial \xi_{v_2}} & 0 \\ 0 & 0 & \frac{\partial \mathscr{L}_{v_3}}{\partial \xi_{v_3}} \end{bmatrix},$$

$$J_{\lambda_v,\xi_t} = \begin{bmatrix} \frac{\partial \mathscr{L}_{v_1}}{\partial \xi_t} & \frac{\partial \mathscr{L}_{v_2}}{\partial \xi_t} & \frac{\partial \mathscr{L}_{v_3}}{\partial \xi_t} \end{bmatrix}^T,$$

$$J_{H,\xi_a} = \begin{bmatrix} \frac{\partial \mathscr{L}_H}{\partial \xi_{a_1}} & \frac{\partial \mathscr{L}_H}{\partial \xi_{a_2}} & \frac{\partial \mathscr{L}_H}{\partial \xi_{a_3}} \end{bmatrix},$$

$$J_{H,\xi_r} = \begin{bmatrix} \frac{\partial \mathscr{L}_H}{\partial \xi_{r_1}} & \frac{\partial \mathscr{L}_H}{\partial \xi_{r_2}} & \frac{\partial \mathscr{L}_H}{\partial \xi_{r_3}} \end{bmatrix},$$

$$J_{H,\xi_v} = \begin{bmatrix} \frac{\partial \mathscr{L}_H}{\partial \xi_{v_1}} & \frac{\partial \mathscr{L}_H}{\partial \xi_{v_2}} & \frac{\partial \mathscr{L}_H}{\partial \xi_{v_3}} \end{bmatrix},$$

$$J_{H,\xi_t} = \frac{\partial \mathscr{L}_H}{\partial \xi_t}, \tag{42}$$

where all partial derivatives are provided in the Appendix.

4.2. Initialization. When using the iterative least-square method to calculate, some parameters need to be estimated reasonably, so that the iterative process of the algorithm can go on without violating the basic mathematical principles. The simplest initialization is to set them equal to zero. This is equivalent to connecting the boundary value problem with the simplest interpolating expression. However, the λ_v is related to the thrust direction; initialization of $\lambda_v = 0$ will cause issues in the TFC method because $I_b = -\lambda_v/\lambda_v$ and $\| I_b \| \equiv 1$. Thus, the coefficient ξ_v is initialized using

$$\xi_{v_0} = \frac{v_0}{\|v_0\|}, \quad \xi_{v_f} = -\frac{r_0}{\|r_0\|}. \tag{43}$$

In addition, the first guess of ξ_r, ξ_a is set equal to zero, and for setting the first guess of ξ_t, an estimate of the final time t_f is needed; in this paper, the initial t_f is set to be 300, and the initial ξ_t can be expressed as

$$\xi_t = \sqrt{\frac{2}{t_f - t_0}}. \tag{44}$$

These are uniformly discrete according to the number of polynomials m.

5. Simulations

In this section, we apply the proposed algorithm to the ascent problem of the launch vehicle to verify the feasibility of the algorithm, and the results are compared with those of the GPOPS-II and classical indirect method solutions of the other two algorithms. All numerical results are obtained on a desktop with Intel Xeon E3-1230 3.4 GHz. Table 1 lists the parameters of the launch vehicle and mission in the numerical simulations.

Table 2 shows the initial and terminal parameters of the experiment, where the orbital elements corresponding to the terminal position and velocity are also given because the launch vehicle generally uses the orbital elements for the target.

To prove the validity and effectiveness of the algorithm proposed in this study, the results of the algorithm proposed in this paper are compared with those obtained by the traditional indirect method and GPOPS-II. The final time calculated by GPOPS-II is 300.97 s, that obtained by the single shooting method is 301.01 s, and the final time obtained by TFC is 301.25 s. The locations of the vehicle and the velocity vector are provided in Figures 2 and 3, respectively. Figure 4 shows the thrust vector of the launch vehicle. In the bottom part of Figure 4, the purple line representing the sum of squares of thrust directions is equal to 1, which also indicates the validity of the TFC solution, and the other three lines are the vector of the body axis. Considering that the pitch angle and yaw angle are generally used as the guidance command of the launch

Table 1: Parameters for launch vehicle.

