Robust Bang-Off-Bang Low-Thrust Guidance Using Model Predictive Static Programming

Yang Wang * and Francesco Topputo†

Politecnico di Milano, Milano, Italy

Model Predictive Static Programming (MPSP) was always used under the assumption of continuous control, which impedes it for applications with bang-off-bang control directly. In this paper, MPSP is employed for the first time as a guidance scheme for low-thrust transfers with bang-off-bang control where the fuel-optimal trajectory is used as the nominal solution. In our method, dynamical equations in Cartesian coordinates are augmented by the mass costate equation, while the unconstrained velocity costate vector is used as control variable, and is expressed as a combination of Fourier basis functions with corresponding weights. A two-loop MPSP algorithm is designed where the weights and the initial mass costate are updated in the inner loop and continuation is conducted on the outer loop in case of large perturbations. The sensitivity matrix (SM) is recursively calculated using analytical derivatives and SM at switching points is compensated based on calculus of variations. An sample interplanetary CubeSat mission to an asteroid is used as study case to illustrate the effectiveness of the method developed.

I. Introduction

In recent decades, highly efficient propulsion systems, such as electric propulsion and solar sails, have made low-thrust engines an alternative to enable ambitious space missions. Extensive work has focused on high-fidelity modeling and open-loop optimal low-thrust trajectory design, solved by direct or indirect methods [1–4]. However, in real-world applications, disturbances such as solar radiation pressure, irregular gravitational fields, outgassing, and unmodeled accelerations, deviate the spacecraft from the nominal trajectory, which requires to update the control profile. The commonly used strategy is to up-link control commands from ground. This requires massive off-line computations and frequent communications between the ground station and the spacecraft. Due to rapid proliferation of space probes, this strategy hardly meets the always-increasing demand for autonomy [5].

In literature, a number of guidance laws were proposed for low-thrust orbit transfer problems. Edelbaum [6] employed the constant-thrust steering law for quasi-circular orbits. Casalino and Colasurdo [7] further improved the Edelbaum’s method by considering variable specific impulse and thrust magnitude with constant power level.

Preliminary results were presented as paper IAC-19,C1,9,3,x49975 at the 70th International Astronautical Congress, Washington D. C., America, 21-25 October 2019.

*PhD Candidate, Department of Aerospace Science and Technology, Via Masa 32, Milano, 20156, Italy.
†Assistant Professor, Department of Aerospace Science and Technology, Via Masa 32, Milano, 20156, Italy
Kluever [8] designed the simple guidance scheme that blended the individual control law which maximize the time rate of change of a desired orbital element. Petropoulos [9] presented the so-called Q-law for low-thrust orbit transfers, where the proximity quotient Q is served as a candidate Lyapunov function. Hernandez and Akella [10, 11] designed Lyapunov control methods for low-thrust orbit transfers using Levi-Civita and Kustaanheimo-Stiefel transformations.

Nonlinear optimal control theory (NOCP) is attractive to design the guidance and control law since it can handle constraints while optimizing given performance index. However, the previous mentioned methods are not based on NOCP. Several methods based on NOCP have been designed to track the nominal solution which is computed offline. Neighboring optimal control (NOC) calculates feedback control by optimizing the second-order performance index, which calculates and stores the gain matrix at each time instance off-line and extracts gain matrix using interpolation on-line [12]. Pontani et al [13] proposed variable-time-domain neighboring optimal guidance (VTD-NOG) that avoids numerical difficulty caused by the singularity of gain matrices at terminal time. Zheng [14, 15] proposed the backward sweeping algorithm from geometric point of view, for both fixed terminal time and free terminal time of low-thrust transfer problems. Di Lizia et al [16, 17] designed the high-order NOC control law, by using high-order Taylor series automatically achieved by differential algebra around the nominal trajectory. Besides, model predictive control (NMPC) which employs the iterative and finite-horizon optimization strategy has been applied to design the controller. Based on orbit averaging techniques, Gao designed NMPC [18] to track the mean orbit elements. Huang et al [19] proposed a NMPC strategy using differential transformation based optimization method to track the nominal trajectory. However, the tracking approaches is the overdependence on the nominal profile. For example, the bang-bang thrust sequence is assumed to be unchanged under perturbations when using the NOC method [14]. Also, these techniques lack the operational flexibility since the trajectory is restricted to the vicinity of nominal solution. In order to overcome these drawbacks, the algorithms that enable the spacecraft to re-compute the entire nominal trajectory on-line at the beginning of each guidance interval is attractive. There have been several attempts to design the efficient algorithms. Wang [20] proposed to use convex programming to calculate fuel-optimal spacecraft trajectory. Pesch [21, 22] used multiple shooting method to re-compute the trajectory for general optimal control problems.

In this work, model Predictive Static Programming (MPSP), an optimal control design technique that combines the philosophy of model predictive control and approximate dynamic programming [23], is designed for low-thrust neighboring control law. The innovation of this method has three aspects [24]. Firstly, it successfully converts a dynamic programming problem to a static programming problem, and thus it requires only a static costate vector for the update of the control profile. Secondly, the symbolic costate vector enables a closed-form solution, which reduces computational load. Thirdly, the sensitivity matrix (SM) which is necessary for the calculation of the static costate vector can be computed recursively. These advantages promote wide applications of the MPSP technique, e.g., terminal guidance [25], reentry guidance [26], and lunar landing guidance [27], etc. Some variations of the MPSP technique have also been proposed to enhance the algorithm performance. For example, the generalized MPSP [24]
formulates the problem in continuous-time framework, which does not require any discretization process to begin with. Quasi-Spectral MPSP [28] expresses the control profile as a weighted sum of basis functions, enabling the method to optimize only a set of coefficients instead of optimizing the control variable at every grid point. However, most works assume continuity of the control profile, which impedes its application for low-thrust transfer missions with bang-off-bang control.

Considering that MPSP technique is an inherent Newton-type method that requires a good initial guess solution [29], MPSP as a potential neighboring control law for low-thrust transfer problems is investigated in this work. Firstly, the fuel-optimal low-thrust problem is stated in Cartesian coordinates, where the necessary conditions are formulated based on Pontryagin minimum principle (PMP). The fuel-optimal solution is used as the nominal solution solved by an indirect method. Secondly, inspired by the natural feedback controller given by PMP, the unconstrained costate variable related to the velocity is used as new control variable in MPSP design. In order to ensure the continuity of the switching function at switching points, dynamical equations are augmented by the mass costate equation. Thirdly, SM is recursively calculated using analytical derivative, where SM at switching points is compensated based on calculus of variations. Since SM discontinuity would result in discontinuity of discrete control sequence, the control profile is represented by the combination of Fourier basis functions and corresponding weights, where the weights are initialized based on nominal trajectory using least square method, and updated using Newton’s method. Two-loop MPSP algorithm structure is designed for both small and large perturbations, where Newton’s method and continuation are implemented in inner and outer loops respectively. The presented MPSP technique is successfully applied to bang-off-bang control for the first time in literature, without resorting to the additional optimization solver. Several numerical simulations are conducted, showing the effectiveness of the proposed method, so enhancing mission flexibility.

This paper is structured as follows: Section 2 states the control problem by using MPSP method. Section 3 depicts the detailed MPSP guidance design. Section 4 presents numerical simulations for a CubeSat mission to an asteroid. Conclusions are given in Section 5.

