Y. Wang, F. Topputo
*Robust Bang-Off-Bang Low-Thrust Guidance Using Model Predictive Static Programming*
Acta Astronautica, Vol. 176, 2020, p. 357-370
doi:10.1016/j.actaastro.2020.06.037

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

Yang Wang [a,1,*], Francesco Topputo[a,2]

[a]*Department of Aerospace Science and Technology, Politecnico di Milano,*
*Via La Masa 34, Milano, 20156, Italy.*

## Abstract

Model Predictive Static Programming (MPSP) has been always used under the assumption of continuous or impulsive control, but never in low-thrust transfers featuring bang-off-bang control. Following the observation that the sensitivity matrix (SM) is discontinuous across the switching time, this work introduces a two-loop MPSP guidance scheme using fuel-optimal trajectories as nominal solutions. In our method, the equations of motion are augmented by the mass costate equation, while the velocity costate is the MPSP control, expressed by a weighted sum of Fourier basis functions. The SM is computed at the switching time by using calculus of variations. Both the MPSP control and the initial mass costate are updated in an inner loop using Newton's method, and continuation is employed in an outer loop to face large perturbations. A sample interplanetary CubeSat mission to an asteroid is used as study case to illustrate the effectiveness and robustness of the method developed.

*Keywords:* Low-thrust transfers, Model predictive static programming, Bang-off-bang control.

[*] Corresponding author.

[1] PhD Candidate. E-mail address: yang.wang@polimi.it.

[2] Associate Professor. E-mail address: francesco.topputo@polimi.it.

Preliminary results of this work were presented as Paper IAC-19,C1,9,3,x49975 at the 70th International Astronautical Congress, Washington D.C., USA, 21–25 October 2019.

## 1. Introduction

Highly efficient propulsion systems, such as electric thrusters and solar sails, have made low-thrust propulsion an alternative to enable ambitious space missions. Extensive works have focused on open-loop optimal low-thrust trajectory design [1–4]. However, in real-world flight, disturbances such as solar radiation pressure and unmodeled accelerations, require to continuously update the control profile. The commonly used strategy is to uplink control commands from ground. This involves massive off-line computations and frequent communications with ground, with a consequent increase of operation costs. Due to rapid proliferation of miniaturized interplanetary space probes, this strategy hardly meets the always-increasing demand for operation cost reduction through autonomy [5].

Nonlinear optimal control problem (NOCP) theory is used to design guidance and control laws: it allows satisfying constraints while simultaneously optimizing a performance index. Neighboring optimal control (NOC) is the widely used NOCP-based technique to track a nominal solution. NOC formulates the feedback control by optimizing a second-order performance index [6]. Several works have investigated NOC methods for bang-off-bang control problems. Kornhauser and Lion [7] derived the corrective controller by establishing and solving an accessory minimum problem. Zheng [8, 9] investigated the NOC from a geometric point of view, for both fixed terminal time and free terminal time multi-burn transfer problems. Di Lizia et al [10] designed high-order NOC using differential algebra.

Nonlinear model predictive control (NMPC) is another tracking strategy, which employs iterative and finite-horizon optimization. For low-thrust transfers, Gao [11] designed NMPC using orbit averaging techniques to track the mean orbit elements. Huang et al [12] proposed a NMPC strategy that utilizes differential transformation in optimization to track the nominal trajectory. The tracking strategy enforces the spacecraft to be in close vicinity of the nominal solution for any flight conditions [13]. In bang-off-bang control problems, the controller design is usually based on the assumption that the bang-off-bang structure is unchanged under perturbations [8, 10, 14]. Therefore, this strategy lacks operational flexibility.

In order to overcome this drawback, algorithms that allow re-computing the entire nominal trajectory at the beginning of each guidance interval have been sought. One strategy involves solving a two-point boundary value problem (TPBVP). Pesch [15, 16] applied multiple shooting method to recompute the trajectory for general optimal control problems. Wang [17] used convex programming to compute fuel-optimal spacecraft trajectory, yet costate estimation is required to accurately capture the bang-off-bang structure [18]. Recently, Furfaro and Mortari [19] developed an energy-optimal guidance method by using the theory of function connections (TFC). Johnston et al [20] further extended the TFC-based guidance to fuel-optimal landing problems, based on a prescribed thrust sequence. The drawback of solving a TPBVP is that it employs iterative methods, and thus it requires consistent computational capabilities and exposes to convergence risks [13].

Alternatively, dynamic programming relies on Hamilton–Jacobi–Bellman (HJB) equation [6]. Beard [21] approximated the solution to HJB equation by using Galerkin's spectral method. Park and Scheeres [22] employed generating functions to design feedback control laws. Vadali and Sharma [23] formulated feedback control law by power series expansion of the cost-to-go function with time-dependent gains. Heydari and Balakrishnan [24] developed a neural-network-based control method, where the dynamical programming theory is used to train the network. Sharma and York [25] used a pseudo-linear dynamical system to design state-dependent feedback control for problems with linear terminal constraints. In summary, few works deal with guidance and control laws for low-thrust transfers with bang-off-bang policy through dynamical programming.

Model Predictive Static Programming (MPSP), a NOCP-based technique that combines model predictive control and dynamic programming [26], is a possibly promising technique. The innovation of MPSP is threefold [27]. Firstly, it successfully converts a dynamic programming problem to a static programming problem, and thus it requires only a static costate vector for the control update. Secondly, the symbolic costate vector has a closed-form solution, which reduces the computational load. Thirdly, the sensitivity matrix (SM) that is necessary to compute the static costate vector can be calculated recursively. These advantages favor applications of the MPSP in various contexts [13, 28–30]. Some variations of MPSP have also been proposed. For example, the generalized MPSP [27] formulated the problem in continuous-time framework, not requiring any discretization process to begin with. Quasi-Spectral MPSP [31] expressed 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. Recently, Sakode and Padhi [32] extended MPSP to impulse control, but with fixed impulse time instance and continuous impulse magnitude. Nevertheless, to the best of the authors knowledge, MPSP has not been investigated for low-thrust transfers with bang-off-bang control. This is because, as shown in the present paper, the SM across the switching time is discontinuous and compensation is thus required. Therefore, since state-of-the-art MPSP methods always assume continuous or impulsive control, they fail to get accurate SM.

In this work, a robust two-loop MPSP guidance scheme is presented for low-thrust transfers with bang-off-bang control. Firstly, the fuel-optimal low-thrust problem is stated in Cartesian coordinates, and the fuel-optimal solution is used as the nominal solution. Inspired by the natural feedback controller given by Pontryagin minimum principle, the unconstrained velocity costate vector is treated as MPSP control. In order to ensure the continuity of the switching function at the switching time, dynamical equations are augmented by the mass costate equation. Moreover, the SM is calculated recursively, and switches are compensated using calculus of variations. This validates our assumption that the SM is discontinuous across the switching time. In order to ensure the continuity of the thrust angle profile, the MPSP control is represented by a weighted sum of Fourier basis functions, where the weights are initialized based on the nominal trajectory using a least square method. Finally, a two-loop MPSP method is designed for both small and large perturbations, where Newton's method and continuation are implemented in inner and outer loops, respectively. For the first time, a MPSP guidance scheme handling bang-off-bang control in low-thrust transfers is designed. A number of numerical simulations demonstrate the effectiveness and robustness of the proposed method.

This paper is structured as follows: Section 2 states the guidance 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. Final remarks are given in Section 5.

## 2. Problem Statement

### 2.1. Equations of Motion

This work considers the heliocentric phase of an interplanetary transfer mission, where the spacecraft is subject to the gravitational attraction of the

Sun. The equations of motion of the spacecraft are

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\alpha}, u) \Rightarrow \begin{pmatrix} \dot{\boldsymbol{r}} \\ \dot{\boldsymbol{v}} \\ \dot{m} \end{pmatrix} = \begin{pmatrix} \boldsymbol{v} \\ \boldsymbol{g}(\boldsymbol{r}) + u\dfrac{T_{\max}}{m}\boldsymbol{\alpha} \\ -u\dfrac{T_{\max}}{c} \end{pmatrix} \tag{1}$$

where $\mu$ is the Sun gravitational parameter, $\boldsymbol{g}(\boldsymbol{r}) \coloneqq -\mu\boldsymbol{r}/r^3$, $\boldsymbol{r} \coloneqq [x, y, z]^\top$, and $\boldsymbol{v} \coloneqq [v_x, v_y, v_z]^\top$ are the gravitational vector field, the spacecraft position vector, and its velocity vector, respectively; $m$ is the spacecraft mass, $T_{\max}$ 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), $u$ is the thrust throttle factor, $\boldsymbol{\alpha}$ is the thrust pointing vector. The state vector is $\boldsymbol{x} \coloneqq [\boldsymbol{r}, \boldsymbol{v}, m]^\top$. Both $T_{\max}$ and $c$ are assumed constant during flight.

### *2.2. Fuel-Optimal Problem*

The nominal solution is a fuel-optimal trajectory. The performance index is

$$J = \frac{T_{\max}}{c}\int_{t_0}^{t_f} u \;\mathrm{d}t \tag{2}$$

where $t_0$ and $t_f$ are initial and terminal time, both fixed. The initial state is known, i.e., $\boldsymbol{x}(t_0) = \boldsymbol{x}_0$. For an interplanetary mission, the final position and velocity are known, i.e.,

$$\boldsymbol{r}(t_f) = \boldsymbol{r}_f, \qquad \boldsymbol{v}(t_f) = \boldsymbol{v}_f \tag{3}$$

The inequality constraint for the thrust throttle factor is

$$0 \le u \le 1 \tag{4}$$

The Hamiltonian function reads [6]

$$H = \frac{T_{\max}}{c}u + \boldsymbol{\lambda}_r^\top\boldsymbol{v} + \boldsymbol{\lambda}_v^\top\left[\boldsymbol{g}(\boldsymbol{r}) + u\frac{T_{\max}}{m}\boldsymbol{\alpha}\right] - \lambda_m u\frac{T_{\max}}{c} \tag{5}$$

where $\boldsymbol{\lambda} \coloneqq [\boldsymbol{\lambda}_r, \boldsymbol{\lambda}_v, \lambda_m]$ is the costate vector associated to $\boldsymbol{x}$. Dynamical equations for $\boldsymbol{\lambda}$ are

$$\dot{\boldsymbol{\lambda}} = -\left(\frac{\partial H}{\partial \boldsymbol{x}}\right)^\top \Rightarrow \begin{pmatrix} \dot{\boldsymbol{\lambda}}_r \\ \dot{\boldsymbol{\lambda}}_v \\ \dot{\lambda}_m \end{pmatrix} = \begin{pmatrix} -\boldsymbol{G}(\boldsymbol{r})^\top\boldsymbol{\lambda}_v \\ -\boldsymbol{\lambda}_r \\ uT_{\max}/m^2\boldsymbol{\lambda}_v^\top\boldsymbol{\alpha} \end{pmatrix} \tag{6}$$

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

$$\lambda_m(t_f) = 0 \tag{7}$$

According to Pontryagin minimum principle, the optimal thrust direction is [33]

$$\boldsymbol{\alpha}^* = -\frac{\boldsymbol{\lambda}_v}{\lambda_v}, \quad \text{if} \quad \lambda_v \neq 0 \tag{8}$$

where $\lambda_v = \|\boldsymbol{\lambda}_v\|_2$. Substituting Eq. (8) into Eq. (5) yields

$$H = \boldsymbol{\lambda}_r^\top \boldsymbol{v} + \boldsymbol{\lambda}_v^\top \boldsymbol{g}(\boldsymbol{r}) + \frac{u T_{\max}}{c} S \tag{9}$$

where the switching function $S$ is defined as

$$S = -\lambda_v \frac{c}{m} - \lambda_m + 1 \tag{10}$$

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

$$u^* = \begin{cases} 0, & \text{if} \quad S > 0 \\ 1, & \text{if} \quad S < 0 \end{cases} \tag{11}$$

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

In indirect methods the problem is to find $\boldsymbol{\lambda}_0$ that (together with $\boldsymbol{x}_0$) allows integrating Eqs. (1) and (6) with the control law in Eqs. (8) and (11) and verifies the terminal constraints (3) and (7) [2]. Singular thrust arcs are not considered here since they have been shown to be non-optimal in general [34]. Once optimal $\boldsymbol{\alpha}^*(t)$ and $u^*(t)$ are determined, the nominal spacecraft trajectory can be generated by integrating Eqs. (1) and (6).

### *2.3. MPSP Dynamics and Control*

In real-world flight, it would be desirable to enable the spacecraft to autonomously update the control profile to cancel unmodeled perturbations or to achieve new mission goals. Model Predictive Static Programming (MPSP) is a promising technique to do so. Differently from indirect methods whose aim is to find $\boldsymbol{\lambda}_0$, MPSP directly manipulates and updates the control history through an iterative process [26]. However, the original MPSP for unconstrained problems [26] cannot be used in the present context because of the constraint in Eq. (4) and the control structure in Eq. (11).