Parameter	Value
Longitude of launch point (°)	110.95
Latitude of launching point (°)	19.61
Launch azimuth (°)	90
The specific impulse, I_{sp} (m/s)	3365
Thrust, T (N)	2843599.98
Mass rate, $\dot{m}$ (kg/s)	845.052

Table 2: Parameters of the boundary conditions.

Parameter	Value
Initial position, r_0 (m)	[371973.739, 6493779.849, -13899.978]
Initial velocity, v_0 (m/s)	[3652.033, 556.843, -2.666]
Initial mass, m_0 (kg)	350306
Semimajor (m)	6595487
Eccentricity	0.0053
Inclination (°)	20.009
Ascending node (°)	17.250
Argument of perigee (°)	32.390
True anomaly (°)	77.969
Terminal position, r_f (m)	[1912866.558, 6304148.648, 2551.256]
Terminal velocity, v_f (m/s)	[7457.930, -2220.619, 178.661]

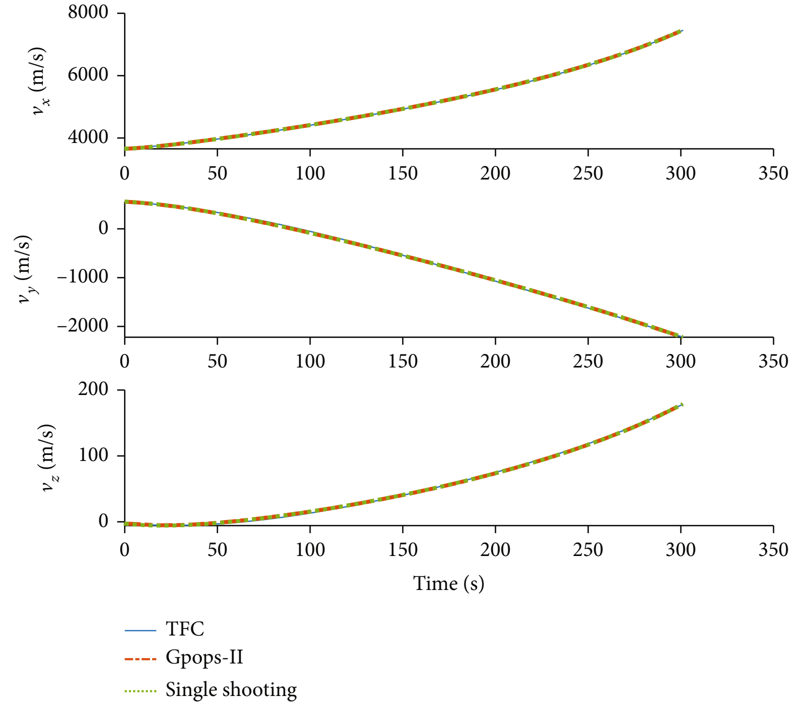

Figure 2: Vector of velocity solution by TFC, GPOPS-II, and single shooting.

vehicle, the results of pitch angle and yaw angle are also given here. Figures 5 and 6 show the results of the height and the velocity, respectively. It can be seen that the results solved by the three methods are basically the same, which again shows the validity of the TFC algorithm proposed in this study. To quantify the accuracy of the TFC method,