II. Problem Statement

A. Equations of Motion

This work considers the heliocentric phase of an interplanetary transfer mission. The restricted two-body problem is employed, where the spacecraft subjects to the gravitational attraction of the Sun. The spacecraft natural motion consists of Keplerian orbits around the Sun, corresponding to the equation of motion [30]

\[
\ddot{r} + \frac{\mu}{r^3}r = 0
\]
where $\mathbf{r}$ is the spacecraft position vector relative to the center of the Sun and $\mu$ is the gravitational parameter. When the low-thrust engine is considered, Eq. (1) is modified as

$$
\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}) \Rightarrow \begin{pmatrix}
\dot{\mathbf{r}} \\
\dot{\mathbf{v}} \\
\dot{\mathbf{m}}
\end{pmatrix} = \begin{pmatrix}
v \\
g(\mathbf{r}) + \frac{T_{\text{max}}}{m} \mathbf{\alpha} \\
-\mathbf{A} \dot{\mathbf{r}}
\end{pmatrix}
$$

where $g(\mathbf{r}) = -\mu \mathbf{r} / \mathbf{r}^3$, $\mathbf{r} := [x, y, z]^T \in \mathbb{R}^3$ and $\mathbf{v} := [v_x, v_y, v_z]^T \in \mathbb{R}^3$ are the gravitational vector field, the spacecraft position vector, and its velocity vector, respectively; $m$ is the spacecraft mass, $T_{\text{max}}$ is the maximum thrust magnitude, $c = I_{\text{sp}}g_0$ is the exhaust velocity ($I_{\text{sp}}$ is the engine specific impulse, $g_0$ is the gravitational acceleration at sea level), $\mathbf{u}$ is the thrust throttle factor, $\mathbf{\alpha}$ is the thrust pointing vector. The state vector is $\mathbf{x} = [\mathbf{r}, \mathbf{v}, m] \in \mathbb{R}^7$. Both $T_{\text{max}}$ and $c$ are assumed constant during flight.

### B. Fuel-Optimal Problem

In this work, fuel-optimal low-thrust trajectory is employed as the reference trajectory. The corresponding performance index is

$$
J = \frac{T_{\text{max}}}{c} \int_{t_0}^{t_f} u \, dt
$$

where $t_0$ and $t_f$ are initial and terminal time instants, both fixed. The initial state is known, i.e., $\mathbf{x}(t_0) = \mathbf{x}_0$. For the interplanetary mission to the asteroid, the fixed terminal constraint is considered, as

$$
\mathbf{x}(t_f) = \mathbf{x}_f
$$

The inequality constraint for thrust throttle factor $u$ is

$$
0 \leq u \leq 1
$$

The Hamiltonian function reads [12]

$$
H = \frac{T_{\text{max}}}{c} u + \lambda^T \mathbf{r} + \lambda^T \left[ g(\mathbf{r}) + \frac{T_{\text{max}}}{m} \mathbf{\alpha} \right] - \lambda_m \frac{u T_{\text{max}}}{c}
$$
where $\lambda = [\lambda_r, \lambda_v, \lambda_m]$ is the costate vector associated with $x$. Dynamical equations of $\lambda$ are

$$
\dot{\lambda} = -\left( \frac{\partial H}{\partial x} \right)^\top = \begin{pmatrix}
\dot{\lambda}_r \\
\dot{\lambda}_v \\
\dot{\lambda}_m
\end{pmatrix} = \begin{pmatrix}
-G^\top \lambda_v \\
-\lambda_r \\
uT_{\text{max}}/m^2 \lambda_v \alpha
\end{pmatrix}
$$

(7)

where $G = \partial g(r)/\partial r$. Since the final mass is free, there exists

$$
\lambda_m(t_f) = 0
$$

(8)

According to PMP, the optimal thrust direction is along the opposite direction of the primer vector $\lambda_v$, as

$$
\alpha^* = -\frac{\lambda_r}{\lambda_v}, \quad \text{if} \quad \lambda_v \neq 0
$$

(9)

substituting Eq. (9) into Eq. (6) yields

$$
H = \lambda_r^\top v + \lambda_v^\top g(r) + \frac{uT_{\text{max}}}{c} S
$$

(10)

where the switching function $S$ is defined as

$$
S = -\lambda_r \frac{c}{m} - \lambda_m + 1
$$

(11)

The optimal $u^*$ is governed by $S$ through

$$
u^* = \begin{cases}
0, & \text{if} \quad S > 0 \\
1, & \text{if} \quad S < 0
\end{cases}
$$

(12)

which is a bang-off-bang control type, forming the thrust sequence.

The fuel-optimal problem is solved by indirect method, which is to find $\lambda_0$ that (together with $x_0$) allows integrating Eqs. (2) and (7) with the control law in Eqs. (9) and (12) and verifies the terminal constraints (4) and (8) [3]. Singular thrust arcs are not considered here since they have been shown to be non-optimal in general [31]. Once the optimal $\alpha^*(t)$ and $u^*(t)$ are determined, the spacecraft trajectory can be generated by integrating Eqs. (2) and (7).

C. MPSP Dynamics and Control

In real world flight, disturbances or new mission requirements need the spacecraft to have the capability update the control sequence automatically. The guidance scheme based on Model predictive static programming (MPSP) is
of interest [23]. However, the fuel-optimal problem is the optimal control problem with control constraint. MPSP cannot be applied to it directly, since MPSP is originally designed for the unconstrained problem [23]. In this work, the augmented dynamics and the new control variable are proposed.

Let \( x^*(t) \) and \( \lambda^*(t) \) denote the reference state and costate profiles, and let \( x(t) \) and \( \lambda(t) \) be the associated, off-nominal profiles. Let \( \Delta \lambda(t) \) be the costate deviation, the two functions [14]

\[
\begin{align*}
  u(x, \lambda^* + \Delta \lambda) &= 1 - \text{Sgn}[S(x, \lambda^* + \Delta \lambda)] \\
  \alpha(x, \lambda^* + \Delta \lambda) &= -\frac{\lambda_v^* + \Delta \lambda_v}{\|\lambda_v^* + \Delta \lambda_v\|} \quad \text{if } \|\lambda_v^* + \Delta \lambda_v\| \neq 0
\end{align*}
\]

(13)

define the feedback controller associated with \( x \) at time instant \( t \), where Sgn function is defined as

\[
\text{Sgn}(z) = \begin{cases} 
  0, & \text{if } z > 0 \\
  1, & \text{if } z < 0
\end{cases}
\]

(14)

In this work, the unconstrained costate vector is used as a new control variable for MPSP controller design. This idea also has been utilized in NOC design [14] and Lyapunov guidance design [32]. Notice from Eqs. (11) and (13) that, costate variables which affect \( u \) and \( \alpha \) are \( \lambda_v \) and \( \lambda_m \). However, only \( \lambda_v \) is seen as the new control variable, based on three facts. Firstly, it can be seen from \( \dot{\lambda}_m \) in Eq. (7) that \( \lambda_v \) and \( \lambda_m \) are dependent, and \( \lambda_m \) profile is determined by \( \lambda_v \). Secondly, if \( \lambda_m \) is also used as a control variable, \( \dot{\lambda}_m \) cannot be expressed by Eq. (7). The derivative of the switching function \( S \) in Eq. (11)

\[
\dot{S} = -\frac{\lambda_v}{m} - \lambda_v \frac{uT_{\max}}{m^2} - \dot{\lambda}_m
\]

(15)

would hardly be continuous because of the presence of \( u \). On the other hand, if \( \dot{\lambda}_m \) is remained as Eq. (7), \( \dot{S} \) becomes simply to be

\[
\dot{S} = -\frac{\lambda_v}{m}
\]

(16)

which is naturally implicitly dependent on \( u \). Thirdly, the second-order differential of \( \lambda_m \) w.r.t time is not continuous due to discontinuity of \( u \) in \( \dot{\lambda}_m \). Since basis functions are used to approximate the control profile in this work, these may not be appropriate to efficiently capture the discontinuity [33].
Thus, the dynamical equations used for MPSP algorithm design in this work are

\[ \dot{X} = \mathcal{F}(t, X, U) \Rightarrow \begin{pmatrix} \dot{r} \\ \dot{\psi} \\ \dot{m} \\ \dot{\lambda}_m \end{pmatrix} = \begin{pmatrix} v \\ g(r) - \frac{u T_{\text{max}}}{m \lambda_v} \\ -\frac{T_{\text{max}}}{c} \\ -u T_{\text{max}}/m^2 \lambda_v \end{pmatrix} \]  

(17)

where \( X = [x, \lambda_m] \in \mathbb{R}^8 \), \( U = \lambda_v \in \mathbb{R}^3 \), and optimal thrust direction Eq. (9) is embedded into Eq. (17). The relationships between thrust angles and \( U \) are

\[ \begin{align*}
\alpha &= \arctan(\frac{\lambda_{v2}}{\lambda_{v1}}) \\
\beta &= \arcsin(\frac{\lambda_{v3}}{||\lambda_v||_2})
\end{align*} \]  

(18)

where \( \alpha \in [0, 360^\circ] \) is the in-plane angle, \( \beta \in [-180^\circ, 180^\circ] \) is the out-of-plane angle and \( \lambda_{v,i} \) is the \( i \)-th element of \( \lambda_v \). Once \( X(t) \) and \( U(t) \) are determined, the profile of \( S \) is decided automatically, which then determines the switching time and thrust sequence. In this work, the task of the MPSP algorithm is to determine a suitable \( \Delta U \) and \( \Delta \lambda_{m0} \) such that the trajectory of the spacecraft obtained by integrating Eq. (17) satisfies the required boundary conditions Eqs. (4) and (8), while conducting bang-off-bang control.