Let $\boldsymbol{x}^*(t)$ and $\boldsymbol{\lambda}^*(t)$ denote the nominal state and costate profiles, respectively, and let $\boldsymbol{x}(t)$ and $\boldsymbol{\lambda}(t)$ be their associated, off-nominal values. Let $\Delta\boldsymbol{\lambda}(t)$ be the costate deviation. Then, the two functions [8]

$$\begin{cases} u(\boldsymbol{x}, \boldsymbol{\lambda}^* - \Delta\boldsymbol{\lambda}) = \dfrac{1}{2} \ \{1 - \mathrm{Sgn}\,[\ \mathrm{S}(\boldsymbol{x}, \boldsymbol{\lambda}^* - \Delta\boldsymbol{\lambda})\ ]\} \\ \boldsymbol{\alpha}(\boldsymbol{x}, \boldsymbol{\lambda}^* - \Delta\boldsymbol{\lambda}) = -\dfrac{\boldsymbol{\lambda}_v^* - \Delta\boldsymbol{\lambda}_v}{\|\boldsymbol{\lambda}_v^* - \Delta\boldsymbol{\lambda}_v\|_2}, \quad \text{if } \|\boldsymbol{\lambda}_v^* - \Delta\boldsymbol{\lambda}_v\|_2 \neq 0 \end{cases} \tag{12}$$

define the feedback controller associated with $\boldsymbol{x}$ at $t$, where Sgn is defined as

$$\mathrm{Sgn(z)} = \begin{cases} 1, & \text{if} \quad z > 0 \\ -1, & \text{if} \quad z < 0 \end{cases} \tag{13}$$

From Eq. (12) it is clear that updating $u$ and $\boldsymbol{\alpha}$ can be achieved by updating the unconstrained costate vector, thus this work treats the costate vector as the MPSP control variable. This idea also has been utilized in NOC design [8] and Lyapunov guidance design [35]. Notice from Eqs. (10) and (12) that the costate components affecting $u$ and $\boldsymbol{\alpha}$ are $\boldsymbol{\lambda}_v$ and $\lambda_m$. However, only $\boldsymbol{\lambda}_v$ is used as the MPSP control, based on three facts. Firstly, it can be seen from $\dot{\lambda}_m$ and $\dot{\boldsymbol{\lambda}}_v$ in Eq. (6) that $\dot{\lambda}_m$ is dependent on $\boldsymbol{\lambda}_v$, but $\dot{\boldsymbol{\lambda}}_v$ is independent on $\lambda_m$. Secondly, if $\lambda_m$ is used as a control variable, $\dot{\lambda}_m$ cannot be expressed by Eq. (6). Indeed, the time derivative of the switching function $S$

$$\dot{S} = -\dot{\lambda}_v \frac{c}{m} - \lambda_v \frac{uT_{max}}{m^2} - \dot{\lambda}_m \tag{14}$$

would hardly be continuous because of the presence of $u$. On the other hand, if $\lambda_m$ is augmented to the state vector, $\dot{\lambda}_m$ is the same as Eq. (6). Then $\dot{S}$ simply becomes

$$\dot{S} = -\dot{\lambda}_v \frac{c}{m} = -\frac{c}{m\lambda_v} \boldsymbol{\lambda}_v^\top \dot{\boldsymbol{\lambda}}_v \tag{15}$$

which is implicitly dependent on $u$. Thirdly, $\dot{\lambda}_m$ is discontinuous due to the presence of $u$. Since continuous basis functions are used to approximate the MPSP control profile in this work, they may be inappropriate to efficiently capture the discontinuity [19].

Thus, the MPSP dynamical equations used are

$$\dot{\boldsymbol{X}} = \boldsymbol{\mathcal{F}}(t, \boldsymbol{X}, \boldsymbol{U}) \Rightarrow \begin{pmatrix} \dot{\boldsymbol{r}} \\ \dot{\boldsymbol{v}} \\ \dot{m} \\ \dot{\lambda}_m \end{pmatrix} = \begin{pmatrix} \boldsymbol{v} \\ \boldsymbol{g}(\boldsymbol{r}) - u\dfrac{T_{\max}}{m\lambda_v}\boldsymbol{\lambda}_v \\ -u\dfrac{T_{\max}}{c} \\ -u\dfrac{T_{\max}}{m^2}\lambda_v \end{pmatrix} \tag{16}$$

where $\boldsymbol{X} \coloneqq [\boldsymbol{x}, \lambda_m]$, $\boldsymbol{U} \coloneqq \boldsymbol{\lambda}_v$. The optimal thrust direction Eq. (8) is already embedded into Eq. (16), and $u$ is computed by Eq. (11). The thrust angles are

$$\begin{cases} \alpha = \arctan\left(\dfrac{\lambda_{v,2}}{\lambda_{v,1}}\right) \\ \beta = \arcsin\left(\dfrac{\lambda_{v,3}}{\lambda_v}\right) \end{cases} \tag{17}$$

where $\alpha \in [0, 2\pi]$ is the in-plane angle[3], $\beta \in [-\pi/2, \pi/2]$ is the out-of-plane angle, and $\lambda_{v,i}$ is the $i^{\text{th}}$ element of $\boldsymbol{\lambda}_v$. Once $\boldsymbol{X}(t)$ and $\boldsymbol{U}(t)$ are determined, the profile of $S$ is known, which then determines $u$ and $\boldsymbol{\alpha}$ through Eqs. (11) and (8). Suppose that the off-nominal trajectory is given, the task of the MPSP guidance is to determine $\Delta\boldsymbol{U}(t)$ and $\Delta\lambda_{m0}$ such that the future solution obtained by integrating Eq. (16) satisfies the boundary conditions Eqs. (3) and (7), while undergoing a bang-off-bang control.

## 3. MPSP Algorithm Design

### *3.1. Sensitive Matrix Calculation*

Differently from problems with continuous control profile, dynamical discontinuity happens at switching points. After the integration of Eq. (16), the obtained trajectory is split into multiple segments. The time instants bounding each segment are $\{t_0, t_{1,j}, \cdots, t_{M_j-1,j}, t_{M_j}\}$, where $M_j$ is the number of segments, $t_0$ and $t_{M_j} = t_f$ are initial and final time, respectively, and $t_{k,j}, k = 1, 2, \cdots, M_j - 1$ are switching times; the subscript $j$ denotes the $j^{\text{th}}$ iteration,

[3] $\alpha$ is mathematically defined in $[-\pi/2, \pi/2]$ because of "arctan" function. Here, the quadrant is considered to better resolve $\alpha$.

our method being iterative. Let $\{t_{k,j}^{0-}, t_{k,j}^{0+}, t_{k,j}^{1}, ...t_{k,j}^{N_{k,j}}\}, k = 0, 1, ..., M_j - 1$ denote an evenly-spaced time grid within $[t_{k,j}, t_{k+1,j}]$, where $t_{k,j}^{0-}$ and $t_{k,j}^{0+}$ are the time instants across the instantaneous switch-on or off of the engine, and

$$N_{k,j} = \text{Ceil}\left(\frac{t_{k+1,j} - t_{k,j}}{h_{\max}}\right) \tag{18}$$

where "Ceil" is the round up operator and $h_{\max}$ is the prescribed maximum discretization interval. To ease notation, $N_{k,j}$ and $M_j$ are denoted $N$ and $M$. Note that $t_{k,j}^{N} = t_{k+1,j}^{0-}$, thus they label the same time instant, but belong to adjacent segments. Suppose there is no switching at the initial time; then, $t_0^{0-} = t_0^{0+} = t_0$.

Consider the $k^{\text{th}}$ segment $[t_{k,j}, t_{k+1,j}]$, the discretized system dynamics and its output are

$$\boldsymbol{X}_{k,j}^{i+1} = \boldsymbol{F}_{k,j}^{i}(\boldsymbol{X}_{k,j}^{i}, \boldsymbol{U}_{k,j}^{i}), \quad \boldsymbol{Y}_{k,j}^{i} = \boldsymbol{O}(\boldsymbol{X}_{k,j}^{i}) \tag{19}$$

where $\boldsymbol{Y}_{k,j}^{i}$, is the output at $t_{k,j}^{i}$ and $\boldsymbol{O}$ is the output operator. $\boldsymbol{F}_{k,j}^{i}(\boldsymbol{X}_{k,j}^{i}, \boldsymbol{U}_{k,j}^{i})$ is the discretized dynamics, which can be obtained from Eq. (16) using standard integration formulae [26]. High-order integration formulae result in higher accuracy, but larger computational load. In this work, the standard 4th order Runge–Kutta integration scheme is used, see Appendix A.

Given the initial state $\boldsymbol{x}_0$, the primary objective is to update the control history $\boldsymbol{U}_{k,j}^{i}$ and the initial mass costate $\lambda_{m0,j}$ such that the output $\boldsymbol{Y}_{k,j}^{i}$ at $t_f$, i.e., $\boldsymbol{Y}_{M-1,j}^{N}$, reaches the desired value $\boldsymbol{Y}_d$. Writing $\boldsymbol{Y}_{M-1,j}^{N}$ about $\boldsymbol{Y}_d$ in Taylor series expansion and neglecting high-order terms, the error at the terminal output $\Delta\boldsymbol{Y}_{M-1,j}^{N} = \boldsymbol{Y}_{M-1,j}^{N} - \boldsymbol{Y}_d$ is approximated as

$$\Delta\boldsymbol{Y}_{M-1,j}^{N} \cong \mathrm{d}\boldsymbol{Y}_{M-1,j}^{N} = \left[\frac{\partial \boldsymbol{Y}_{M-1,j}^{N}}{\partial \boldsymbol{X}_{M-1,j}^{N}}\right] \mathrm{d}\boldsymbol{X}_{M-1,j}^{N} \tag{20}$$

The derivation of $\mathrm{d}\boldsymbol{X}_{k,j}^{i+1}$ should consider whether a discontinuity appears or not. For the intervals $[t_{k,j}^{i}, t_{k,j}^{i+1}]$, $i = 1, \cdots, N-1$, where there is no switching time, there exists

$$\mathrm{d}\boldsymbol{X}_{k,j}^{i+1} = \left[\frac{\partial \boldsymbol{F}_{k,j}^{i}}{\partial \boldsymbol{X}_{k,j}^{i}}\right] \mathrm{d}\boldsymbol{X}_{k,j}^{i} + \left[\frac{\partial \boldsymbol{F}_{k,j}^{i}}{\partial \boldsymbol{U}_{k,j}^{i}}\right] \mathrm{d}\boldsymbol{U}_{k,j}^{i} \tag{21}$$

See Appendix A the computation of $\partial\boldsymbol{F}_{k,j}^{i}/\partial\boldsymbol{X}_{k,j}^{i}$ and $\partial\boldsymbol{F}_{k,j}^{i}/\partial\boldsymbol{U}_{k,j}^{i}$.

For the interval $[t_{k,j}^{0-}, t_{k,j}^{1}]$ that contains a switching point, we have

$$\begin{aligned}
\mathrm{d}\boldsymbol{X}_{k,j}^{1} &= \left[\frac{\partial \boldsymbol{F}_{k,j}^{0+}}{\partial \boldsymbol{X}_{k,j}^{0+}}\right] \mathrm{d}\boldsymbol{X}_{k,j}^{0+} + \left[\frac{\partial \boldsymbol{F}_{k,j}^{0+}}{\partial \boldsymbol{U}_{k,j}^{0+}}\right] \mathrm{d}\boldsymbol{U}_{k,j}^{0+} \\
&= \left[\frac{\partial \boldsymbol{F}_{k,j}^{0+}}{\partial \boldsymbol{X}_{k,j}^{0+}}\right] \left[\frac{\partial \boldsymbol{X}_{k,j}^{0+}}{\partial \boldsymbol{X}_{k,j}^{0-}}\right] \mathrm{d}\boldsymbol{X}_{k,j}^{0-} + \left[\frac{\partial \boldsymbol{F}_{k,j}^{0+}}{\partial \boldsymbol{X}_{k,j}^{0+}}\right] \left[\frac{\partial \boldsymbol{X}_{k,j}^{0+}}{\partial \boldsymbol{U}_{k,j}^{0-}}\right] \mathrm{d}\boldsymbol{U}_{k,j}^{0-} + \left[\frac{\partial \boldsymbol{F}_{k,j}^{0+}}{\partial \boldsymbol{U}_{k,j}^{0+}}\right] \mathrm{d}\boldsymbol{U}_{k,j}^{0+} \\
&= \left[\frac{\partial \boldsymbol{F}_{k,j}^{0}}{\partial \boldsymbol{X}_{k,j}^{0}}\right] \mathrm{d}\boldsymbol{X}_{k,j}^{0} + \left[\frac{\partial \boldsymbol{F}_{k,j}^{0}}{\partial \boldsymbol{U}_{k,j}^{0}}\right] \mathrm{d}\boldsymbol{U}_{k,j}^{0}
\end{aligned} \tag{22}$$

where $\mathrm{d}\boldsymbol{U}_{k,j}^{0} := \mathrm{d}\boldsymbol{U}_{k,j}^{0-} = \mathrm{d}\boldsymbol{U}_{k,j}^{0+}$ due to continuity of the thrust angle, $\mathrm{d}\boldsymbol{X}_{k,j}^{0} := \mathrm{d}\boldsymbol{X}_{k,j}^{0-}$, and

$$\left[\frac{\partial \boldsymbol{F}_{k,j}^{0}}{\partial \boldsymbol{X}_{k,j}^{0}}\right] := \left[\frac{\partial \boldsymbol{F}_{k,j}^{0+}}{\partial \boldsymbol{X}_{k,j}^{0+}}\right] \left[\frac{\partial \boldsymbol{X}_{k,j}^{0+}}{\partial \boldsymbol{X}_{k,j}^{0-}}\right], \quad \left[\frac{\partial \boldsymbol{F}_{k,j}^{0}}{\partial \boldsymbol{U}_{k,j}^{0}}\right] := \left[\frac{\partial \boldsymbol{F}_{k,j}^{0+}}{\partial \boldsymbol{X}_{k,j}^{0+}}\right] \left[\frac{\partial \boldsymbol{X}_{k,j}^{0+}}{\partial \boldsymbol{U}_{k,j}^{0-}}\right] + \left[\frac{\partial \boldsymbol{F}_{k,j}^{0+}}{\partial \boldsymbol{U}_{k,j}^{0+}}\right] \tag{23}$$