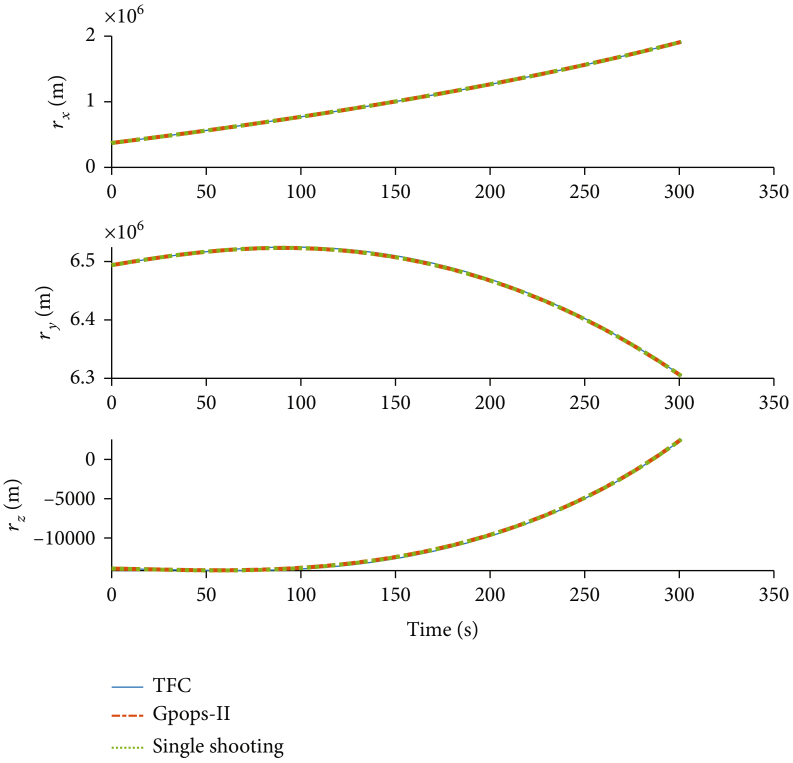

Figure 3: Vector of position solution by TFC, GPOPS-II, and single shooting.

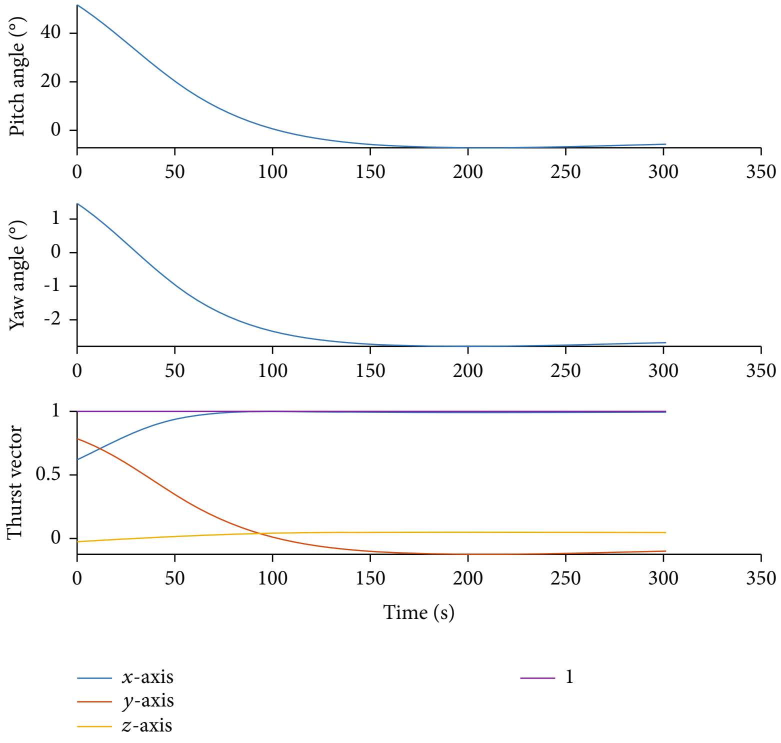

Figure 4: Vector of thrust solution by TFC.

Figure 7 shows the residual of acceleration, and it can be seen that the residual of TFC is about 10^{-14} or less for the whole solution domain.

In this simulation, the cost time of the proposed method is 0.23 s and those of GPOPS-II and the single shooting method are 3.88 and 0.34 s, respectively. It is also known that

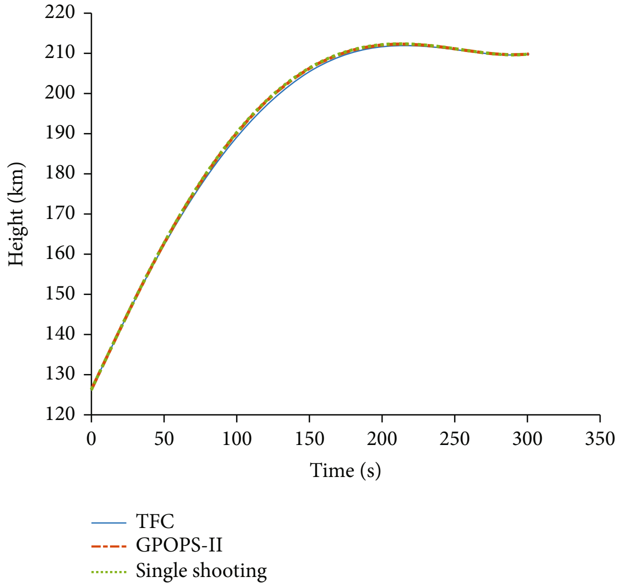

Figure 5: Height solution by TFC, GPOPS-II, and single shooting.