### III. MPSP Algorithm Design

**A. Sensitive Matrix Calculation**

Different from problems with continuous control profile, dynamical discontinuity happens at switching points. Thus, the trajectory cannot be treated as a whole. In this work, the trajectory is split into multiple segments with switching points located at the boundary of each segment. The time instants at the boundary of each segment are \( \{t_0, t_1, \ldots, t_M\} \), where \( M \) is the number of total segments, \( t_0 \) and \( t_M = t_f \) are initial and final time, respectively, and \( t_k, k = 1, 2, \ldots, M - 1 \) are the switching times. Let \( \{t_{k-1}^0, t_k^0, t_k^1, \ldots, t_{k+1}^N\} \), \( k = 0, 1, \ldots, M - 1 \) denote an evenly-spaced time grid within \([t_k, t_{k+1}]\), where \( t_{k-1}^0 \) and \( t_k^0 \) are the time instants across the impulse. For \( k \)-th segment, \( N_k \) is the minimum number of points, such that the time step is just less than prescribed maximum time step \( h_{\text{max}} \). Note also that \( t_{k+1}^N = t_{k+1}^0 \). To ease notation, \( N_k \) is denoted \( N \). Suppose there is no impulse at initial time, then \( t_k^0 = t_0^0 = t_0 \).

Consider the \( k \)-th time interval \([t_k, t_{k+1}]\), the discrete system dynamics and the output can be written as

\[ X_k^{i+1} = F_k^i(X_k^i, U_k^i) \quad Y_k^i = O(X_k^i) \]  

(19)
where $Y^i_k$ is the output at the $i$th step, which is the function of $X^i_k$. $F'_k(X^i_k, U^i_k)$ can be obtained using standard integration formula, such as the Euler method [23]. High-order integration results in higher accuracy, but larger computational load. In this work, the standard 4th order Runge–Kutta integration is used, see the Appendix for the computation of Eq. (19).

The primary objective is to obtain an updated control history $U^i_k$ and initial state $X^0_0$ such that the output $Y$ at terminal time, i.e., $Y^N_{M-1}$, reaches to the desired value $Y_d$. Writing $Y^N_{M-1}$ about $Y_d$ in Taylor series expansion and neglecting high-order terms, the error at the terminal output $\Delta Y^N_{M-1} = Y^N_{M-1} - Y_d$ is approximated as

$$\Delta Y^N_{M-1} = dY^N_{M-1} = \left[ \frac{\partial Y^N_{M-1}}{\partial X^N_{M-1}} \right] dX^N_{M-1}$$  

(20)

The deduction of $dX^{i+1}_k$ should consider whether the impulse happens or not. For the interval $[\bar{t}^i_k, \bar{t}^{i+1}_k]$, $i = 1, 2, \cdots, N - 1$ where there is no switching point, according to Eq. (19), there exists

$$dX^{i+1}_k = \left[ \frac{\partial F^i_k}{\partial X^i_k} \right] dX^i_k + \left[ \frac{\partial F^i_k}{\partial U^i_k} \right] dU^i_k$$  

(21)

where $dX^i_k$ and $dU^i_k$ are full differentials of state and control vectors.

For the interval $[\bar{t}^0_k, \bar{t}^1_k]$ that contains an impulse, we have

$$dX^1_k = \frac{\partial F^0_k}{\partial X^0_k} dX^0_k + \frac{\partial F^0_k}{\partial U^0_k} dU^0_k$$

$$= \frac{\partial F^0_k}{\partial X^0_k} dX^0_k + \frac{\partial F^0_k}{\partial U^0_k} dU^0_k$$

$$= \frac{\partial F^0_k}{\partial X^0_k} dX^0_k + \frac{\partial F^0_k}{\partial U^0_k} dU^0_k$$

(22)

where $dU^0_k = dU^0_k = dU^0_k$ due to thrust angle continuity, $dX^0_k = dX^0_k$, and

$$\frac{\partial F^0_k}{\partial X^0_k} = \frac{\partial F^0_k}{\partial X^0_k} + \frac{\partial F^0_k}{\partial U^0_k} dU^0_k$$

$$\frac{\partial F^0_k}{\partial U^0_k} = \frac{\partial F^0_k}{\partial X^0_k} + \frac{\partial F^0_k}{\partial U^0_k} dU^0_k$$

(23)

The expressions of $\partial X^0_k / \partial X^0_k$ and $\partial X^0_k / \partial U^0_k$ are [34]

$$\left[ \frac{\partial X^0_k}{\partial X^0_k} \right] = I + (X^0_k - X^0_k) S_X / S$$

(24)
\[
\frac{\partial X^0_k}{\partial U^0_k} = \left( \dot{X}^0_k - X^0_k \right) S_X / \dot{S}
\]  

(25)

where \(S_X\) and \(S_U\) are row vectors that are the partial derivative of switching function \(S\) w.r.t \(X\) and \(U\) respectively, and \(\dot{S}\) is calculated according to Eq. (16).

Combining Eq. (21) with Eq. (22) yields the uniform form of \(dX^i_{k+1}\) as

\[
dX^i_{k+1} = \frac{\partial F^i_k}{\partial X^i_k} dX^i_k + \frac{\partial F^i_k}{\partial U^i_k} dU^i_k, \quad i = 0, \ldots, N - 1
\]  

(26)

Substituting Eq. (26) into Eq. (20) yields

\[
dY^N_{M-1} = \left[ \frac{\partial Y^N_{M-1}}{\partial X^N_{M-1}} \right] \left[ \frac{\partial F^{N-1}_{M-1}}{\partial X^{N-1}_{M-1}} \right] dX^{N-1}_{M-1} + \left[ \frac{\partial F^{N-1}_{M-1}}{\partial U^{N-1}_{M-1}} \right] dU^{N-1}_{M-1}
\]  

(27)

Similarly, the state differential at time step \((N-1)\) can be expanded in terms of state and control differentials at time step \((N-2)\). Next, \(dX^{N-2}_{M-1}\) can be expanded in terms of \(dX^{N-3}_{M-1}\) and \(dU^{N-3}_{M-1}\). For \((M-1)\)th segment, this process is continued to \(dX^0\). Notice that \(dX^N_{M-2} = dX^0_{M-1}\), the same process is continued at \((M-2)\)th segment. Extending the process until \(X^0\), one obtains

\[
dY^N_{M-1} = AdX^0 + B^0_dU^0 + B^1_dU^1 + B^{N-1}_{M-1}dU^{N-1}_{M-1}
\]

(28)

where the compact form of coefficients \(A\) and \(B^i_k\) in Eq. (28) are

\[
A = \left[ \frac{\partial Y^N_{M-1}}{\partial X^N_{M-1}} \right], \quad B^i_k = \left[ \frac{\partial F^i_k}{\partial X^i_k} \right] \left[ \frac{\partial F^{N-1}_{M-1}}{\partial U^{N-1}_{M-1}} \right]
\]  