Based on calculus of variations, the expressions of $\partial \boldsymbol{X}_{k,j}^{0+}/\partial \boldsymbol{X}_{k,j}^{0-}$ and $\partial \boldsymbol{X}_{k,j}^{0+}/\partial \boldsymbol{U}_{k,j}^{0-}$ are (see Appendix B for the derivation of Eqs. (24) and (25))

$$\left[\frac{\partial \boldsymbol{X}_{k,j}^{0+}}{\partial \boldsymbol{X}_{k,j}^{0-}}\right] = \boldsymbol{I} + \left(\dot{\boldsymbol{X}}_{k,j}^{0+} - \dot{\boldsymbol{X}}_{k,j}^{0-}\right) S_{\boldsymbol{X}} / \dot{S} \tag{24}$$

$$\left[\frac{\partial \boldsymbol{X}_{k,j}^{0+}}{\partial \boldsymbol{U}_{k,j}^{0-}}\right] = \left(\dot{\boldsymbol{X}}_{k,j}^{0+} - \dot{\boldsymbol{X}}_{k,j}^{0-}\right) S_{\boldsymbol{U}} / \dot{S} \tag{25}$$

where $S_{\boldsymbol{X}}$ and $S_{\boldsymbol{U}}$ are row vectors that are the partial derivative of $S$ w.r.t. $\boldsymbol{X}$ and $\boldsymbol{U}$, respectively, i.e., $S_{\boldsymbol{X}} = \left[\boldsymbol{0}_{1\times 6}, \frac{c}{m^2}\lambda_v, -1\right]$ and $S_{\boldsymbol{U}} = -\frac{c}{m\lambda_v}\boldsymbol{\lambda}_v^{\top}$, and $\dot{S}$ is calculated according to Eq. (15).

Combining Eq. (21) with Eq. (22) yields the unified form of $\mathrm{d}\boldsymbol{X}_{k,j}^{i+1}$ as

$$\mathrm{d}\boldsymbol{X}_{k,j}^{i+1} = \left[\frac{\partial \boldsymbol{F}_{k,j}^{i}}{\partial \boldsymbol{X}_{k,j}^{i}}\right] \mathrm{d}\boldsymbol{X}_{k,j}^{i} + \left[\frac{\partial \boldsymbol{F}_{k,j}^{i}}{\partial \boldsymbol{U}_{k,j}^{i}}\right] \mathrm{d}\boldsymbol{U}_{k,j}^{i}, \quad i = 0, 1, \cdots, N-1 \tag{26}$$

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

$$\mathrm{d}\boldsymbol{Y}_{M-1,j}^{N} = \left[\frac{\partial \boldsymbol{Y}_{M-1,j}^{N}}{\partial \boldsymbol{X}_{M-1,j}^{N}}\right] \left\{ \left[\frac{\partial \boldsymbol{F}_{M-1,j}^{N-1}}{\partial \boldsymbol{X}_{M-1,j}^{N-1}}\right] \mathrm{d}\boldsymbol{X}_{M-1,j}^{N-1} + \left[\frac{\partial \boldsymbol{F}_{M-1,j}^{N-1}}{\partial \boldsymbol{U}_{M-1,j}^{N-1}}\right] \mathrm{d}\boldsymbol{U}_{M-1,j}^{N-1} \right\} \tag{27}$$

Similarly, $\mathrm{d}\boldsymbol{X}_{M-1,j}^{N-1}$ at $t_{M-1,j}^{N-1}$ can be expanded in terms of $\mathrm{d}\boldsymbol{X}_{M-1,j}^{N-2}$ and $\mathrm{d}\boldsymbol{U}_{M-1,j}^{N-2}$ at $t_{M-1,j}^{N-2}$. Next, $\mathrm{d}\boldsymbol{X}_{M-1,j}^{N-2}$ can be expanded in terms of $\mathrm{d}\boldsymbol{X}_{M-1,j}^{N-3}$ and $\mathrm{d}\boldsymbol{U}_{M-1,j}^{N-3}$. For the $(M-1)^{\text{th}}$ segment, this process goes back to $\mathrm{d}\boldsymbol{X}_{M-1,j}^{0}$. Notice that $\mathrm{d}\boldsymbol{X}_{M-2,j}^{N} = \mathrm{d}\boldsymbol{X}_{M-1,j}^{0}$, and thus the same process can be extended back for the $(M-2)^{\text{th}}$ segment. Extending the process until $\boldsymbol{X}_{0,j}^{0}$, one obtains

$$\begin{aligned}\mathrm{d}\boldsymbol{Y}_{M-1,j}^{N} &= \boldsymbol{A}_j \ \mathrm{d}\boldsymbol{X}_{0,j}^{0} + \boldsymbol{B}_{0,j}^{0} \ \mathrm{d}\boldsymbol{U}_{0,j}^{0} + \boldsymbol{B}_{0,j}^{1} \ \mathrm{d}\boldsymbol{U}_{0,j}^{1} + \cdots + \boldsymbol{B}_{M-1,j}^{N-1} \ \mathrm{d}\boldsymbol{U}_{M-1,j}^{N-1} \\ &= \boldsymbol{A}_j \ \mathrm{d}\boldsymbol{X}_{0,j}^{0} + \sum_{k=0}^{M-1} \sum_{i=0}^{N-1} \boldsymbol{B}_{k,j}^{i} \ \mathrm{d}\boldsymbol{U}_{k,j}^{i}\end{aligned} \tag{28}$$

The compact form of coefficients $\boldsymbol{A}_j$ and $\boldsymbol{B}_{k,j}^{i}$ in Eq. (28) are

$$\begin{gathered}\boldsymbol{A}_j = \left[\frac{\partial \boldsymbol{Y}_{M-1,j}^{N}}{\partial \boldsymbol{X}_{M-1,j}^{N}}\right] \prod_{k=M-1}^{0} \prod_{i=N-1}^{0} \left[\frac{\partial \boldsymbol{F}_{k,j}^{i}}{\partial \boldsymbol{X}_{k,j}^{i}}\right] \\ \boldsymbol{B}_{k,j}^{i} = \left[\frac{\partial \boldsymbol{Y}_{M-1,j}^{N}}{\partial \boldsymbol{X}_{M-1,j}^{N}}\right] \left\{\prod_{p=M-1}^{k+1} \prod_{q=N-1}^{0} \left[\frac{\partial \boldsymbol{F}_{p,j}^{q}}{\partial \boldsymbol{X}_{p,j}^{q}}\right]\right\} \left\{\prod_{q=N-1}^{i+1} \left[\frac{\partial \boldsymbol{F}_{k,j}^{q}}{\partial \boldsymbol{X}_{k,j}^{q}}\right]\right\} \left[\frac{\partial \boldsymbol{F}_{k,j}^{i}}{\partial \boldsymbol{U}_{k,j}^{i}}\right]\end{gathered} \tag{29}$$

where

$$\prod_{q=i}^{\Omega} \left[\frac{\partial \boldsymbol{F}_{k,j}^{q}}{\partial \boldsymbol{X}_{k,j}^{q}}\right] = \begin{cases} 1, & \text{if} \quad \Omega > i \\ \left[\frac{\partial \boldsymbol{F}_{k,j}^{i}}{\partial \boldsymbol{X}_{k,j}^{i}}\right] \left[\frac{\partial \boldsymbol{F}_{k,j}^{i-1}}{\partial \boldsymbol{X}_{k,j}^{i-1}}\right] \cdots \left[\frac{\partial \boldsymbol{F}_{k,j}^{\Omega}}{\partial \boldsymbol{X}_{k,j}^{\Omega}}\right], & \text{if} \quad \Omega \leq i \end{cases} \tag{30}$$

$$\prod_{k=i}^{\Omega} \left[\frac{\partial \boldsymbol{F}_{k,j}^{q}}{\partial \boldsymbol{X}_{k,j}^{q}}\right] = \begin{cases} 1, & \text{if} \quad \Omega > i \\ \left[\frac{\partial \boldsymbol{F}_{i,j}^{q}}{\partial \boldsymbol{X}_{i,j}^{q}}\right] \left[\frac{\partial \boldsymbol{F}_{i-1,j}^{q}}{\partial \boldsymbol{X}_{i-1,j}^{q}}\right] \cdots \left[\frac{\partial \boldsymbol{F}_{\Omega,j}^{q}}{\partial \boldsymbol{X}_{\Omega,j}^{q}}\right], & \text{if} \quad \Omega \leq i \end{cases} \tag{31}$$

Note that the initial state $\boldsymbol{x}_0$ is known and fixed ($\mathrm{d}\boldsymbol{x}_0 = \boldsymbol{0}$), while the initial value of $\lambda_{m0}$ can be adjusted. Then, Eq. (28) is modified as

$$\mathrm{d}\boldsymbol{Y}_{M-1,j}^{N} = \boldsymbol{A}_{\lambda,j} \ \mathrm{d}\lambda_{m0,j} + \sum_{k=0}^{M-1} \sum_{i=0}^{N-1} \boldsymbol{B}_{k,j}^{i} \ \mathrm{d}\boldsymbol{U}_{k,j}^{i} \tag{32}$$

where

$$\boldsymbol{A}_{\lambda,j} = \boldsymbol{A}_j \left[\frac{\partial \boldsymbol{X}_0^0}{\partial \lambda_{m0}}\right] = \boldsymbol{A}_j \ [\boldsymbol{0}_{1\times 7}, 1]^{\top} \tag{33}$$

The outcome of Eq. (33) is to assign the last column vector of $\boldsymbol{A}_j$ to $\boldsymbol{A}_{\lambda,j}$. The coefficient $\boldsymbol{B}_{k,j}^i$ is called sensitivity matrix (SM) [13]. The MPSP technique is desirable because the computation of SM can be reduced to an iterative calculation. Define

$$\boldsymbol{B}_{M-1,j,0}^N = \left[\frac{\partial \boldsymbol{Y}_{M-1,j}^N}{\partial \boldsymbol{X}_{M-1,j}^N}\right] \tag{34}$$

there exists

$$\boldsymbol{B}_{k,j,0}^i = \boldsymbol{B}_{k,j,0}^{i+1}\left[\frac{\partial \boldsymbol{F}_{k,j}^{i+1}}{\partial \boldsymbol{X}_{k,j}^{i+1}}\right], \qquad \boldsymbol{B}_{k,j}^i = \boldsymbol{B}_{k,j,0}^i\left[\frac{\partial \boldsymbol{F}_{k,j}^i}{\partial \boldsymbol{U}_{k,j}^i}\right] \tag{35}$$

In traditional MPSP technique, $\mathrm{d}\boldsymbol{U}_{k,j}^i$ is linear with $\boldsymbol{B}_{k,j}^i$ [13]. However, it is clearly that the SM is discontinuous across the switching time due to Eq. (22). Applying the classic MPSP will result in a discontinuous thrust angle, which is meaningless from a physical point of view.

### *3.2. MPSP Control Representation and Update*

In this work, the MPSP control profile is expressed by using a weighted sum of continuous basis functions. The advantages are twofold. Firstly, the continuity of the thrust angle profile can be ensured automatically due to the continuity of basis functions. Secondly, $\dot{S}$ as in Eq. (15) can be calculated analytically. The MPSP control is then expressed as

$$\boldsymbol{U}_{k,j}^i(\eta) = \boldsymbol{P}_{k,j}^i(\eta)\boldsymbol{\epsilon}_j \tag{36}$$

where $\boldsymbol{\epsilon}_j$ is the weight to the basis functions. In Eq. (36),

$$\boldsymbol{P}(\eta) = \begin{bmatrix} \boldsymbol{h}^\top(\eta) & & & \\ & \boldsymbol{h}^\top(\eta) & & \\ & & \ddots & \\ & & & \boldsymbol{h}^\top(\eta) \end{bmatrix} \tag{37}$$

where $\boldsymbol{h}(\eta)$ is the collection of different orders of basis functions, and $\eta \in [\eta_0, \eta_f]$ is the independent variable for them.

According to Eq. (36), the update of $\boldsymbol{U}_{k,j}^i(\eta)$ can be achieved by updating $\boldsymbol{\epsilon}_j$. The differential of Eq. (36) w.r.t. $\boldsymbol{\epsilon}_j$ is

$$\mathrm{d}\boldsymbol{U}_{k,j}^i(t) = \boldsymbol{P}_{k,j}^i(t)\mathrm{d}\boldsymbol{\epsilon}_j \tag{38}$$

where $\eta$ is replaced by $t$ using the linear projection as

$$\eta = \frac{\eta_f - \eta_0}{t_f - t_0}\,(t - t_0) + \eta_0 \tag{39}$$

Substituting Eq. (38) into Eq. (32) yields

$$\mathrm{d}\boldsymbol{Y}^N_{M-1,j} = \boldsymbol{A}_{\lambda,j}\ \mathrm{d}\lambda_{m0,j} + \sum_{k=0}^{M-1}\sum_{i=0}^{N-1}\ \boldsymbol{B}^i_{k,,j}\boldsymbol{P}^i_{k,j}\mathrm{d}\boldsymbol{\epsilon}_j \tag{40}$$

The performance index is set as

$$J = \frac{1}{2}\mathrm{d}\boldsymbol{\epsilon}_j^\top \boldsymbol{R}_\epsilon \mathrm{d}\boldsymbol{\epsilon}_j + \frac{1}{2}R_\lambda\ \mathrm{d}\lambda^2_{m0,j} \tag{41}$$

Since the MPSP control used is unrelated to the thrust throttle factor $u$, Eq. (41) is set in order to update the solution in the neighborhood of the nominal path.