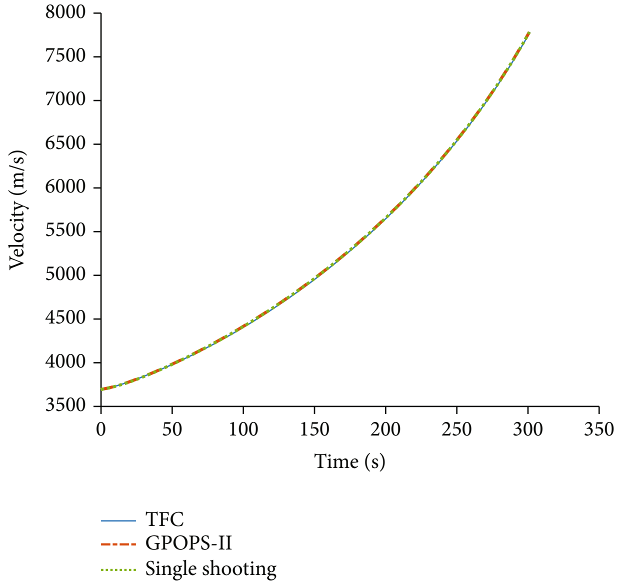

Figure 6: Velocity solution by TFC, GPOPS-II, and single shooting.

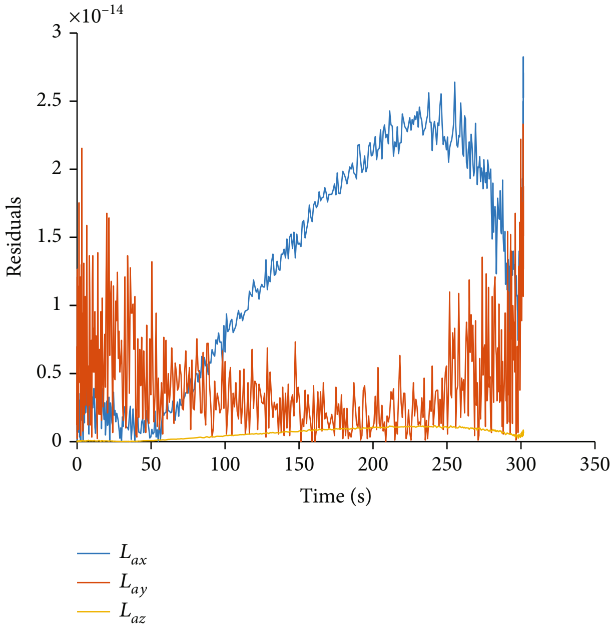

Figure 7: Residual by TFC.

Table 3: Solutions of different discrete points and polynomial numbers.

Discrete points	State and costate polynomial numbers	First-order optimality	Cost time (s)	Iteration number
100	10, 4	0.414	0.15	61
100	30, 4	0.00138	0.16	14
100	60, 4	2.09*e*-06	0.23	15
100	80, 4	0.00526	0.29	14
100	80, 10	2.13*e*-05	0.42	29
100	60, 60	50.3	—	Max
50	10, 4	0.349	0.16	54
50	30, 4	5.94*e*-05	0.23	45
50	50, 4	0.000555	0.25	26

the MATLAB programming language is 10 times slower than C++; thus, the algorithm proposed in this study has online application potential. In terms of solution accuracy, the TFC method is not dominant among the three methods, but combined with the analysis of calculation efficiency, it shows that the proposed method is an effective method.

The above results show the comparison between the results of the proposed algorithm, GPOPS-II, and single shooting method, which verifies the validity of the proposed algorithm. Next, the effect of the number of discrete points and polynomials on the algorithm is studied. Table 3 shows that the excessive number of discrete points will not only reduce the calculation efficiency, but also reduce the accuracy of the solution. In addition, the selection of the number of polynomials is also analyzed in this paper. In [28], the number of state polynomials and costate polynomials is the same and not studied separately. In our simulation, the results show that the selection of the number of state polynomials and costate polynomials can be different, and a better result can be obtained. If the number of the state polynomials is selected too much, then not only the calculation performance will be degraded but also the accuracy of the solution will be greatly affected. It is seen from Table 3 that when the number of costate polynomials is as large as the number of state polynomials, the allowed iteration number is reached and the iteration progress will not converge; it means the solution of the problem cannot be solved. In addition, appropriately increasing the number of costate polynomials can increase the accuracy of the solution, but it will reduce the computational efficiency. Thus, the number of discrete points and polynomials should be select carefully.