(29)

The presented MPSP is desirable because the computation of the sensitivity matrix \(B^i_k\) can be reduced to an iterative calculation. Define

\[
B^N_{M-1,0} = \left[ \frac{\partial Y^N_{M-1}}{\partial X^N_{M-1}} \right]
\]  

(30)

there exists

\[
B^i_{k,0} = B^i_{k,0} \left[ \frac{\partial F^{i+1}_k}{\partial X^i_k} \right], \quad B^i_k = B^i_{k,0} \left[ \frac{\partial F^i_k}{\partial U^i_k} \right]
\]  

(31)
**B. Control Representation and Update**

For applications with continuous control profile, discrete control sequence works due to the continuity of SM. However, $B_i^j$ is discontinuous before and after impulse, resulting in the discontinuity of the thrust angle sequence if the control profile is discretized, which is meaningless from the physical point of view. In this work, the control profile is expressed by basis functions. The advantages lies in two facets. Firstly, the continuity of the thrust angle profile can be ensured automatically due to the continuity of basis functions. Secondly, the time derivative of the switching function $S$ can be calculated analytically. The control is expressed as

$$U_i^k(\eta) = P_i^k(\eta)\epsilon$$  \hspace{1cm} (32)

where $\epsilon$ is the weight to the basis functions, and

$$P(\eta) = \begin{bmatrix} h^T(\eta) \\ h^T(\eta) \\ \vdots \\ h^T(\eta) \end{bmatrix}$$  \hspace{1cm} (33)

$h(\eta)$ is the collection of different orders of basis functions. The $\eta$ range is determined by the basis functions chosen. The linear projection of $\eta$ w.r.t time $t$ is used as

$$\eta = \frac{\eta_f - \eta_0}{t_f - t_0} (t - t_0) + \eta_0$$  \hspace{1cm} (34)

From Eq. (32), the update of the control profile is achieved by updating $\epsilon$. The differential of Eq. (32) is

$$dU_i^k = P_i^k(\eta)d\epsilon$$  \hspace{1cm} (35)

Substituting Eq. (35) into Eq. (28) yields

$$dY_{M-1}^{N} = AdX_0^0 + \sum_{k=0}^{M-1} \sum_{i=0}^{N-1} B_i^k P_i^k d\epsilon$$  \hspace{1cm} (36)

Since the control used is not directly linked to the thrust throttle factor $u$, the solution in the neighborhood of the reference solution is preferred. The performance index is set to

$$J = \frac{1}{2} d\epsilon^T R_\epsilon d\epsilon + \frac{1}{2} (dX_0^0)^T R_0 dX_0^0$$  \hspace{1cm} (37)
Denote \( B_v = \sum_{k=0}^{M-1} \sum_{i=0}^{N-1} B^i_k P^i_k \), the augmented performance index reads

\[
\hat{J} = \frac{1}{2} \sum d\epsilon^T R \epsilon + \frac{1}{2} \left( \sum dX_0^0 \right)^T R_0 dX_0^0 + p^T \left( \sum dY^N_{M-1} - A dX_0^0 - B_v d\epsilon \right)
\]  

(38)

where \( p \) is the associated static costate vector. The optimal conditions read

\[
\begin{align*}
\left( \frac{d\hat{J}}{d(d\epsilon)} \right)^T &= R \epsilon - B_v^T p = 0 \\
\left( \frac{d\hat{J}}{d(dX_0^0)} \right)^T &= R_0 dX_0^0 - A^T p = 0
\end{align*}
\]

(39)

Substituting Eq. (39) into Eq. (36) yields

\[
p = \left( A R_0^{-1} A^T + B_v R_0^{-1} B_v^T \right)^{-1} \sum dY^N_{M-1}
\]

(40)

and substituting Eq. (40) into Eq. (39) yields the Newton direction \( d\epsilon \) and \( dX_0^0 \) as

\[
\begin{align*}
d\epsilon &= R^{-1}_v B_v^T p \\
dX_0^0 &= R_0^{-1} A^T p
\end{align*}
\]

(41)

Since the spacecraft initial position, velocity and mass are known and fixed, \( d\lambda m_0 \) is to be solved. \( \epsilon \) and \( \lambda m_0 \) are updated at \( j \)th iteration as

\[
\begin{align*}
\epsilon_{j+1} &= \epsilon_j - \kappa d\epsilon_j \\
\lambda m_{0,j+1} &= \lambda m_{0,j} - \kappa d\lambda m_{0,j}
\end{align*}
\]

(42)

where \( \kappa \) is the Newton step length. Then the updated \( U(t) \) and \( X(t) \) are determined through calculating Eq. (32) and integrating Eq. (17).

However, particular attention needs to be paid for Newton step length \( \kappa \). If \( \kappa \) is too large, the thrust sequence may be apparently changed, which amplifies undesired terminal error and further deteriorates algorithm performance. On the other hand, the unconstrained and changeable thrust sequence contributes toward enlarging the convergence domain. Therefore, rigorous strategy for \( \kappa \) selection should be designed to maintain the algorithm stability and simultaneously enlarge the convergence domain. In this work, the thrust sequence at each iteration is checked and restricted. Denote \( N_{seg,i} \) and \( N_{seg,ref} \) as the sum of thrust segments and coast segments for the trajectory at \( i \)th iteration and reference trajectory, respectively. For \( (i+1) \)th iteration, the restriction on \( N_{seg,i+1} \) is implemented such that \( \| N_{seg,i+1} - N_{seg,ref} \| \leq N_{seg,tol} \), where \( N_{seg,tol} \) is the tolerance for the varied segments. Otherwise, \( \kappa \) is reduced.
C. Nominal Solution Generation

The fixed-time fuel-optimal open-loop problem can be formulated into a two-point boundary value problem (TPBVP) and it requires to search a zero of the shooting function associated with TPBVP [3]. In this work, the method that combines analytic derivatives, switching detection technique and numerical continuation is applied to find the fuel-optimal low-thrust trajectory[3], which is used as nominal solution. There is no need to assign control structure a priori, and it is also useful in cases where very low-thrust accelerations are used in highly nonlinear vector fields.

The nominal discrete control sequence at \( k \)th evenly time grid is denoted as \( U_{k,\text{ref}} \). Collecting all discrete points yields the following linear algebraic equation

\[
A\epsilon_0 = B
\]

where

\[
A = [P_1, P_2, \cdots, P_N]
\]

\[
B = [U_{1,\text{ref}}, U_{2,\text{ref}}, \cdots, U_{N,\text{ref}}]
\]

The least-square solution is used as nominal solution, as

\[
\epsilon_0 = (A^\top A)^{-1}A^\top b
\]

D. Implementation

The MPSP algorithm for bang-off-bang low-thrust transfers requires to detect the switching time accurately, which is based on two reasons. Firstly, if the switching time is not detected, the integration error will be accumulated around the switching points, which deteriorates the performance of the Newton’s method. Secondly, SM is discontinuous as shown in Eqs. (21) and (22), which requires to detect switching time for accurate calculation of SM. The switching detection technique is embedded into four-order Runge-Kutta fixed step trajectory integral scheme. The detection is active as soon as the switching function Eq. (11) traverses zero at time interval \([t_k, t_{k+1}]\). The bisection method is used to find the switching time \( t_{\text{sw}} \in [t_k, t_{k+1}] \) such that the absolute value of switching function is within the tolerance \( 10^{-12} \).