Let $\boldsymbol{B}_{v,j} = \sum_{k=0}^{M-1}\sum_{i=0}^{N-1}\boldsymbol{B}^i_{k,j}\boldsymbol{P}^i_{k,j}$, the augmented performance index is

$$\hat{J} = \frac{1}{2}\mathrm{d}\boldsymbol{\epsilon}_j^\top \boldsymbol{R}_\epsilon \mathrm{d}\boldsymbol{\epsilon}_j + \frac{1}{2}R_\lambda\ \mathrm{d}\lambda^2_{m0,j} + \boldsymbol{p}_j^\top\left(\mathrm{d}\boldsymbol{Y}^N_{M-1,j} - \boldsymbol{A}_{\lambda,j}\ \mathrm{d}\lambda_{m0,j} - \boldsymbol{B}_{v,j}\mathrm{d}\boldsymbol{\epsilon}_j\right) \tag{42}$$

where $\boldsymbol{p}_j$ is the static costate vector associated to Eq. (40). The necessary conditions of optimality read

$$\begin{aligned}\left(\frac{\mathrm{d}\hat{J}}{\mathrm{d}\left(\mathrm{d}\boldsymbol{\epsilon}_j\right)}\right)^\top &= \boldsymbol{R}_\epsilon\ \mathrm{d}\boldsymbol{\epsilon}_j - \boldsymbol{B}_{v,j}^\top\boldsymbol{p}_j = \boldsymbol{0}\\ \frac{\mathrm{d}\hat{J}}{\mathrm{d}(\mathrm{d}\lambda_{m0,j})} &= R_\lambda\ \mathrm{d}\lambda_{m0,j} - \boldsymbol{A}_{\lambda,j}^\top\boldsymbol{p}_j = \boldsymbol{0}\end{aligned} \tag{43}$$

Substituting Eq. (43) into Eq. (40) yields

$$\boldsymbol{p}_j = \left(\boldsymbol{A}_{\lambda,j}\boldsymbol{A}_{\lambda,j}^\top/R_\lambda + \boldsymbol{B}_{v,j}\boldsymbol{R}_\epsilon^{-1}\boldsymbol{B}_{v,j}^\top\right)^{-1}\mathrm{d}\boldsymbol{Y}^N_{M-1,j} \tag{44}$$

with the assumption that $\boldsymbol{A}_{\lambda,j}\boldsymbol{A}_{\lambda,j}^\top/R_\lambda + \boldsymbol{B}_{v,j}\boldsymbol{R}_\epsilon^{-1}\boldsymbol{B}_{v,j}^\top$ is regular. Substituting Eq. (44) into Eq. (43) gives the Newton gradient $\mathrm{d}\boldsymbol{\epsilon}_j$ and $\mathrm{d}\lambda_{m0,j}$ at $j^{\mathrm{th}}$ iteration, as

$$\begin{aligned}\mathrm{d}\boldsymbol{\epsilon}_j &= \boldsymbol{R}_\epsilon^{-1}\boldsymbol{B}_{v,j}^\top\boldsymbol{p}_j\\ \mathrm{d}\lambda_{m0,j} &= \boldsymbol{A}_{\lambda,j}^\top\boldsymbol{p}_j/R_\lambda\end{aligned} \tag{45}$$

An iterative update is implemented: $\boldsymbol{\epsilon}_j$ and $\lambda_{m0,j}$ are updated using Newton's method as

$$\begin{aligned}\boldsymbol{\epsilon}_{j+1} &= \boldsymbol{\epsilon}_j - \mathrm{d}\boldsymbol{\epsilon}_j \\ \lambda_{m0,j+1} &= \lambda_{m0,j} - \mathrm{d}\lambda_{m0,j}\end{aligned} \tag{46}$$

where $\boldsymbol{\epsilon}_0$ and $\lambda_{m0,0}$ are the nominal solution. Then, $\Delta \boldsymbol{U}_k^i$ and $\Delta\lambda_{m0}$ satisfy

$$\begin{aligned}\Delta \boldsymbol{U}_k^i &= \boldsymbol{P}_{k,0}^i \mathrm{d}\boldsymbol{\epsilon}_0 + \boldsymbol{P}_{k,1}^i \mathrm{d}\boldsymbol{\epsilon}_1 + \cdots \\ \Delta\lambda_{m0} &= \mathrm{d}\lambda_{m0,0} + \mathrm{d}\lambda_{m0,1} + \cdots\end{aligned} \tag{47}$$

Through MPSP technique, the feedback control of $u$ and $\boldsymbol{\alpha}$ in Eq. (12) is achieved by an iterative process in Eqs. (46) and (47). This process converges when the terminal error satisfies the prescribed tolerance. Note that at $j^{\text{th}}$ iteration, $u$ and $\boldsymbol{\alpha}$ profiles are updated using $\epsilon_j$, $\lambda_{m0,j}$, $\boldsymbol{\alpha}$ in Eq. (8) and $u$ in Eq. (11) through integration, which is then used to calculate the Newton gradient $\mathrm{d}\boldsymbol{\epsilon}_j$ and $\mathrm{d}\lambda_{m0,j}$, and further $\epsilon_{j+1}$ and $\lambda_{m0,j+1}$. In order to maintain the algorithm stability, Newton's method is executed if the thrust sequence is varied within a given tolerance. This problem is addressed in Section 3.4.

### *3.3. Nominal Solution Generation*

The open-loop, fuel-optimal problem requires to find a zero of the shooting function associated with the TPBVP [2]. This work adopts a robust method that combines analytic derivatives, switching detection technique and homotopy method to find the fuel-optimal low-thrust trajectory; see [2]. Specifically, a smoothing technique is implemented to gradually enforce bang-off-bang discontinuity. The performance index is set as

$$\tilde{J} = \frac{T_{\max}}{c} \int_{t_0}^{t_f} \left[u - \zeta u(1-u)\right] \, \mathrm{d}t \tag{48}$$

where $\zeta$ is the continuation parameter. The strategy is to solve an energy-optimal problem ($\zeta = 1$) and to continue the solution manifold while gradually reducing $\zeta$, until the fuel-optimal problem ($\zeta = 0$) is solved. To solve the energy-optimal problem, the algorithm generates the initial guess solution using the Adjoint Control Transformation [36], and the shooting method is used to find the solution. The main feature of this method is the accurate calculation of state transition matrix which enhances its robustness.

As stated in Section 3.1, the evenly-spaced time grid is used for each segment. Let $\hat{N}$ be the total number of discrete points, and denote the nominal discrete control sequence as $\boldsymbol{U}_{k,\mathrm{nom}},\ k = 1, \cdots, \hat{N}$. The nominal $\lambda_{m0,0}$ can be directly obtained from the open-loop solution. The nominal weight $\boldsymbol{\epsilon}_0$ is computed by solving the following linear algebraic equation

$$\hat{\boldsymbol{P}}\boldsymbol{\epsilon}_0 = \hat{\boldsymbol{U}} \tag{49}$$

where

$$\hat{\boldsymbol{P}} = \begin{bmatrix} \boldsymbol{P}_1 \\ \boldsymbol{P}_2 \\ \vdots \\ \boldsymbol{P}_{\hat{N}} \end{bmatrix} \quad \hat{\boldsymbol{U}} = \begin{bmatrix} \boldsymbol{U}_{1,\mathrm{nom}} \\ \boldsymbol{U}_{2,\mathrm{nom}} \\ \vdots \\ \boldsymbol{U}_{\hat{N},\mathrm{nom}} \end{bmatrix}$$

The least-square solution is

$$\boldsymbol{\epsilon}_0 = (\hat{\boldsymbol{P}}^\top \hat{\boldsymbol{P}})^{-1}\hat{\boldsymbol{P}}^\top \hat{\boldsymbol{U}} \tag{50}$$

### *3.4. Two-Loop MPSP Guidance Scheme*

The integration of Eq. (16) is conducted using a 4th order Runge–Kutta fixed step scheme, with time step $h_{\max}$. Also, the switching detection technique [2] is embedded into integration process. The switching time is required to be detected accurately, based on two facts. Firstly, if the switching time is not detected, the integration error will accumulate around the switching time. Secondly, since SM is discontinuous across the switching time, the detection of the switching time is necessary for accurate calculation of SM. The detection occurs when the switching function $S$ traverses zero within $[t^i_{k,j}, t^{i+1}_{k,j}]$. The bisection method is used to find the switching time $t_s \in [t^i_{k,j}, t^{i+1}_{k,j}]$ at $j^{\mathrm{th}}$ iteration, such that $|S(t_s)| \leq 10^{-12}$.

The variation of thrust sequence is not restricted while updating $\epsilon$ and $\lambda_{m0}$. However, for large perturbations, the change of thrust sequence during the iteration amplifies the terminal error and then deteriorates the convergence. In order to stabilize the algorithm, a two-loop MPSP algorithm is designed made of an inner loop and an outer loop. The inner loop MPSP is illustrated in Algorithm 1, which consists of updating $\boldsymbol{\epsilon}$ and $\lambda_{m0}$ for the given perturbed conditions using Newton's method. It is worth stressing what follows. Firstly, Newton's method is implemented only when the $\mathrm{L}_2$ norm of terminal error at $j^{\mathrm{th}}$ iteration is less than a threshold $\Delta_{\max}$, i.e., $\|\Delta\boldsymbol{Y}^N_{M-1,j}\|_2 \leq \Delta_{\max}$. *Sign* is used to label the success (*Sign* $= 1$) or failure (*Sign* $= 0$) of the inner loop MPSP. Secondly, the interpolation is required since evenly-spaced state and control for each segment is used. The shape-preserving piecewise cubic interpolation which corresponds to "pchip" in Matlab's "interp1" is used. Thirdly, denote $N_{\text{seg},j}$ and $N_{\text{seg,nom}}$ as the sum of thrust segments and coast segments for the updated trajectory and the nominal trajectory at $j^{\text{th}}$ iteration. The solution is updated only when $|N_{\text{seg},j} - N_{\text{seg,nom}}| \leq N_{\text{seg,tol}}$, where $N_{\text{seg,tol}}$ is the restriction on variations of thrust sequence. Otherwise, $Sign = 0$ is returned.

The outer loop MPSP is shown in Algorithm 2, which is triggered when $Sign = 0$ of the inner loop MPSP is returned. In the outer loop MPSP, the continuation from nominal conditions to perturbed conditions is conducted. Denote $\boldsymbol{C}_{\text{nom}}$ as nominal conditions, $\boldsymbol{C}_{\text{per}}$ as perturbed conditions and $\tau$ as continuation parameter. Starting from $\tau = 0$ that corresponds to $\boldsymbol{C}_{\text{nom}}$, continuation proceeds with step $\delta\tau$ until $\tau = 1$ that corresponds to $\boldsymbol{C}_{\text{per}}$. At each step, the inner loop MPSP is applied to find the solution corresponding to the conditions

$$\boldsymbol{C}_{\tau} = (1 - \tau)\boldsymbol{C}_{\text{nom}} + \tau\boldsymbol{C}_{\text{per}} \tag{51}$$

Note that $\boldsymbol{C}_{\text{nom}}$ and $\boldsymbol{C}_{\text{per}}$ denote nominal and perturbed conditions, including boundary conditions and parameters, but not the entire history of perturbed control or state.

Beside, $N_{\text{seg,tol}}$ is initially set to 0 in the outer loop MPSP. Thus, the presented MPSP algorithm tries to find the updated solution with the nominal thrust sequence first. The value of $N_{\text{seg,tol}}$ increases once the continuation fails with current $N_{\text{seg,tol}}$. The outer loop MPSP fails if $\delta\tau \leq 0.1$ before $\tau$ reaches 1, or $N_{\text{seg,nom}} - N_{\text{seg,tol}} < -1$. The value of $\delta\tau$ threshold is based on numerical experiments.

The proposed two-loop MPSP guidance logic restricts the variation of thrust sequence in the inner-loop, to enhance algorithm robustness. As a side effect, more iterations are needed when the thrust sequence changes.