6. Conclusion

In this study, we proposed a new approach to solve the fuel-optimal problem in the ascending phase of the launching vehicle using TFC. The main conclusions can be summarized as follows:

(1) The first-order necessary condition of the optimization problem is constructed by the indirect method; the problem's constraints are embedded in the expression by using the TFC method

(2) The constrained optimization problem is transformed into an unconstrained optimization problem by using the TFC method, which reduces the search space of the solution, and a simple root-finding algorithm can be used to obtain the solution of the problem

(3) The residual of the solution is about 10^{-14} or less; for obtaining more accurate numerical solutions, the number of discrete points and polynomials should be selected carefully

(4) The proposed algorithm has the potential for online application. The calculation time of the algorithm is within 300 ms with MATLAB programming

Appendix

Partial Derivative of Loss Functions

The partial derivative of loss functions is

$$\frac{\partial \mathscr{L}_{a_i}}{\partial \xi_{v_j}} = \begin{cases} \dfrac{T}{m}\left(\dfrac{h_v^T(z)}{\|\lambda_v\|} - \dfrac{h_v^T(z)\lambda_{v_j}^2}{\|\lambda_v\|^3}\right), & i = j, \\ -\dfrac{h_v^T(z)\lambda_{v_i}\lambda_{v_j}}{\|\lambda_v\|^3}, & i \neq j, \end{cases} \tag{A.1}$$

$$\begin{aligned} \frac{\partial \mathscr{L}_{a_i}}{\partial \xi_t} = {} & 4b^3\left(h''(z) - \Phi_1''(z)h_0 - \Phi_2''(z)h_f - \Phi_3''(z)h_0' - \Phi_4''(z)h_f'\right)^T \xi_{a_i} \\ & + 2b\left(\Phi_3''(z)v_{0_i} + \Phi_4''(z)v_{f_i}\right) - \frac{-2T}{\left(m_0 - (z+1)\dot{m}/b^2\right)^2}\frac{(z+1)\dot{m}}{b^3} \\ & + \frac{-2\left(\Phi_3(z)v_{0_i} + \Phi_4(z)v_{f_i}\right)}{b^3\|r\|^3} + \frac{6\|r\|\left(\Phi_3(z)v_{0_i} + \Phi_4(z)v_{f_i}\right)}{b^3\|r\|^5}, \end{aligned} \tag{A.2}$$

$$\frac{\partial \mathscr{L}_{r_i}}{\partial \xi_{v_j}} = \begin{cases} h_v^T \dfrac{r_1}{\|r\|^3}, & i = j, \\ 0, & i \neq j, \end{cases} \quad \frac{\partial \mathscr{L}_{r_i}}{\partial \xi_t} = 2b{h'}_r^T \xi_{r_i}, \tag{A.3}$$

$$\frac{\partial \mathscr{L}_{v_i}}{\partial \xi_{r_j}} = \begin{cases} {h'}_r^T, & i = j, \\ 0, & i \neq j, \end{cases} \quad \frac{\partial \mathscr{L}_{v_i}}{\partial \xi_{v_j}} = \begin{cases} b^2{h'}_v^T, & i = j, \\ 0, & i \neq j, \end{cases} \tag{A.5}$$

$$\frac{\partial \mathscr{L}_{v_i}}{\partial \xi_t} = 2b{h'}_v^T \xi_{v_i}, \quad \frac{\partial \mathscr{L}_H}{\partial \xi_{r_i}} = h_{rf}^T v_{\text{if}}, \tag{A.6}$$

$$\frac{\partial \mathscr{L}_H}{\partial \xi_{v_i}} = \left(-\frac{T}{m_0 - \left(\left(z_f - z_0\right)/b^2\right)\dot{m}}\frac{\lambda_{v_{\text{if}}}}{\|\lambda_{vf}\|} - \frac{r_f}{\|r_f\|^3}\right)h_{vf}^T, \tag{A.7}$$