Based on the techniques proposed above, a two-loop MPSP algorithm is designed consisting of inner-loop and outer-loop parts. The inner-loop algorithm is illustrated in Algorithm 1, which is to update \( \epsilon \) and \( \lambda_{m,0} \) for the input boundary conditions using Newton’s method. In inner-loop algorithm, the Newton’s method is implemented only when the L2-norm of terminal error at \( j \)th iteration is less than a maximum error tolerance \( \Delta_{\text{max}} \), i.e., \( \|Y_{M-1}^N\|_2 \leq \Delta_{\text{max}} \). \( \Delta_{\text{max}} \) is the conservative value indicating the failure of the iteration or the encounter of large perturbations. \( \text{Sign} \) is used to label the success (\( \text{Sign} = 1 \)) or failure (\( \text{Sign} = 0 \)) of the inner-loop part. The failure occurs when the
terminal error exceeds the tolerance or the step length is small enough. The outer-loop MPSP algorithm is shown in Algorithm 2, which is triggered when \( \text{Sign} = 0 \) is returned. In this case, the continuation from nominal conditions to perturbed conditions is conducted. Denote \( C_{\text{ref}} \) as reference boundary conditions, \( C_{\text{per}} \) as perturbed conditions and \( \tau \) as continuation parameter. Starting from \( \tau = 0 \) which corresponds to \( C_{\text{ref}} \), continuation proceeds until \( \tau = 1 \) which corresponds to \( C_{\text{per}} \). At each step, the inner-loop MPSP algorithm is applied to find the solution corresponding to the conditions

\[
C_\tau = (1 - \tau)C_{\text{ref}} + \tau C_{\text{per}}
\]

(47)

Beside, the \( N_{\text{seg},\text{tol}} \) used in the inner-loop algorithm is initially set to be 0 at the outer-loop algorithm. Thus, the MPSP algorithm tries to find the solution with the same thrust sequence as the nominal solution first. The value of \( N_{\text{seg},\text{tol}} \) increases once the inner-loop MPSP algorithm fails.

**Algorithm 1 Inner-Loop MPSP Algorithm**

**Require:** Boundary conditions \( C_\tau \), weight of control sequence \( \varepsilon \), \( \Delta_{\text{max}} \), \( N_{\text{seg},\text{tol}} \), reference solution.

**Ensure:** The updated \( \varepsilon \), \( \lambda_{m,0} \) and label \( \text{Sign} \)

1: Integrate the trajectory using initial condition \( X_0 \) and \( \varepsilon \).
2: Set \( \text{Sign} = 1, i = 0 \).
3: while The terminal error does not satisfy requirement do
4: if \( \|Y_{N-1}^i\|_2 \geq \Delta_{\text{max}} \) then
5: \( \text{Sign} = 0 \). Return.
6: end if
7: Calculate \( A \) and \( B \), matrix in Eq. (29) and the static costate vector \( p \) in Eq. (40). Calculate Newton direction \( \frac{\partial f_j}{\partial \varepsilon} \) and \( \frac{\partial f_j}{\partial \lambda_{m,0,j}} \). Set initial \( \kappa = 1 \).
8: while \( \kappa > 1/2 \) do
9: \( \varepsilon_{\text{old}} := \varepsilon \) and \( \lambda_{m,0,\text{old}} := \lambda_{m,0} \).
10: Update \( \varepsilon \) and \( \lambda_{m,0} \). Integrate dynamical equations to obtain the trajectory. Get \( N_{\text{seg},i+1} \). \( i := i + 1 \)
11: if \( \|N_{\text{seg},i+1} - N_{\text{seg},\text{ref}}\| \leq N_{\text{seg},\text{tol}} \) then
12: Save the updated control. Break.
13: else
14: \( \kappa := \kappa / 2 \). \( \varepsilon := \varepsilon_{\text{old}} \). \( \lambda_{m,0} := \lambda_{m,0,\text{old}} \).
15: if \( \kappa \leq 1/2^5 \) then
16: \( \text{Set Sign} = 0 \). Return.
17: end if
18: end if
19: end while
20: end while

**IV. Numerical Simulations**

**A. Fuel-optimal Trajectory**

An interplanetary Cubesats mission to the asteroid 99942 Apophis is considered. The related physical constants are listed in Tab. 1. The initial mass \( m_0 \), maximum thrust magnitude \( T_{\text{max}} \) and specific impulsive \( I_{sp} \) are set to be 25kg, \( 1.5 \times 10^{-3} \)N and 3000s, respectively. The departure epoch is October 1st, 2020, the arrival epoch is December
Algorithm 2 Outer-Loop MPSP Algorithm

1: Solving the fuel-optimal low-thrust transfer problem using indirect method [3]. The initial weight \( e_0 \) is calculated using least square method.
2: Calculate perturbed condition or thruster parameters.
3: Set error tolerance \( \Delta_{max} \), continuation parameter variation \( \delta \tau \) and \( N_{seg,tol} = 0 \).
4: Apply inner-loop MPSP algorithm. Denote the returned label of success as \( Sign1 \).
5: if \( Sign1 = 0 \) then
6: while 1 do
7: Set \( \tau = \delta \tau, \tau_{old} = 0 \).
8: while \( \delta \tau > 0 \) do
9: Calculate the perturbed condition for current \( \tau \).
10: Apply inner-loop MPSP algorithm. Denote the returned label of success as \( Sign2 \).
11: if \( Sign2 = 0 \) then
12: \( \delta \tau := \delta \tau / 2 \).
13: else
14: Save the solution as new initial guess solution for the next iteration.
15: Set \( \delta \tau := \min(1 - \tau, 2\delta \tau), \tau_{old} = \tau \)
16: end if
17: \( \tau = \tau_{old} + \delta \tau \).
18: if \( \delta \tau \leq 0.01 \) and \( \tau_{old} \neq 1 \) then
19: Break
20: end if
21: end while
22: if \( \delta \tau = 0 \) then
23: break
24: else
25: \( N_{seg,tol} := N_{seg,tol} + 2 \)
26: if \( N_{seg,ref} - N_{seg,tol} < 0 \) then
27: Print failure information; break
28: end if
29: end if
30: end while
31: else
32: Save the solution.
33: end if
Table 1 Physical constants.

| Physical constant | Values                  |
|-------------------|-------------------------|
| Mass parameter $\mu$ | $1.327124 \times 10^{11} \text{km}^3/\text{s}^2$ |
| Gravitational field, $g_0$ | $9.80655 \text{m/s}^2$ |
| Length unit, LU | $1.495979 \times 10^8 \text{km}$ |
| Time unit, TU | $5.022643 \times 10^6 \text{s}$ |
| Velocity unit, VU | $29.784692 \text{km/s}$ |
| Mass unit, MU | $25 \text{kg}$ |

Table 2 Boundary Conditions.

| Boundary Condition | Values                                                                 |
|--------------------|------------------------------------------------------------------------|
| Initial position vector (LU) | $r_0 = [1.001367, 0.140622, -6.594513 \times 10^{-6}]^T$ |
| Initial velocity vector (VU) | $v_0 = [-0.155386, 0.986258, -4.827818 \times 10^{-5}]^T$ |
| Terminal position vector (LU) | $r_f = [-1.044138, -0.122918, -0.018183]^T$ |
| Terminal velocity vector (VU) | $v_f = [0.222668, -0.875235, 0.051944]^T$ |

1st, 2023, and the departure position is Sun-Earth L2. The boundary conditions of the spacecraft at the initial and rendezvous time are listed in Tab. 2. The fuel-optimal solution is employed as the nominal solution which is solved by using indirect method [3]. The optimal transfer orbit is shown in Fig. 1, where the red line denotes the thrust segment, i.e., $u = 1$, while the blue dash line denotes the coast segment, i.e., $u = 0$. The variations of thrust throttle $u$, switching function $S$ and the mass $m$ w.r.t time are shown in Fig. 2. The optimal trajectory consists of five thrust segments and four coast segments, and the final mass of the spacecraft is 21.062kg.

In the following numerical simulations, Fourier basis polynomials with maximum 15th order are used to approximate the control sequence. The largest time step for each segment is set to be $h_{\text{max}} = 0.0005 \times t_f$. The convergence conditions to terminate the MPSP algorithm is such that the terminal position error $\|\Delta r_f\|_2 \leq 500\text{km}$, the terminal velocity error $\|\Delta v_f\|_2 \leq 0.1\text{km/s}$ and $|\lambda_{mf}| \leq 10^{-6}$. For other parameter setting, $\tau_0 = 0.5$, $\delta \tau = 0.5$ and $\Delta_{\text{max}} = 1$. All simulations are conducted under Intel Core i7-9750H, CPU@2.6GHz, Windows 10 system within MATLAB environment. Considering the paper length, the simulations with large number of random perturbations are not reported.