## 4. Numerical Simulations

### *4.1. Nominal Trajectory*

An interplanetary CubeSat mission to the asteroid 99942 Apophis leaving from the Sun–Earth L$_2$ Lagrange point is considered. The physical constants are listed in Table 1. Moreover, $m_0 = 25$ kg, $T_{\max} = 1.5 \times 10^{-3}$ N, and $I_{\text{sp}} = 3000$ s. The transfer duration is 1156 days, from 1 October 2020 to 1 December 2023. The boundary conditions are given in Table 2. The fuel-optimal transfer orbit is shown in Fig. 1, where the red line denotes the

**Algorithm 1** Inner Loop MPSP Algorithm

```
Require: $\boldsymbol{C}_\tau$, $\boldsymbol{\epsilon}$, $\lambda_{m0}$, $\Delta_{\max}$, $N_{\text{seg,tol}}$, $N_{\text{seg,nom}}$ and $h_{\max}$.
Ensure: $\boldsymbol{\epsilon}_j$, $\lambda_{m0,j}$ and label $Sign$
Integrate Eq. (16) using $\boldsymbol{C}_\tau$, $\boldsymbol{\epsilon}$, $\lambda_{m0}$, $\boldsymbol{\alpha}$ in Eq. (8), $u$ in Eq. (11) and $\lambda_v = \|\boldsymbol{\lambda}_v\|_2$, with time step $h_{\max}$ and switching time detection.
Set $j = 0$, $\boldsymbol{\epsilon}_j = \boldsymbol{\epsilon}$, $\lambda_{m0,j} = \lambda_{m0}$.
while The terminal error does not satisfy requirements do
    Extract $\{t_0, t_{1,j}, \cdots, t_{M-1,j}, t_M\}$. Interpolate the trajectory.
    if $\|\Delta \boldsymbol{Y}^N_{M-1,j}\|_2 > \Delta_{\max}$ then
        Return $Sign = 0$, $\boldsymbol{\epsilon}_j$ and $\lambda_{m0,j}$.
    end if
    Calculate $\boldsymbol{A}_{\lambda,j}$ in Eq. (33), $\boldsymbol{B}_{v,j}$ in Eq. (29), $\boldsymbol{p}_j$ in Eq. (44), $\mathrm{d}\boldsymbol{\epsilon}_j$ and $\mathrm{d}\lambda_{m0,j}$ in Eq. (45). Then calculate $\boldsymbol{\epsilon}_{j+1}$ and $\lambda_{m0,j+1}$ in Eq. (46).
    Integrate Eq. (16) using $\boldsymbol{C}_\tau$, $\boldsymbol{\epsilon}_{j+1}$, $\lambda_{m0,j+1}$, $\boldsymbol{\alpha}$ in Eq. (8), $u$ in Eq. (11) and $\lambda_v = \|\boldsymbol{\lambda}_v\|_2$, with time step $h_{\max}$ and switching time detection. Then compute $N_{\text{seg},j}$.
    if $|N_{\text{seg},j} - N_{\text{seg,nom}}| \leq N_{\text{seg,tol}}$ then
        $j \coloneqq j + 1$.
    else
        Return $Sign = 0$, $\boldsymbol{\epsilon}_j$ and $\lambda_{m0,j}$.
    end if
end while
Return $Sign = 1$, $\boldsymbol{\epsilon}_j$ and $\lambda_{m0,j}$.
```

**Algorithm 2** Outer Loop MPSP Algorithm

```
Ensure: $\boldsymbol{\epsilon}_k$ and $\lambda_{m0,k}$.
Set $\Delta_{\max}$, $h_{\max}$, $\delta\tau_0$, $N_{\text{seg,tol}}$. $k = 0$.
Solve the fuel-optimal low-thrust transfer problem [2].
Calculate $\boldsymbol{\epsilon}_0$, $\boldsymbol{C}_{\text{per}}$, $\lambda_{m0,0}$ and $N_{\text{seg,nom}}$.
Implement inner loop MPSP with input $\boldsymbol{C}_{\text{per}}$, $\boldsymbol{\epsilon}_k$, $\lambda_{m0,k}$, $\Delta_{\max}$, $N_{\text{seg,tol}}$, $N_{\text{seg,nom}}$, $h_{\max}$, and output $\boldsymbol{\epsilon}_{k+1}$, $\lambda_{m0,k+1}$, $Sign1$.
if $Sign1 = 0$ then
    while $N_{\text{seg,nom}} - N_{\text{seg,tol}} \geq -1$ do
        Set $\tau = \delta\tau_0$, $\delta\tau = \delta\tau_0$, $\tau_{\text{old}} = 0$, $k = 0$.
        while $\delta\tau > 0$ do
            Calculate $\boldsymbol{C}_\tau$ in Eq. (51).
            Implement inner loop MPSP with input $\boldsymbol{C}_\tau$, $\boldsymbol{\epsilon}_k$, $\lambda_{m0,k}$, $\Delta_{\max}$, $N_{\text{seg,tol}}$, $N_{\text{seg,nom}}$, $h_{\max}$, and output $\boldsymbol{\epsilon}_{k+1}$, $\lambda_{m0,k+1}$, $Sign2$.
            if $Sign2 = 0$ then
                $\delta\tau := \delta\tau/2$.
            else
                $\delta\tau := \min(1-\tau, 2\delta\tau)$, $\tau_{\text{old}} = \tau$, $k := k+1$.
            end if
            $\tau = \tau_{\text{old}} + \delta\tau$.
            if $\delta\tau \leq 0.1$ and $\tau_{\text{old}} \neq 1$ then
                Break.
            end if
        end while
        if $\delta\tau = 0$ then
            Return $\boldsymbol{\epsilon}_k$, $\lambda_{m0,k}$.
        else
            $N_{\text{seg,tol}} := N_{\text{seg,tol}} + 2$.
        end if
    end while
    Return failure information.
else
    Return $\boldsymbol{\epsilon}_k$, $\lambda_{m0,k}$.
end if
```

Table 1: Physical constants.

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

Table 2: Boundary Conditions.

| Boundary Conditions | Values |
|---|---|
| Initial position vector (LU) | $\boldsymbol{r}_0 = [1.001367, 0.140622, -6.594513 \times 10^{-6}]^\top$ |
| Initial velocity vector (VU) | $\boldsymbol{v}_0 = [-0.155386, 0.986258, -4.827818 \times 10^{-5}]^\top$ |
| Terminal position vector (LU) | $\boldsymbol{r}_f = [-1.044138, -0.122918, -0.018183]^\top$ |
| Terminal velocity vector (VU) | $\boldsymbol{v}_f = [0.222668, -0.875235, 0.051944]^\top$ |

thrust segment ($u = 1$), and the blue dashed line denotes the coast segment ($u = 0$). The thrust throttle $u$, the switching function $S$ and the mass $m$ as function of time are shown in Fig. 2. The optimal trajectory consists of five thrust segments and four coast segments, and the final spacecraft mass is 21.062 kg.

In the following numerical simulations, Fourier basis functions (see Appendix C) up to the 15th order are used to approximate the MPSP control profile. The convergence conditions to terminate the algorithm are such that the terminal position error $\|\Delta \boldsymbol{r}_f\|_2 \leq 500$ km, the terminal velocity error $\|\Delta \boldsymbol{v}_f\|_2 \leq$ 0.1 km/s and $|\lambda_{mf}| \leq 10^{-6}$. For the other parameters settings, $h_{\max} = 5 \times 10^{-4}\ t_f$, $\eta_0 = -\pi$, $\eta_f = \pi$, $\Delta_{\max} = 1$, initial $\delta\tau$ as $\delta\tau_0 = 0.5$, $\boldsymbol{R}_\epsilon$ is identity matrix and $R_\lambda = 1$. In the open-loop solver, the step of $\zeta$ in Eq. (48) is set to 0.05. All simulations are conducted under an Intel Core i7-9750H, CPU@2.6GHz, Windows 10 system with MATLAB R2019a. The inner loop MPSP code is converted to MEX file to speed up simulations.

### 4.2. Perturbations on Initial Conditions

The proposed MPSP guidance scheme is tested by assuming perturbations on the initial conditions. Different perturbation magnitudes for

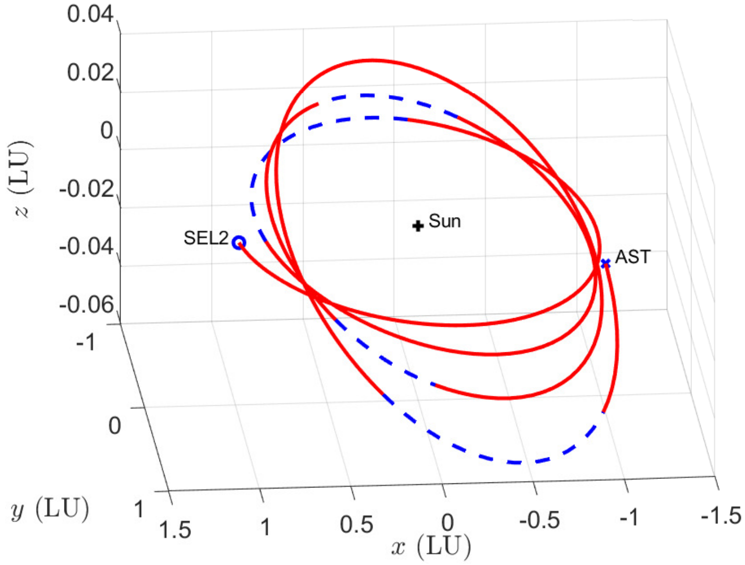


Figure 1: Fuel-optimal trajectory for boundary conditions in Table 2; SEL2 denotes Sun–Earth $L_2$ Lagrange point and AST denotes the asteroid position upon arrival.

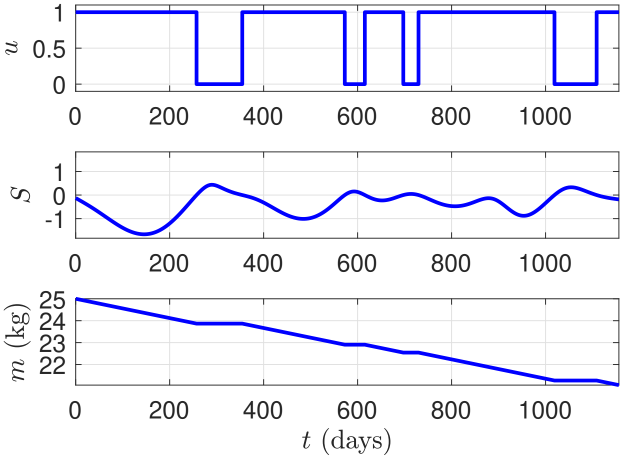


Figure 2: Thrust throttle $u$, switching function $S$, and mass $m$ profiles corresponding to the fuel-optimal trajectory in Fig. 1.

the same basic perturbation case are simulated. The basic case is randomly generated, with perturbations on initial position and velocity as $\delta\boldsymbol{r}_0 = [1.6712, -1.0659, -4.1460] \times 10^{-2}$ LU and $\delta\boldsymbol{v}_0 = [1.0876, 2.3763, 4.6092] \times 10^{-2}$ VU. The perturbed initial conditions are $\hat{\boldsymbol{r}}_0 = \boldsymbol{r}_0 + \kappa\ \delta\boldsymbol{r}_0$ and $\hat{\boldsymbol{v}}_0 = \boldsymbol{v}_0 + \kappa\ \delta\boldsymbol{v}_0$ with $\kappa \in [-3, -2.5, -2, -1, 1, 2, 2.5, 3, 3.5]$, which corresponds to 9 cases. Cases 1–4 have opposite perturbation direction w.r.t. cases 5–9. The simulation results are summarized in Table 3 which gives the terminal position and velocity errors, the total Newton's iteration[4] in the inner loop MPSP (NI), the percentage of fuel increase w.r.t. the corresponding optimal solutions (FI), the computational time for both the MPSP (MPSP-CT) and indirect method (TPBVP-CT). The same abbreviations are used in the remainder. It can be observed that the algorithm works successfully since the terminal errors are all within the tolerance.

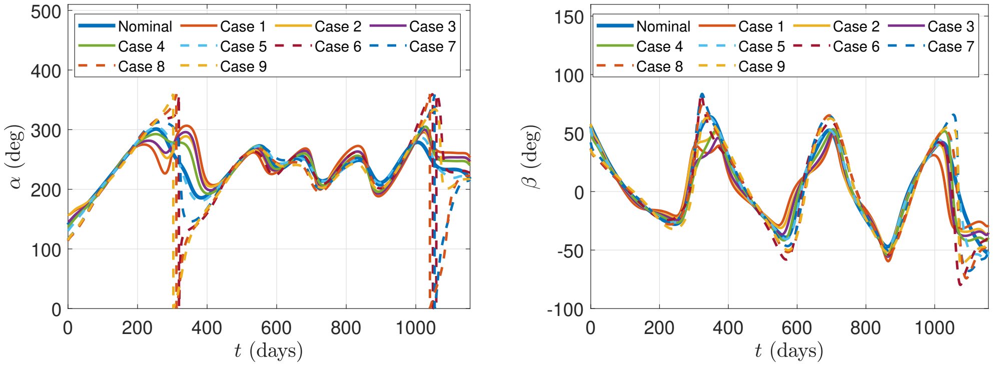
The comparisons of thrust angles between the MPSP solutions and the nominal solution are shown in Fig. 3. It can be seen that the variations of $\alpha$ for cases 5–9 are more obvious than that of cases 1–4, while the $\beta$ oscillations for all cases remain in the vicinity of the nominal solution. Note that in Fig 3a, the abrupt changes of $\alpha$ are caused by the definition of angle range. Since most of them are locate inside the coast arcs, these variations contain no useful information to the analysis.

The comparisons of thrust sequences among the MPSP solutions, the corresponding fuel-optimal solutions and the nominal solution are shown in Fig. 4. We can see that the optimal solutions gradually produce new coast segments from case 4 to 1, while the MPSP solutions retain the nominal thrust sequence. However, for cases 1–4, it is clear from Fig. 4 that the time duration of each thrust segment is updated to satisfy the perturbed conditions. From case 5 to 9, the initial conditions are becoming tighter, and more actuation is required to drive the spacecraft to the target. The MPSP solutions and optimal solutions show a similar trend, i.e., they gradually increase the number of the thrust segments.