$$\frac{\partial \mathscr{L}_H}{\partial \xi_t} = -2T\|\lambda_{vf}\| \frac{-2\dot{m}}{b^3\left(m_0 - \left(\left(z_f - z_0\right)/b^2\right)\dot{m}\right)^2}, \tag{A.8}$$

$$\begin{aligned} \frac{\partial \mathscr{L}_H}{\partial \xi_{ai}} = {} & -\lambda_v \frac{\left(h(z) - \Phi_1(z)h_0 - \Phi_2(z)h_f - \Phi_3(z)h_0' - \Phi_4(z)h_f'\right)^{\mathrm{T}}}{\|r\|^3} \\ & - \frac{3r_i^2\left(h(z) - \Phi_1(z)h_0 - \Phi_2(z)h_f - \Phi_3(z)h_0' - \Phi_4(z)h_f'\right)^T}{\|r\|^5}, \end{aligned} \tag{A.8}$$

$$\frac{\partial \mathscr{L}_{a_i}}{\partial \xi_{a_j}} = \begin{cases} b^4\left(h''(z) - \Phi_1''(z)h_0 - \Phi_2''(z)h_f - \Phi_3''(z)h_0' - \Phi_4''(z)h_f'\right) + \dfrac{\left(h(z) - \Phi_1(z)h_0 - \Phi_2(z)h_f - \Phi_3(z)h_0' - \Phi_4(z)h_f'\right)^T}{\|r\|^3} - \dfrac{3r_i^2\left(h(z) - \Phi_1(z)h_0 - \Phi_2(z)h_f - \Phi_3(z)h_0' - \Phi_4(z)h_f'\right)^T}{r^5}, & i = j, \\ -\dfrac{3r_ir_j\left(h(z) - \Phi_1(z)h_0 - \Phi_2(z)h_f - \Phi_3(z)h_0' - \Phi_4(z)h_f'\right)^T}{\|r\|^5}, & i \neq j, \end{cases} \tag{A.9}$$

$$\frac{\partial \mathscr{L}_{r_i}}{\partial \xi_{a_j}} = \begin{cases} \lambda_{v_i}^T\left[\dfrac{\left(h(z) - \Phi_1(z)h_0 - \Phi_2(z)h_f - \Phi_3(z)h_0' - \Phi_4(z)h_f'\right)^T}{\|r\|^3}, \right. \\ \left. -\dfrac{3r_i^2\left(h(z) - \Phi_1(z)h_0 - \Phi_2(z)h_f - \Phi_3(z)h_0' - \Phi_4(z)h_f'\right)^T}{\|r\|^5}\right], & i = j, \\ -\lambda_{v_i}^T\dfrac{3r_ir_j\left(h(z) - \Phi_1(z)h_0 - \Phi_2(z)h_f - \Phi_3(z)h_0' - \Phi_4(z)h_f'\right)^T}{\|r\|^5}, & i \neq j. \end{cases} \tag{A.10}$$

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] P. Lu, H. Sun, and B. Tsai, "Closed-loop endoatmospheric ascent guidance," *Journal of Guidance, Control, and Dynamics*, vol. 26, no. 2, pp. 283–294, 2003.

[2] P. Lu and B. Pan, "Highly constrained optimal launch ascent guidance," *Journal of Guidance, Control, and Dynamics*, vol. 33, no. 2, pp. 404–414, 2010.

[3] M. J. Grant and R. D. Braun, "Rapid indirect trajectory optimization for conceptual design of hypersonic missions," *Journal of Spacecraft and Rockets*, vol. 52, no. 1, pp. 177–182, 2014.

[4] F. Fahroo and I. M. Ross, "Advances in pseudospectral methods for optimal control," in *AIAA Guidance, Navigation and Control Conference and Exhibit*, Honolulu, Hawaii, 2008.

[5] M. A. Patterson and A. V. Rao, "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, pp. 1–37, 2014.

[6] I. M. Ross and M. Karpenko, "A review of pseudospectral optimal control: from theory to flight," *Annual Reviews in Control*, vol. 36, no. 2, pp. 182–197, 2012.

[7] H. Peng, X. Wang, M. Li, and B. Chen, "An hp symplectic pseudospectral method for nonlinear optimal control," *Communications in Nonlinear Science and Numerical Simulation*, vol. 42, pp. 623–644, 2017.