B. Perturbations on Initial Conditions

In this section, the proposed MPSP algorithm is tested by assuming the perturbations on the initial conditions. Different scales of perturbation magnitude for the single base perturbation are used for analysis. The random base perturbation is set to be as $\delta x_0 = [1.6712, -1.0659, -4.1460, 1.0876, 2.3763, 4.6091] \times 10^{-2}$. The 9 cases of different scales of perturbation magnitude are simulated, and the corresponding perturbed initial condition is $\delta \hat{x}_0 = x_0 + \kappa \delta x_0$.
Fig. 1  The fuel-optimal trajectory for boundary conditions in Tab. 2, where 'SEL2' denotes Sun-Earth L2 starting point and 'AST' denotes the asteroid position.

Fig. 2  The variations of thrust throttle $u$, switching function $S$ and mass $m$ w.r.t time corresponding to the fuel-optimal trajectory in Fig. 1.
where $\kappa \in [-3, -2.5, -2, -1, 1, 2, 2.5, 3, 3.5]$. The perturbation direction of the former 4 cases are opposite to that of the later 5 cases. The simulation results are summarized in Tab. 3 which gives the terminal errors, the Newton’s iteration in the inner-loop MPSP algorithm and the percentage of fuel increase w.r.t the corresponding optimal solutions. It can be observed that, the algorithm works successfully since the terminal errors are all within the tolerance.

The comparisons between the thrust angles of the converged MPSP solutions and the nominal thrust angles are shown in Fig. 3, where the variations of thrust angles remain to be smooth. The variations of $\alpha$ for cases 5-9 is more apparent than that of cases 1-4, while the $\beta$ oscillations for all cases remain in the vicinity of nominal solution. The comparisons between thrust sequences of the converged MPSP solutions and the corresponding fuel-optimal thrust sequences are shown in Fig. 4. Case 4 requires the minimum iterations. In this case, the outer-loop MPSP continuation process is not triggered. Case 9 requires the maximum iterations, since the thrust sequence is changed dramatically compared with nominal thrust sequence. From case 4 to 1, the optimal solutions gradually emerges new coast segments, but the obtained MPSP solutions remain the nominal thrust sequence. From case 5 to 9, the initial conditions are becoming tighter, and more thrust is required to drive the spacecraft to the target. The MPSP solutions and optimal solutions are shown a similar trend which gradually increases the thrust segments and reduces the coast segments. From Tab. 3, the largest increase of fuel consumption is the case 1, which is around 9%, while the minimum increase of fuel consumption is the case 4, which is only 0.16%. From case 5 to 9, even though the proposed algorithm requires more iterations for tighter initial conditions, the fuel consumption is nearly optimal. The differences between converged MPSP solutions and the nominal solution on coordinates are shown in Fig. 5. It is interesting to see that the the differences are symmetric for opposite direction of initial perturbations.
Fig. 4 The comparison of optimal thrust sequences and thrust sequences of MPSP solutions for cases in Tab. 3. The red dash line: optimal thrust sequence; Blue line: thrust sequences of MPSP solutions.

Fig. 5 Differences on coordinates between converged MPSP trajectories and the nominal trajectory for cases in Tab. 3.
Table 3  Perturbations on initial conditions.

| Case | $\kappa$ | $\| x_f - x(t_f) \|_2$ (km) | $\| v_f - v(t_f) \|_2$ (km/s) | Newton’s iteration | Fuel Increase (%) |
|------|----------|--------------------------|--------------------------|-------------------|------------------|
| 1    | −3       | 216.45                   | 4.20 × 10^{-5}           | 3.74 × 10^{-8}    | 40               | 9.01             |
| 2    | −2.5     | 28.77                    | 5.66 × 10^{-6}           | 3.11 × 10^{-9}    | 21               | 6.82             |
| 3    | −2       | 495.99                   | 9.04 × 10^{-5}           | 3.26 × 10^{-7}    | 5                | 2.64             |
| 4    | −1       | 328.34                   | 6.07 × 10^{-5}           | 1.28 × 10^{-7}    | 2               | 2.27             |
| 5    | 1        | 128.41                   | 2.43 × 10^{-5}           | 2.48 × 10^{-8}    | 40               | 0.16             |
| 6    | 2        | 9.24                     | 1.86 × 10^{-6}           | 1.09 × 10^{-9}    | 48               | 5.02             |
| 7    | 2.5      | 13.18                    | 2.55 × 10^{-6}           | 4.74 × 10^{-10}   | 67               | 1.11             |
| 8    | 3        | 40.11                    | 1.66 × 10^{-5}           | 1.82 × 10^{-7}    | 98               | 2.54             |
| 9    | 3.5      | 454.25                   | 8.13 × 10^{-5}           | 2.41 × 10^{-7}    | 135              | 1.94             |

C. Perturbations on Terminal Conditions

Different variations on the terminal positions are simulated to test the proposed method. The totally 8 perturbed terminal positions lie in the vertex of the cube where the reference position locates in the center of the cube, and the side of the cube is set to be 0.04 LU. The corresponding 8 cases are shown in Tab. 4. The simulation results including the terminal errors, the Newton’s iteration in the inner-loop MPSP algorithm and the percentage of fuel increase w.r.t the corresponding optimal solutions are shown in Tab. 5. From the results, it can be observed that all obtained terminal values are within the tolerance.

Fig. 6 represents the variations of thrust angles $\alpha$ and $\beta$ w.r.t time. It can be seen that the obtained angle variations are close to the nominal case. Since the terminal $x_f$ for cases 1-4 are farther than cases 5-8 w.r.t the Sun, the $\alpha$ starts more inward in cases 1-4 than cases 5-8. The similar trend is seen from $\beta$ variation. The comparison between the thrust sequences for the converged MPSP solutions and the corresponding optimal thrust sequences are shown in Fig. 7. It is nice to see that the converged MPSP solutions coincide with the optimal solutions well in most cases. For cases 1 and 3, optimal thrust sequences have more coast segments than nominal one. MPSP solutions in these cases prefer to maintain to be the same as the nominal thrust sequence. For cases 2, 4 and 8, MPSP solutions capture the main structure of optimal thrust sequences except some near-impulse thrust segments. In cases 5, 6 and 7, MPSP solutions perfectly coincide with the optimal thrust sequences. In cases 1, 3 and 7 where MPSP solutions remain the nominal thrust sequence, correspond to just 5-6 Newton’s iterations. In cases 2, 4, 5 and 8, MPSP requires around 30 Newton’s iterations since one less thrust segment is required with $N_{seg,tol} = 2$. In case 6, MPSP requires more iterations because two less coast segment is required with $N_{seg,tol} = 4$. However, for all cases, the fuel consumption is very close to the optimal solution. The minimum fuel consumption is the case 2, which is only 0.79% more than its optimal fuel consumption. The maximum fuel consumption is case 6, which is only 2.36% more than its optimal fuel consumption. The fuel consumption remains nearly optimal even though the thrust sequence is changed w.r.t the nominal thrust.
Table 4  Cases study for Perturbations on terminal positions.

| Case | $\delta x_f$ (LU) | $\delta y_f$ (LU) | $\delta z_f$ (LU) |
|------|------------------|------------------|------------------|
| 1    | 0.02             | 0.02             | 0.02             |
| 2    | 0.02             | 0.02             | -0.02            |
| 3    | 0.02             | -0.02            | 0.02             |
| 4    | 0.02             | -0.02            | -0.02            |
| 5    | -0.02            | 0.02             | 0.02             |
| 6    | -0.02            | 0.02             | -0.02            |
| 7    | -0.02            | -0.02            | 0.02             |
| 8    | -0.02            | -0.02            | -0.02            |