The deviations between the MPSP solutions and the nominal solution on coordinates are shown in Fig. 5. It is interesting to see that the deviations are symmetric for the two groups with opposite directions of initial perturbations.

From Table 3, case 4 requires the fewest NI and the outer loop MPSP is not triggered. On the other hand, case 9 requires the highest NI, since

[4] Specifically, it refers to the computational times of Eq. (45).



(a) (b)

Figure 3: In-plane angle $\alpha$ and out-of-plane angle $\beta$ of the converged MPSP trajectories for variations of initial conditions in Table 3.

the thrust profile changes considerably compared with the nominal thrust sequence as shown in Fig. 4. For cases $\kappa = -3$ and $\kappa = 3$, even though the perturbation magnitudes are the same, case $\kappa = 3$ requires nearly twice NI than case $\kappa = -3$, because case $\kappa = 3$ requires two coast segments fewer than that of the nominal thrust sequence, whereas the thrust sequence of MPSP solution for case $\kappa = -3$ retains the nominal thrust sequence. Case 1 is the most expensive featuring FI by 6.09%, while the cheapest is case 5 with FI just 0.16%. As for computational time, MPSP-CT has a positive correlation with iterations. TPBVP-CT is steady, except for cases 2 and 5 where the nearly impulsive segment requires smaller continuation step and thus more iterations. In case 4 where only inner loop MPSP is used, MPSP-CT is much faster than TPBVP-CT. From Table 3, MPSP-CT is faster than TPBVP-CT if MPSP converges within 25 NI.

### *4.3. Perturbations on Terminal Conditions*

Different variations on terminal positions are simulated to test the developed method. 8 perturbed terminal positions are taken on the vertex of a cube centered at the nominal terminal position. The side length of the cube is set to 0.04 LU. The corresponding 8 cases for $\delta \boldsymbol{r}_f = [\delta x_f, \delta y_f, \delta z_f]^\top$ are shown in Table 4. The simulation results are shown in Table 5. It can be observed that all obtained terminal values are within the tolerance.

Fig. 6 represents the variations of thrust angles $\alpha$ and $\beta$. It can be seen that the obtained angle variations are smooth and close to the nominal case.

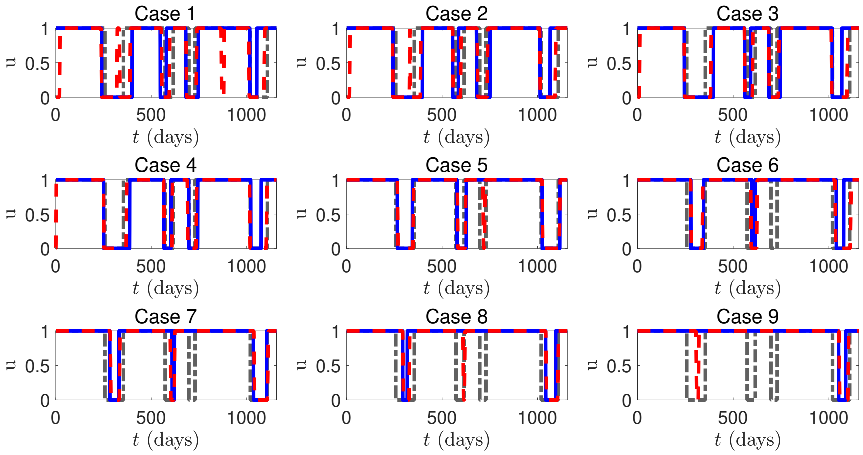


Figure 4: Comparison of optimal thrust sequences and thrust sequences of MPSP solutions for cases in Table 3. Red dashed line: optimal thrust sequences; Blue line: thrust sequences of MPSP solutions; Grey dash-dot line: nominal thrust sequences.

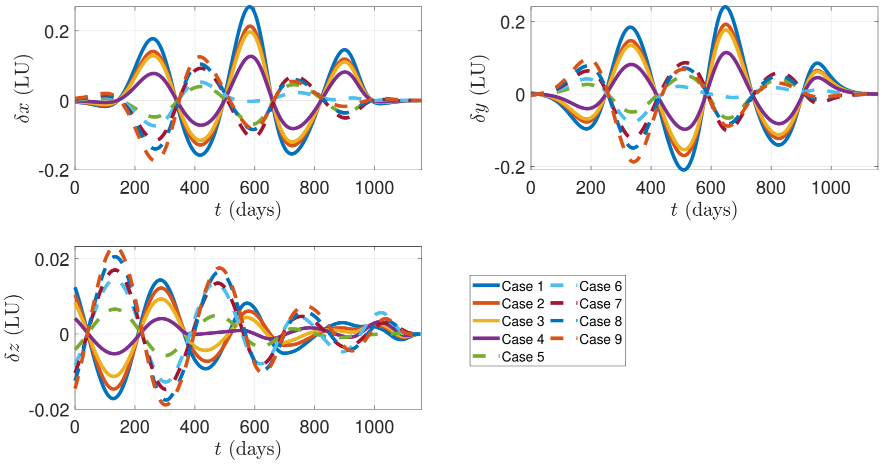


Figure 5: Deviations between converged MPSP trajectories and the nominal trajectory on coordinates for cases in Table 3.

Table 3: Simulation results for perturbations on initial conditions.

| Case | $\kappa$ | $\\|\boldsymbol{r}_f - \boldsymbol{r}(t_f)\\|_2$ | $\\|\boldsymbol{v}_f - \boldsymbol{v}(t_f)\\|_2$ | $\lvert\lambda_m(t_f)\rvert$ | NI | FI | MPSP-CT | TPBVP-CT |
|---|---|---|---|---|---|---|---|---|
| | | (km) | (km/s) | (–) | | (%) | (s) | (s) |
| 1 | $-3$ | 15.37 | $3.07 \times 10^{-6}$ | $1.43 \times 10^{-8}$ | 28 | 6.09 | 6.88 | 6.31 |
| 2 | $-2.5$ | 41.38 | $1.74 \times 10^{-5}$ | $4.21 \times 10^{-7}$ | 23 | 2.81 | 5.56 | 19.08 |
| 3 | $-2$ | 101.18 | $1.77 \times 10^{-5}$ | $2.51 \times 10^{-7}$ | 16 | 3.42 | 3.87 | 6.23 |
| 4 | $-1$ | 11.94 | $3.85 \times 10^{-6}$ | $1.21 \times 10^{-7}$ | 5 | 2.28 | 1.12 | 6.57 |
| 5 | 1 | 16.48 | $3.40 \times 10^{-6}$ | $2.34 \times 10^{-8}$ | 32 | 0.16 | 8.15 | 12.53 |
| 6 | 2 | 2.57 | $5.48 \times 10^{-7}$ | $1.54 \times 10^{-9}$ | 25 | 5.02 | 5.94 | 6.26 |
| 7 | 2.5 | 2.38 | $4.78 \times 10^{-7}$ | $9.38 \times 10^{-10}$ | 32 | 1.11 | 7.62 | 6.58 |
| 8 | 3 | 53.97 | $1.45 \times 10^{-5}$ | $2.05 \times 10^{-7}$ | 60 | 2.55 | 14.51 | 6.63 |
| 9 | 3.5 | 226.28 | $4.21 \times 10^{-5}$ | $1.56 \times 10^{-7}$ | 99 | 1.96 | 22.89 | 6.92 |

The comparison of thrust sequences among the MPSP solutions, the corresponding optimal solutions and the nominal solution are shown in Fig. 7. We notice that the MPSP solutions coincide with the optimal solutions well for most cases. For cases 1 and 3, the optimal thrust sequences feature more coast segments than the nominal solution. For cases 2, 4 and 8, MPSP solutions capture the main structure of optimal thrust sequences except some near-impulse segments. For cases 5, 6 and 7, MPSP solutions perfectly coincide with the optimal thrust sequences.

The deviation between nominal solution and the MPSP solutions on coordinates are shown in Fig. 8. The $x$ deviation for case 1–4 and case 5–8 are nearly symmetric. The $y$ deviations also show similar symmetry except the last 200 days. The $z$ deviations tend to oscillate and gradually amplify.

From Table 4, it can be observed that the fuel consumption of the MPSP solutions is very close to the corresponding optimal solutions. The minimum FI is in case 2, which is only 0.78%, while the maximum one is in case 6, which is only 2.40%. Cases 1, 3 and 7 where MPSP solutions retain the nominal thrust sequence, require around 10 NI. For cases 4, 5 and 8, MPSP requires around 20 NI since one fewer coast segment is required. Case 2 is exceptional as it only requires 13 NI. In case 6, MPSP requires the most NI because two coast segments fewer than nominal thrust sequence are required, but FI is just 2.4%. As for the computational time, MPSP-CT is superior to TPBVP-CT, except for case 6 where thrust sequence of MPSP solution has two coast segments fewer than the nominal thrust sequence.

### *4.4. Perturbations on Thruster Parameters*

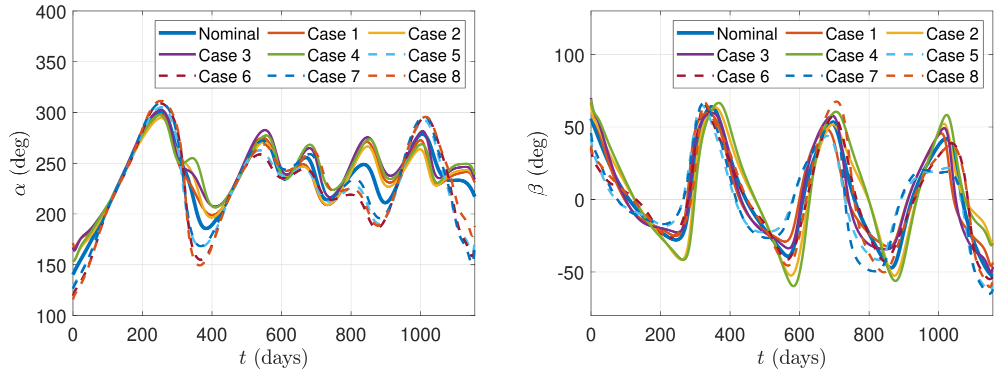
The perturbations on the thruster parameters are simulated. Specifically, the perturbations on $T_{\max}$ are tested. The percentages of the perturbations

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 | $\|\boldsymbol{r}_f - \boldsymbol{r}(t_f)\|_2$ | $\|\boldsymbol{v}_f - \boldsymbol{v}(t_f)\|_2$ | $\lvert\lambda_m(t_f)\rvert$ | NI | FI | MPSP-CT | TPBVP-CT |
|---|---|---|---|---|---|---|---|
| | (km) | (km/s) | $(-)$ | | (%) | (s) | (s) |
| 1 | 43.88 | $9.40 \times 10^{-6}$ | $6.93 \times 10^{-8}$ | 11 | 1.05 | 2.51 | 6.34 |
| 2 | 2.04 | $4.66 \times 10^{-7}$ | $2.53 \times 10^{-9}$ | 13 | 0.78 | 3.15 | 6.27 |
| 3 | 72.21 | $1.33 \times 10^{-5}$ | $1.69 \times 10^{-8}$ | 13 | 2.22 | 3.08 | 6.59 |
| 4 | 40.69 | $8.65 \times 10^{-6}$ | $2.75 \times 10^{-8}$ | 22 | 1.00 | 5.41 | 7.10 |
| 5 | 74.73 | $1.28 \times 10^{-5}$ | $1.50 \times 10^{-7}$ | 20 | 1.28 | 5.03 | 6.46 |
| 6 | 5.05 | $8.55 \times 10^{-7}$ | $1.37 \times 10^{-9}$ | 38 | 2.40 | 9.02 | 5.67 |
| 7 | 8.69 | $1.43 \times 10^{-6}$ | $6.27 \times 10^{-9}$ | 9 | 1.27 | 2.09 | 6.47 |
| 8 | 2.73 | $4.39 \times 10^{-6}$ | $4.46 \times 10^{-9}$ | 24 | 2.10 | 5.72 | 10.17 |



(a) (b)

Figure 6: In-plane angle $\alpha$ and out-of-plane angle $\beta$ of the converged MPSP trajectories for perturbations of terminal positions in Table 4.

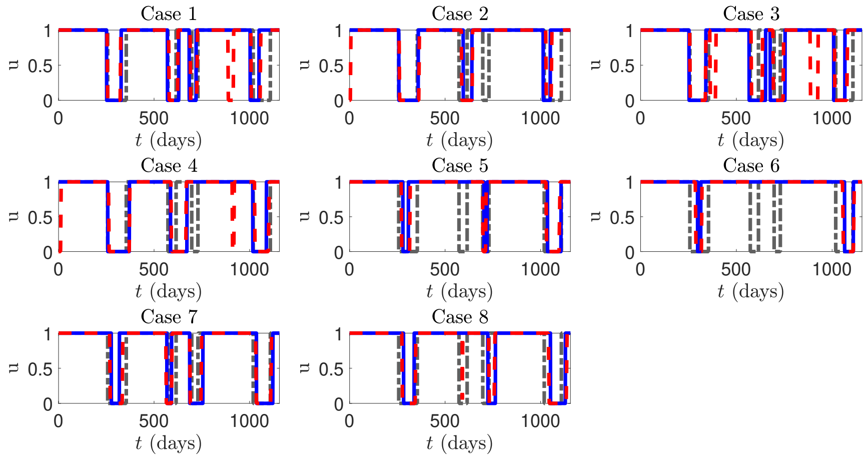


Figure 7: Comparison of optimal thrust sequences and thrust sequences of MPSP solutions for cases in Table 4. Red dashed line: optimal thrust sequences; Blue line: thrust sequences of MPSP solutions; Grey dash-dot line: nominal thrust sequences.