[8] X. Wang, H. Peng, S. Zhang, B. Chen, and W. Zhong, "A symplectic pseudospectral method for nonlinear optimal control problems with inequality constraints," *ISA Transactions*, vol. 68, pp. 335–352, 2017.

[9] H. Peng, X. Wang, S. Zhang, and B. Chen, "An iterative symplectic pseudospectral method to solve nonlinear state-delayed optimal control problems," *Communications in Nonlinear Science and Numerical Simulation*, vol. 48, pp. 95–114, 2017.

[10] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, 1970.

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

[12] M. W. Harris, *Lossless convexification of optimal control problems*, Doctor of Philosophy, The University of Texas at Austin, 2014, https://repositories.lib.utexas.edu/handle/2152/24904.

[13] X. Cheng, H. Li, and R. Zhang, "Efficient ascent trajectory optimization using convex models based on the Newton–Kantorovich/pseudospectral approach," *Aerospace Science and Technology*, vol. 66, pp. 140–151, 2017.

[14] F. F. Al-Bakri, A. F. Al-Bakri, and C. A. Kluever, "Approach and landing guidance for an unpowered gliding vehicle," *Journal of Guidance, Control, and Dynamics*, vol. 43, no. 12, pp. 2366–2371, 2020.

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

[16] B. Acikmese, J. M. Carson, and L. Blackmore, "Lossless convexification of nonconvex control bound and pointing constraints of the soft landing optimal control problem," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 6, pp. 2104–2113, 2013.

[17] Y. Li, B. Pang, C. Wei, N. Cui, and Y. Liu, "Online trajectory optimization for power system fault of launch vehicles via convex programming," *Aerospace Science and Technology*, vol. 98, article 105682, 2020.

[18] Z. Wang and M. J. Grant, "Constrained trajectory optimization for planetary entry via sequential convex programming," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 10, pp. 2603–2615, 2017.

[19] A. E. Bryson and Y. Ho, *Applied Optimal Control*, Technometrics, 1975.

[20] L. Cheng, F. Jiang, Z. Wang, and J. Li, "Multi-constrained real-time entry guidance using deep neural networks," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 57, no. 1, pp. 325–340, 2021.

[21] D. Izzo and E. Öztürk, "Real-time guidance for low-thrust transfers using deep neural networks," *Journal of Guidance, Control, and Dynamics*, vol. 44, no. 2, pp. 315–327, 2021.

[22] L. Cheng, H. Li, Z. Wang, and F. Jiang, "Fast solution continuation of time-optimal asteroid landing trajectories using deep neural networks," *Acta Astronautica*, vol. 167, pp. 63–72, 2020.

[23] D. Mortari, "The theory of connections: connecting points," *Mathematics*, vol. 5, no. 4, p. 4, 2017.

[24] D. Mortari, "Least-squares solutions of linear differential equations," 2017, http://arxiv.org/abs/1702.08437.

[25] H. Johnston and D. Mortari, "Least-squares solutions of boundary-value problems in hybrid systems," *Journal of Computational and Applied Mathematics*, vol. 393, article 113524, 2021.

[26] E. Schiassi, A. D'Ambrosio, A. Scorsoglio, R. Furfaro, and F. Curti, *Class of optimal space guidance problems solved via indirect methods and physics-informed neural networks*, 2021.

[27] E. Schiassi, A. D'Ambrosio, H. Johnston, R. Furfaro, F. Curti, and D. Mortari, *Complete energy optimal landing on small and large planetary bodies via theory of functional connections*, 2020.

[28] K. Drozd, R. Furfaro, E. Schiassi, H. Johnston, and D. Mortari, "Energy-optimal trajectory problems in relative motion solved via theory of functional connections," *Acta Astronautica*, vol. 182, pp. 361–382, 2021.

[29] D. Mortari, H. Johnston, and L. Smith, "High accuracy least-squares solutions of nonlinear differential equations," *Journal of Computational and Applied Mathematics*, vol. 352, pp. 293–307, 2019.

[30] L. Ping, "Introducing computational guidance and control," *Journal of Guidance Control & Dynamics*, vol. 40, no. 2, pp. 193–193, 2017.

[31] D. E. Rutherford and D. F. Lawden, "Optimal trajectories for space navigation," *The Mathematical Gazette*, vol. 48, no. 366, p. 478, 1964.