Table 5  Simulation results for perturbations on terminal positions.

| Case | $\|x_f - x(t_f)\|_2$ (km) | $\|y_f - y(t_f)\|_2$ (km/s) | $\|\lambda_m(t_f)\|$ | Newton’s iteration | Fuel Increase(%) |
|------|--------------------------|----------------------------|----------------------|-------------------|------------------|
| 1    | 7.98                     | $1.13 \times 10^{-6}$       | $8.35 \times 10^{-8}$ | 6                 | 1.33             |
| 2    | 12.71                    | $2.41 \times 10^{-6}$       | $2.49 \times 10^{-9}$ | 40                | 0.79             |
| 3    | 8.57                     | $1.60 \times 10^{-6}$       | $1.14 \times 10^{-9}$ | 6                 | 2.26             |
| 4    | 18.85                    | $3.10 \times 10^{-6}$       | $2.52 \times 10^{-8}$ | 45                | 1.01             |
| 5    | 288.30                   | $5.05 \times 10^{-5}$       | $1.31 \times 10^{-7}$ | 32                | 1.26             |
| 6    | 16.28                    | $2.82 \times 10^{-6}$       | $2.13 \times 10^{-9}$ | 71                | 2.36             |
| 7    | 55.82                    | $1.00 \times 10^{-5}$       | $1.26 \times 10^{-9}$ | 5                 | 1.28             |
| 8    | 53.00                    | $8.75 \times 10^{-6}$       | $1.57 \times 10^{-9}$ | 40                | 2.08             |

sequence. The difference between nominal solution and the converged MPSP solutions are shown in Fig. 8. The $x$ difference for case 1-4 and case 5-8 are near symmetric. The $y$ differences are also shown similar symmetry except the last 200 days. The $z$ differences are shown more complexity and the magnitude of the differences tends to amplify.

D. Perturbations on Thruster Parameters

The perturbations on the thruster parameters are simulated. Specifically, the perturbations on the $T_{\text{max}}$ are tested. The percentage of the perturbation w.r.t the nominal solution is set to be $\eta = [-10\%, -6\%, -3\%, 3\%, 6\%, 10\%]$, which corresponds to 6 simulation cases as shown in Tab. 6. It is assumed that the percentage of the error of $T_{\text{max}}$ remains to be the same throughout the flight. The simulation results are summarized in Tab. 6 which gives the terminal errors, the Newton’s iteration in the inner-loop MPSP algorithm and the percentage of fuel increase w.r.t the corresponding optimal solutions. For all cases, MPSP algorithm converges successfully.

In Fig. 9, it is nice to see from that the variations of $\alpha$ and $\beta$ are in the vicinity of nominal values. Fig. 10 illustrates the comparison between the optimal thrust sequences and thrust sequences of the MPSP solutions. The thrust sequences coincide well even when the thrust sequence is changed. The increase of the fuel consumption w.r.t
Fig. 6 The in-plane angle $\alpha$ and out-of-plane angle $\beta$ of the converged MPSP trajectories for variations of terminal positions in Tab. 4.

Fig. 7 The comparison of optimal thrust sequences and thrust sequences of MPSP solutions for cases in Tab. 4. The red dash line: optimal thrust sequences; Blue line: thrust sequences of MPSP solutions.
the corresponding fuel-optimal solutions are negligible. As expected, the maximum Newton’s iteration occurs in case 1 since the variation of the thrust sequence is the largest. It is also noticed that it only requires 4 to 5 iterations when the thrust sequence remains to be the same as the nominal one. Fig. 11 depicts the trajectory differences between converged MPSP trajectories w.r.t the nominal trajectory on coordinates. The differences for case 1 and 6 are the most obvious since the perturbations on $T_{\text{max}}$ are the largest. Not like Figs. 5 and 8, the differences are not symmetry for the cases such as case 1 and 6.

The outcome of this simulation study indicates: 1) the thrust angles of the converged MPSP trajectories remain smooth; 2) the thrust sequence of the MPSP solution prefers to remain to be the nominal thrust sequence when coast segments can be increased. 3) the thrust sequence of MPSP solution can capture the main structure of the optimal thrust sequence when the coast sequence is reduced; 4) Even though the fuel consumption is not included inside the performance index, the MPSP trajectories are competitive in terms of fuel consumption even when the thrust sequence is changed.

V. Conclusion

Unlike the applications with continuous control profile, this paper shows that the sensitive matrix is discontinuous at the bang-bang switching point. A robust two-loop MPSP algorithm is further designed as the low-thrust guidance scheme. Numerical simulations illustrate that the proposed MPSP algorithm is robust for various kinds of perturbations. Besides, the fuel consumption is near optimal even when the thrust sequence is required to be changed. The future
Table 6  Perturbations on $T_{\text{max}}$.

| Case | $\eta$ (%) | $\|x_f - x(t_f)\|_2$ (km) | $\|v_f - v(t_f)\|_2$ (km/s) | $|\lambda_m(t_f)|$ | Newton’s iteration | Fuel Increase (%) |
|------|-------------|-----------------|-----------------|-----------------|-----------------|------------------|
| 1    | −10         | 1.77            | $3.57 \times 10^{-7}$ | $2.42 \times 10^{-9}$ | 93              | 1.27             |
| 2    | −6          | 373.68          | $5.21 \times 10^{-5}$ | $5.62 \times 10^{-7}$ | 24              | 0.18             |
| 3    | −3          | 267.52          | $5.18 \times 10^{-5}$ | $2.76 \times 10^{-7}$ | 4               | 0.082            |
| 4    | 3           | 6.50            | $1.10 \times 10^{-6}$ | $3.15 \times 10^{-9}$ | 4               | 0.062            |
| 5    | 6           | 12.25           | $2.10 \times 10^{-6}$ | $5.06 \times 10^{-9}$ | 5               | 0.48             |
| 6    | 10          | 55.35           | $9.47 \times 10^{-6}$ | $1.27 \times 10^{-8}$ | 5               | 1.19             |

Fig. 9  The variations of the in-plane angle $\alpha$ and out-of-plane angle $\beta$ for the converged MPSP trajectories for variations of $T_{\text{max}}$ in Tab. 6.
Fig. 10  The comparison of optimal thrust sequences and thrust sequences of MPSP solutions for cases in Tab. 6. The red dash line: optimal thrust sequences; Blue line: thrust sequences of MPSP solutions.

Fig. 11  Differences on coordinates between converged MPSP trajectories and the nominal trajectory for cases of perturbations on $T_{\text{max}}$ in Tab. 6.
work will refine the algorithm design to reduce the total iterations.

Acknowledgment

Yang Wang acknowledges the support of this work by the China Scholarship Council (Grant no.201706290024).

Appendix

A. Trajectory Discretization

Instead of discretization using Euler method, a higher order method is used to increase the accuracy. The classical 4th order Runge–Kutta formula is used as

\[
\begin{align*}
    x_{n+1} &= x_n + \frac{h}{6} (K_1 + 2K_2 + 2K_3 + K_4) \\
    K_1 &= F(t_{n,1}, x_{n,1}), \quad t_{n,1} = t_n, \quad x_{n,1} = x_n \\
    K_2 &= F(t_{n,2}, x_{n,2}), \quad t_{n,2} = t_n + \frac{h}{2}, \quad x_{n,2} = x_n + \frac{h}{2} K_1 \\
    K_3 &= F(t_{n,3}, x_{n,3}), \quad t_{n,3} = t_n + \frac{h}{2}, \quad x_{n,3} = x_n + \frac{h}{2} K_2 \\
    K_4 &= F(t_{n,4}, x_{n,4}), \quad t_{n,4} = t_n + h, \quad x_{n,4} = x_n + h K_3
\end{align*}
\]  

Combining Eq. (48) with Eq. (19) yields

\[
F_n(t_n, x_n) = x_n + \frac{h}{6} (K_1 + 2K_2 + 2K_3 + K_4) \tag{49}
\]

and its differential w.r.t \( x_n \) is

\[
\frac{dF_n(t_n, x_n)}{dx_n} = I_n + \frac{h}{6} \left( \frac{dK_1}{dx_n} + 2 \frac{dK_2}{dx_n} + 2 \frac{dK_3}{dx_n} + \frac{dK_4}{dx_n} \right) \tag{50}
\]

where

\[
\frac{dK_i}{dx_n} = \frac{dF(t_{n,i}, x_{n,i})}{dx_{n,i}} \frac{dx_{n,i}}{dx_n}, \quad i = 1, 2, 3, 4 \tag{51}
\]

References

[1] Tang, S., and Conway, B. A., “Optimization of low-thrust interplanetary trajectories using collocation and nonlinear programming,” *Journal of Guidance, Control, and Dynamics*, Vol. 18, No. 3, 1995, pp. 599–604.