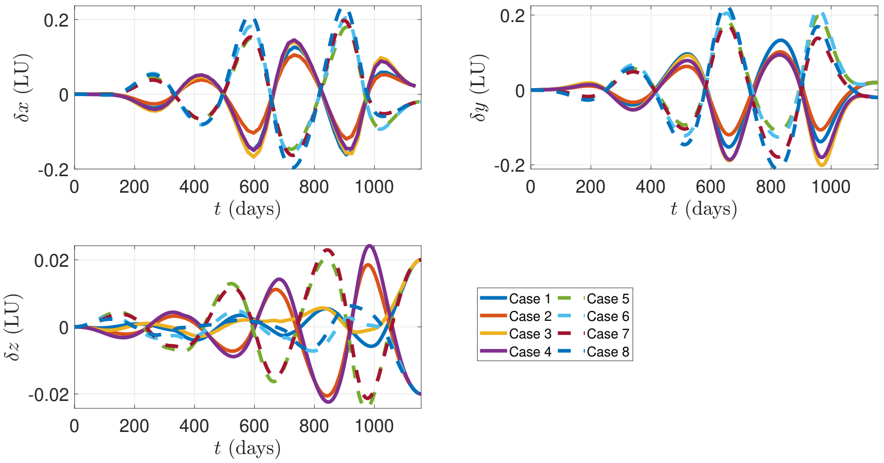


Figure 8: Deviations between converged MPSP trajectories and the nominal trajectory on coordinates for terminal position variation cases in Table 4.

Table 6: Simulation results for perturbations on $T_{\max}$.

| Case | $\eta$<br>(%) | $\|\boldsymbol{r}_f - \boldsymbol{r}(t_f)\|_2$<br>(km) | $\|\boldsymbol{v}_f - \boldsymbol{v}(t_f)\|_2$<br>(km/s) | $\|\lambda_m(t_f)\|$<br>(–) | NI | FI<br>(%) | MPSP-CT<br>(s) | TPBVP-CT<br>(s) |
|---|---|---|---|---|---|---|---|---|
| 1 | $-10$ | 0.41 | $6.60 \times 10^{-8}$ | $2.67 \times 10^{-9}$ | 43 | 1.28 | 10.30 | 5.85 |
| 2 | $-6$ | 0.79 | $1.30 \times 10^{-7}$ | $3.03 \times 10^{-9}$ | 31 | 0.19 | 7.60 | 6.29 |
| 3 | $-3$ | 4.19 | $9.58 \times 10^{-7}$ | $6.09 \times 10^{-9}$ | 9 | 0.081 | 2.11 | 6.61 |
| 4 | 3 | 2.70 | $6.42 \times 10^{-7}$ | $3.25 \times 10^{-9}$ | 4 | 0.062 | 0.90 | 6.32 |
| 5 | 6 | 4.48 | $1.01 \times 10^{-6}$ | $4.37 \times 10^{-9}$ | 5 | 0.48 | 1.08 | 6.75 |
| 6 | 10 | 6.61 | $1.47 \times 10^{-6}$ | $5.76 \times 10^{-9}$ | 10 | 0.55 | 2.37 | 6.57 |

w.r.t. the nominal solution are set to $\eta = [-10\%,\ -6\%,\ -3\%,\ 3\%,\ 6\%,\ 10\%]$, which corresponds to 6 simulation cases. It is assumed that the percentage of the perturbation on $T_{\max}$ remains the same throughout the flight. The simulation results are summarized in Table 6. For all cases, MPSP algorithm converges successfully.

In Fig. 9, we notice that the variations of $\alpha$ and $\beta$ are smooth, and they lie in the vicinity of nominal values. Fig. 10 illustrates the comparisons of thrust sequences among the MPSP solutions, the corresponding fuel-optimal solutions and the nominal solution. The thrust sequences of MPSP solutions coincide well with that of fuel-optimal solutions for all cases.

Fig. 11 depicts the trajectory deviations between converged MPSP trajectories w.r.t. the nominal trajectory on coordinates. The deviations for case 1 and 6 are the most obvious since the perturbations on $T_{\max}$ are the largest. Differently from Figs. 5 and 8, the deviations are not symmetry between cases 1 and 6.

From Table 6, it can be seen that the FI for all cases are negligible. As expected, the highest NI occurs in case 1 since the variation of the thrust sequence is maximum. It is also noticed that few NI are required when the thrust sequence remains the same as the nominal one. As for the computational time, MPSP-CT is superior to TPBVP-CT for cases 3–6 where nominal thrust sequence is remained. Competitive MPSP-CT in case 2 is achieved where one coast segment fewer than nominal thrust sequence is obtained.

### *4.5. Discussion*

The outcome of the three simulation studies indicate that: 1) the proposed MPSP is robust to various kinds of perturbations; 2) the thrust angles of the converged MPSP trajectories remain smooth; 3) the thrust sequences of MPSP solutions tends to retain the nominal thrust sequence; 4) a few Newton's iterations are required if the thrust sequences of MPSP solutions are

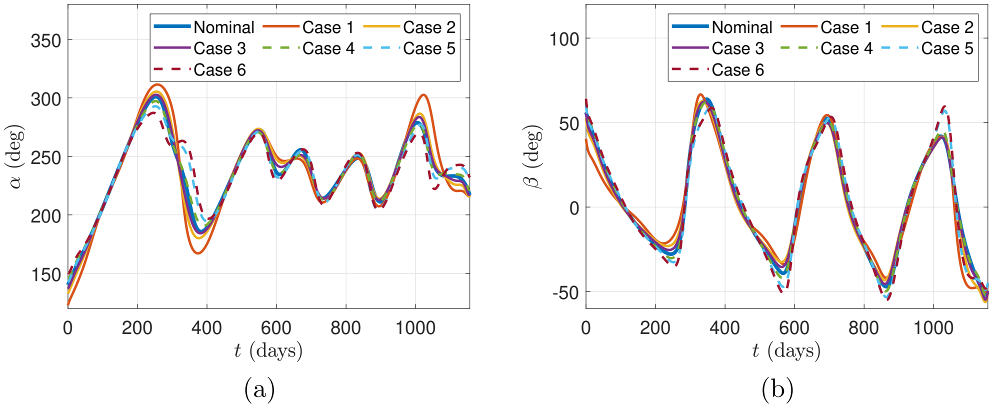


Figure 9: Variations of the in-plane angle $\alpha$ and out-of-plane angle $\beta$ for the converged MPSP trajectories for variations of $T_{\max}$ in Table 6.

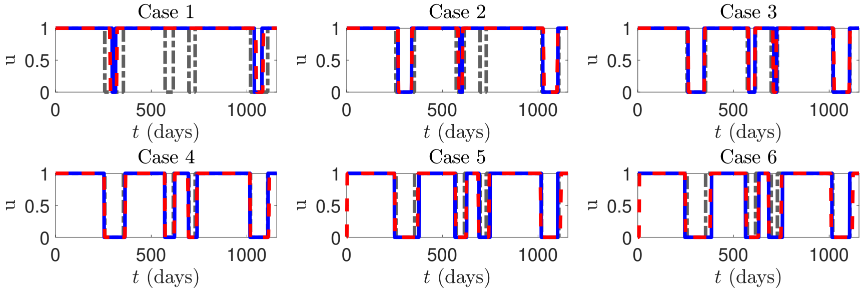


Figure 10: Comparison of optimal thrust sequences and thrust sequences of MPSP solutions for cases in Table 6. Red dashed line: optimal thrust sequences; Blue line: thrust sequences of MPSP solutions; Grey dash-dot line: nominal thrust sequences..

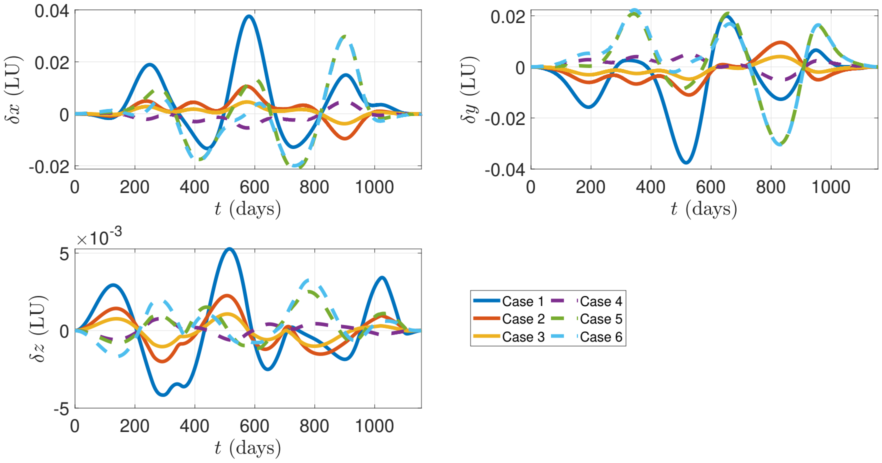


Figure 11: Deviations between converged MPSP trajectories and the nominal trajectory on coordinates for cases of perturbations on $T_{\max}$ in Table. 6.

not or slightly changed w.r.t. the nominal thrust sequence; 5) even though the fuel consumption is not included in the MPSP performance index as Eq. (41), MPSP solutions are near-optimal in terms of fuel consumption even when the thrust sequence changes. 6) MPSP-CT is superior to TPBVP-CT if NI $\leq 25$. However, the real-time capability should be further verified using processor-in-loop simulations [37]. Overall, evidence is given that the developed bang-off-bang MPSP method is effective in providing robust guidance laws.

## 5. Conclusion

This paper extended applications of model predictive static programming to low-thrust transfers with bang-off-bang control. Based on the fact that the sensitive matrix is discontinuous across the switching times, a robust two-loop MPSP guidance scheme is designed. The main feature of the proposed MPSP method is its capability to generate near fuel-optimal solutions featuring bang-off-bang control, even when the thrust sequence changes. Besides, few iterations are required if the thrust sequence is not or slightly changed compared to the nominal thrust sequence. A number of simulations demonstrated the robustness of the proposed method for various kinds of perturbations.

Future works will focus on 1) including the fuel consumption into MPSP design to further improve its performance and robustness, 2) reducing the computational time and iterations especially when the thrust sequence changes, and 3) validating the real-time capability through processor-in-loop simulations [37].

## Acknowledgment

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

## Conflict of Interest Statement

We have no conflict of interest to declare.

## Appendix A. System Discretization

The classic $4^{\text{th}}$ order Runge–Kutta formulae is used to discretize the trajectory. Suppose the solution $\boldsymbol{X}_n$ is known at $t_n$, the issue is to compute $\boldsymbol{X}_{n+1}$ at $t_{n+1} = t_n + h$, where $h$ is the time step. The dynamics in Eq. (16) can be written in the discretized form as follows

$$\begin{cases} \boldsymbol{X}_{n+1} & = \boldsymbol{X}_n + \dfrac{h}{6}\left(\boldsymbol{K}_1 + 2\boldsymbol{K}_2 + 2\boldsymbol{K}_3 + \boldsymbol{K}_4\right) \\ \boldsymbol{K}_1 & = \boldsymbol{\mathcal{F}}(t_{n,1}, \boldsymbol{X}_{n,1}, \boldsymbol{U}_n), \quad t_{n,1} = t_n,\ \boldsymbol{X}_{n,1} = \boldsymbol{X}_n \\ \boldsymbol{K}_2 & = \boldsymbol{\mathcal{F}}(t_{n,2}, \boldsymbol{X}_{n,2}, \boldsymbol{U}_n), \quad t_{n,2} = t_n + \dfrac{h}{2},\ \boldsymbol{X}_{n,2} = \boldsymbol{X}_n + \dfrac{h}{2}\boldsymbol{K}_1 \\ \boldsymbol{K}_3 & = \boldsymbol{\mathcal{F}}(t_{n,3}, \boldsymbol{X}_{n,3}, \boldsymbol{U}_n), \quad t_{n,3} = t_n + \dfrac{h}{2},\ \boldsymbol{X}_{n,3} = \boldsymbol{X}_n + \dfrac{h}{2}\boldsymbol{K}_2 \\ \boldsymbol{K}_4 & = \boldsymbol{\mathcal{F}}(t_{n,4}, \boldsymbol{X}_{n,4}, \boldsymbol{U}_n), \quad t_{n,4} = t_n + h,\ \boldsymbol{X}_{n,4} = \boldsymbol{X}_n + h\boldsymbol{K}_3 \end{cases} \tag{A.1}$$

The discretized system dynamics is written as

$$\boldsymbol{X}_{n+1} = \boldsymbol{F}_n(t_n, \boldsymbol{X}_n, \boldsymbol{U}_n) \tag{A.2}$$

where

$$\boldsymbol{F}_n(t_n, \boldsymbol{X}_n, \boldsymbol{U}_n) = \boldsymbol{X}_n + \frac{h}{6}\left(\boldsymbol{K}_1 + 2\boldsymbol{K}_2 + 2\boldsymbol{K}_3 + \boldsymbol{K}_4\right)$$