[2] Betts, J. T., *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*, second edition ed., Society for Industrial and Applied Mathematics, 2010.
[3] Zhang, C., Topputo, F., Bernelli-Zazzera, F., and Zhao, Y.-S., “Low-thrust minimum-fuel optimization in the circular restricted three-body problem,” *Journal of Guidance, Control, and Dynamics*, Vol. 38, No. 8, 2015, pp. 1501–1510.

[4] Haberkorn, T., Martinon, P., and Gergaud, J., “Low Thrust Minimum-Fuel Orbital Transfer: A Homotopic Approach,” *Journal of Guidance, Control, and Dynamics*, Vol. 27, No. 6, 2004, pp. 1046–1060. doi:10.2514/1.4022, URL http://arc.aiaa.org/doi/10.2514/1.4022.

[5] Quadrelli, M. B., Wood, L. J., Riedel, J. E., McHenry, M. C., Aung, M., Cangahuala, L. A., Volpe, R. A., Beauchamp, P. M., and Cutts, J. A., “Guidance, navigation, and control technology assessment for future planetary science missions,” *Journal of Guidance, Control, and Dynamics*, Vol. 38, No. 7, 2015, pp. 1165–1186.

[6] Edelbaum, T. N., “Propulsion requirements for controllable satellites,” *Ars Journal*, Vol. 31, No. 8, 1961, pp. 1079–1089.

[7] Casalino, L., and Colasurdo, G., “Improved Edelbaum’s approach to optimize low earth/geostationary orbits low-thrust transfers,” *Journal of guidance, control, and dynamics*, Vol. 30, No. 5, 2007, pp. 1504–1511.

[8] Kluever, C. A., “Simple guidance scheme for low-thrust orbit transfers,” *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 6, 1998, pp. 1015–1017.

[9] Petropoulos, A. E., “Simple control laws for low-thrust orbit transfers,” 2003.

[10] Hernandez, S., and Akella, M. R., “Lyapunov-based guidance for orbit transfers and rendezvous in Levi-Civita coordinates,” *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 4, 2014, pp. 1170–1181.

[11] Hernandez, S., and Akella, M. R., “Energy preserving low-thrust guidance for orbit transfers in KS variables,” *Celestial Mechanics and Dynamical Astronomy*, Vol. 125, No. 1, 2016, pp. 107–132.

[12] Bryson, A. E., and Ho, Y.-C., *Applied Optimal Control*, Taylor & Francis, 1975.

[13] Pontani, M., Cecchetti, G., and Teofilatto, P., “Variable-Time-Domain Neighboring Optimal Guidance, Part 1: Algorithm Structure,” *Journal of Optimization Theory and Applications*, Vol. 166, No. 1, 2015, pp. 76–92. doi:10.1007/s10957-014-0676-6, identifier: 676.

[14] Chen, Z., “Neighboring optimal control for fixed-time multi-burn orbital transfers,” *Aerospace Science and Technology*, Vol. 61, 2017, pp. 57–65.

[15] Chen, Z., and Tang, S., “Neighboring optimal control for open-time multiburn orbital transfers,” *Aerospace Science and Technology*, Vol. 74, 2018, pp. 37–45.

[16] Di Lizia, P., Armellin, R., and Lavagna, M., “Application of high order expansions of two-point boundary value problems to astrodynamics,” *Celestial Mechanics and Dynamical Astronomy*, Vol. 102, No. 4, 2008, pp. 355–375.

[17] Di Lizia, P., Armellin, R., Morselli, A., and Bernelli-Zazzera, F., “High order optimal feedback control of space trajectories with bounded control,” *Acta Astronautica*, Vol. 94, No. 1, 2014, pp. 383–394.
[18] Gao, Y., “Low-thrust nonlinear guidance by tracking mean orbital elements,” Journal of Guidance, Control, and Dynamics, Vol. 31, No. 4, 2008, pp. 1103–1110.

[19] Huang, R. C., Hwang, I., and Corless, M. J., “Nonlinear Algorithm for Tracking Interplanetary Low-Thrust Trajectories,” Journal of Guidance, Control, and Dynamics, Vol. 35, No. 2, 2012, pp. 696–700.

[20] Wang, Z., and Grant, M. J., “Minimum-Fuel Low-Thrust Transfers for Spacecraft: A Convex Approach,” IEEE Transactions on Aerospace and Electronic Systems, 2018.

[21] Pesch, H. J., “Real-time computation of feedback controls for constrained optimal control problems. part 1: Neighbouring extremals,” Optimal Control Applications and Methods, Vol. 10, No. 2, 1989, pp. 129–145.

[22] Pesch, H. J., “Real-time computation of feedback controls for constrained optimal control problems. part 2: A correction method based on multiple shooting,” Optimal Control Applications and Methods, Vol. 10, No. 2, 1989, pp. 147–171.

[23] Padhi, R., “Model predictive static programming: A promising technique for optimal missile guidance,” Annals of the Indian National Academy of Engineering, Vol. 5, 2008, pp. 185–194.

[24] Maity, A., Oza, H. B., and Padhi, R., “Generalized model predictive static programming and angle-constrained guidance of air-to-ground missiles,” Journal of Guidance, Control, and Dynamics, Vol. 37, No. 6, 2014, pp. 1897–1913.

[25] Oza, H. B., and Padhi, R., “Impact-angle-constrained suboptimal model predictive static programming guidance of air-to-ground missiles,” Journal of Guidance, Control, and Dynamics, Vol. 35, No. 1, 2012, pp. 153–164.

[26] Halbe, O., Raja, R. G., and Padhi, R., “Robust reentry guidance of a reusable launch vehicle using model predictive static programming,” Journal of Guidance, Control, and Dynamics, Vol. 37, No. 1, 2013, pp. 134–148.

[27] Zhang, B., Tang, S., and Pan, B., “Multi-constrained suboptimal powered descent guidance for lunar pinpoint soft landing,” Aerospace Science and Technology, Vol. 48, 2016, pp. 203–213.

[28] Mondal, S., and Padhi, R., “Angle-Constrained Terminal Guidance Using Quasi-Spectral Model Predictive Static Programming,” Journal of Guidance, Control, and Dynamics, Vol. 41, No. 3, 2017, pp. 783–791.

[29] Pan, B., Ma, Y., and Yan, R., “Newton-Type Methods in Computational Guidance,” Journal of Guidance, Control, and Dynamics, 2018, pp. 1–7.

[30] Battin, R. H., An Introduction to the Mathematics and Methods of Astrodynamics, revised edition, American Institute of Aeronautics and Astronautics, 1999.

[31] Robbins, H., “Optimality of intermediate-thrust arcs of rocket trajectories,” AIAA Journal, Vol. 3, No. 6, 1965, pp. 1094–1098.

[32] Gao, Y., and Li, X., “Optimization of low-thrust many-revolution transfers and Lyapunov-based guidance,” Acta Astronautica, Vol. 66, No. 1-2, 2010, pp. 117–129.
[33] Furfaro, R., and Mortari, D., “Least-squares solution of a class of optimal space guidance problems via theory of connections,” , 2019.

[34] Russell, R. P., “Primer vector theory applied to global low-thrust trade studies,” *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 2, 2007, pp. 460–472.