The differential of $\boldsymbol{F}_n(t_n, \boldsymbol{X}_n, \boldsymbol{U}_n)$ w.r.t. $\boldsymbol{X}_n$ is

$$\frac{\mathrm{d}\boldsymbol{F}_n(t_n, \boldsymbol{X}_n, \boldsymbol{U}_n)}{\mathrm{d}\boldsymbol{X}_n} = \boldsymbol{I}_n + \frac{h}{6}\left(\frac{\mathrm{d}\boldsymbol{K}_1}{\mathrm{d}\boldsymbol{X}_n} + 2\frac{\mathrm{d}\boldsymbol{K}_2}{\mathrm{d}\boldsymbol{X}_n} + 2\frac{\mathrm{d}\boldsymbol{K}_3}{\mathrm{d}\boldsymbol{X}_n} + \frac{\mathrm{d}\boldsymbol{K}_4}{\mathrm{d}\boldsymbol{X}_n}\right) \tag{A.3}$$

where

$$\frac{\mathrm{d}\boldsymbol{K}_i}{\mathrm{d}\boldsymbol{X}_n} = \frac{\partial \boldsymbol{\mathcal{F}}(t_{n,i}, \boldsymbol{X}_{n,i})}{\partial \boldsymbol{X}_{n,i}} \frac{\mathrm{d}\boldsymbol{X}_{n,i}}{\mathrm{d}\boldsymbol{X}_n}, \quad i = 1, 2, 3, 4$$

$$\frac{\mathrm{d}\boldsymbol{X}_{n,1}}{\mathrm{d}\boldsymbol{X}_n} = \boldsymbol{I}_{8\times 8}, \quad \frac{\mathrm{d}\boldsymbol{X}_{n,2}}{\mathrm{d}\boldsymbol{X}_n} = \boldsymbol{I}_{8\times 8} + \frac{h}{2}\frac{\mathrm{d}\boldsymbol{K}_1}{\mathrm{d}\boldsymbol{X}_n},$$
$$\frac{\mathrm{d}\boldsymbol{X}_{n,3}}{\mathrm{d}\boldsymbol{X}_n} = \boldsymbol{I}_{8\times 8} + \frac{h}{2}\frac{\mathrm{d}\boldsymbol{K}_2}{\mathrm{d}\boldsymbol{X}_n}, \quad \frac{\mathrm{d}\boldsymbol{X}_{n,4}}{\mathrm{d}\boldsymbol{X}_n} = \boldsymbol{I}_{8\times 8} + h\frac{\mathrm{d}\boldsymbol{K}_3}{\mathrm{d}\boldsymbol{X}_n},$$

$$\frac{\partial \boldsymbol{\mathcal{F}}(t, \boldsymbol{X}, \boldsymbol{U})}{\partial \boldsymbol{X}} = \begin{bmatrix} \boldsymbol{0}_{3\times 3} & \boldsymbol{I}_{3\times 3} & \boldsymbol{0}_{3\times 1} & \boldsymbol{0}_{3\times 1} \\ \boldsymbol{G}(\boldsymbol{r}) & \boldsymbol{0}_{3\times 3} & u\dfrac{T_{\max}}{m^2\lambda_v}\boldsymbol{\lambda}_v & \boldsymbol{0}_{3\times 1} \\ \boldsymbol{0}_{1\times 3} & \boldsymbol{0}_{1\times 3} & 0 & 0 \\ \boldsymbol{0}_{1\times 3} & \boldsymbol{0}_{1\times 3} & 2u\dfrac{T_{\max}}{m^3}\lambda_v & 0 \end{bmatrix}$$

The differential of $\boldsymbol{F}_n(t_n, \boldsymbol{X}_n, \boldsymbol{U}_n)$ w.r.t. $\boldsymbol{U}_n$ is

$$\frac{\mathrm{d}\boldsymbol{F}_n(t_n, \boldsymbol{X}_n, \boldsymbol{U}_n)}{\mathrm{d}\boldsymbol{U}_n} = \frac{h}{6}\left(\frac{\mathrm{d}\boldsymbol{K}_1}{\mathrm{d}\boldsymbol{U}_n} + 2\frac{\mathrm{d}\boldsymbol{K}_2}{\mathrm{d}\boldsymbol{U}_n} + 2\frac{\mathrm{d}\boldsymbol{K}_3}{\mathrm{d}\boldsymbol{U}_n} + \frac{\mathrm{d}\boldsymbol{K}_4}{\mathrm{d}\boldsymbol{U}_n}\right) \tag{A.4}$$

where

$$\frac{\mathrm{d}\boldsymbol{K}_i}{\mathrm{d}\boldsymbol{U}_n} = \frac{\partial \boldsymbol{\mathcal{F}}(t_{n,i}, \boldsymbol{X}_{n,i}, \boldsymbol{U}_{n,i})}{\partial \boldsymbol{U}_{n,i}} + \frac{\partial \boldsymbol{\mathcal{F}}(t_{n,i}, \boldsymbol{X}_{n,i}, \boldsymbol{U}_{n,i})}{\partial \boldsymbol{X}_{n,i}} \frac{\mathrm{d}\boldsymbol{X}_{n,i}}{\mathrm{d}\boldsymbol{U}_{n,i}}, \quad i = 1, 2, 3, 4$$

$$\frac{\mathrm{d}\boldsymbol{X}_{n,1}}{\mathrm{d}\boldsymbol{U}_n} = \boldsymbol{0}_{8\times 3}, \quad \frac{\mathrm{d}\boldsymbol{X}_{n,2}}{\mathrm{d}\boldsymbol{X}_n} = \frac{h}{2}\frac{\mathrm{d}\boldsymbol{K}_1}{\mathrm{d}\boldsymbol{U}_n},$$
$$\frac{\mathrm{d}\boldsymbol{X}_{n,3}}{\mathrm{d}\boldsymbol{X}_n} = \frac{h}{2}\frac{\mathrm{d}\boldsymbol{K}_2}{\mathrm{d}\boldsymbol{U}_n}, \quad \frac{\mathrm{d}\boldsymbol{X}_{n,4}}{\mathrm{d}\boldsymbol{X}_n} = h\frac{\mathrm{d}\boldsymbol{K}_3}{\mathrm{d}\boldsymbol{U}_n},$$

$$\frac{\partial \boldsymbol{\mathcal{F}}(t, \boldsymbol{X}, \boldsymbol{U})}{\partial \boldsymbol{U}} = \begin{bmatrix} \boldsymbol{0}_{3\times 3} \\ u\dfrac{T_{\max}}{m}\left(-\dfrac{1}{\lambda_v}\boldsymbol{I}_{3\times 3} + \dfrac{1}{\lambda_v^3}\boldsymbol{\lambda}_v\boldsymbol{\lambda}_v^\top\right) \\ \boldsymbol{0}_{1\times 3} \\ -u\dfrac{T_{\max}}{m^2\lambda_v}\boldsymbol{\lambda}_v^\top \end{bmatrix}$$

## Appendix B. Derivation of Eqs. (24) and (25)

Suppose that at the switching time $t_s$, the switching function $S$ satisfies $S(\boldsymbol{X}(t_s), \boldsymbol{U}(t_s), t_s) = 0$. On the perturbed solution, it must be $S(\boldsymbol{X}(t_s + \mathrm{d}t_s), \boldsymbol{U}(t_s + \mathrm{d}t_s), t_s + \mathrm{d}t_s) = 0$. Expanding $S$ at $t_s$ yields

$$\begin{aligned}
\mathrm{d}S &= \frac{\partial S}{\partial \boldsymbol{X}} \mathrm{d}\boldsymbol{X}^- + \frac{\partial S}{\partial \boldsymbol{U}} \mathrm{d}\boldsymbol{U}^- + \frac{\partial S}{\partial t_s} \mathrm{d}t_s \\
&= \frac{\partial S}{\partial \boldsymbol{X}} \delta\boldsymbol{X}^- + \frac{\partial S}{\partial \boldsymbol{X}} \dot{\boldsymbol{X}}^- \delta t_s + \frac{\partial S}{\partial \boldsymbol{U}} \delta\boldsymbol{U}^- + \frac{\partial S}{\partial \boldsymbol{U}} \dot{\boldsymbol{U}}^- \delta t_s + \frac{\partial S}{\partial t_s} \delta t_s \\
&= \frac{\partial S}{\partial \boldsymbol{X}} \delta\boldsymbol{X}^- + \frac{\partial S}{\partial \boldsymbol{U}} \delta\boldsymbol{U}^- + \dot{S} \delta t_s \\
&= 0
\end{aligned} \tag{B.1}$$

where $\delta\boldsymbol{X}^-$ and $\delta\boldsymbol{U}^-$ are the state and control variations immediately before the switching time. Then we have

$$\delta t_s = \left( \frac{\partial S}{\partial \boldsymbol{X}} \delta\boldsymbol{X}^- + \frac{\partial S}{\partial \boldsymbol{U}} \delta\boldsymbol{U}^- \right) / (-\dot{S}) \tag{B.2}$$

For the case of dynamical discontinuity caused by control discontinuity, the state variation immediately after the switching time is [36]

$$\begin{aligned}
\delta\boldsymbol{X}^+ &= \delta\boldsymbol{X}^- + \left( \dot{\boldsymbol{X}}^- - \dot{\boldsymbol{X}}^+ \right) \delta t_s \\
&= \left[ \boldsymbol{I} + \left( \dot{\boldsymbol{X}}^+ - \dot{\boldsymbol{X}}^- \right) \frac{\partial S}{\partial \boldsymbol{X}} / \dot{S} \right] \delta\boldsymbol{X}^- + \left[ \left( \dot{\boldsymbol{X}}^+ - \dot{\boldsymbol{X}}^- \right) \frac{\partial S}{\partial \boldsymbol{U}} / \dot{S} \right] \delta\boldsymbol{U}^-
\end{aligned} \tag{B.3}$$

Thus

$$\frac{\partial \boldsymbol{X}^+}{\partial \boldsymbol{X}^-} = \boldsymbol{I} + \left( \dot{\boldsymbol{X}}^+ - \dot{\boldsymbol{X}}^- \right) S_{\boldsymbol{X}} / \dot{S} \tag{B.4}$$

$$\frac{\partial \boldsymbol{X}^+}{\partial \boldsymbol{U}^-} = \left( \dot{\boldsymbol{X}}^+ - \dot{\boldsymbol{X}}^- \right) S_{\boldsymbol{U}} / \dot{S} \tag{B.5}$$

Equations (B.4) and (B.5) require $\boldsymbol{X}(t_s)$ and $\boldsymbol{U}(t_s)$ at the switching time $t_s$. Since the switching time detection is embedded into dynamical integration (see Section 3.4), the values of $\boldsymbol{X}(t_s)$ and $\boldsymbol{U}(t_s)$ are obtained after the integration of Eq. (16).

## Appendix C. Fourier Basis Functions

The Fourier series expansion for scalar $x(\eta)$ with maximum order $n_x$ is

$$x(\eta) = \frac{a_0}{2} + \sum_{n=1}^{n_x} \{a_n \cos(n\pi\eta) + b_n \sin(n\pi\eta)\} = \boldsymbol{h}(\eta)^\top \boldsymbol{\epsilon} \tag{C.1}$$

where

$$\begin{aligned} \boldsymbol{h}(\eta) &= [\frac{1}{2}, \cos(\pi\eta), \sin(\pi\eta), \cdots, \cos(n_x\pi\eta), \sin(n_x\pi\eta)]^\top \in \mathbb{R}^{2n_x+1} \\ \boldsymbol{\epsilon} &= [a_0, a_1, b_1, \cdots, a_{n_x}, b_{n_x}]^\top \in \mathbb{R}^{2n_x+1} \end{aligned}$$

are used in Eqs. (36) and (37).

## References

[1] J. T. Betts, Practical methods for optimal control and estimation using nonlinear programming, second ed., Society for Industrial and Applied Mathematics, 2010. doi:10.1137/1.9780898718577.

[2] C. Zhang, F. Topputo, F. Bernelli-Zazzera, Y. Zhao, Low-thrust minimum-fuel optimization in the circular restricted three-body problem, Journal of Guidance, Control, and Dynamics 38 (2015) 1501–1510. doi:10.2514/1.G001080.

[3] Z. Zhu, Q. Gan, X. Yang, Y. Gao, Solving fuel-optimal low-thrust orbital transfers with bang-bang control using a novel continuation technique, Acta Astronautica 137 (2017) 98–113. doi:10.1016/j.actaastro.2017.03.032.

[4] T. Haberkorn, P. Martinon, J. Gergaud, Low-thrust minimum-fuel orbital transfer: a homotopic approach, Journal of Guidance, Control, and Dynamics 27 (2004) 1046–1060. doi:10.2514/1.4022.

[5] M. B. Quadrelli, L. J. Wood, J. E. Riedel, M. C. McHenry, M. Aung, L. A. Cangahuala, R. A. Volpe, P. M. Beauchamp, J. A. Cutts, Guidance, navigation, and control technology assessment for future planetary science missions, Journal of Guidance, Control, and Dynamics 38 (2015) 1165–1186. doi:10.2514/1.G000525.

[6] A. E. Bryson, Y.-C. Ho, Applied optimal control: optimization, estimation and control, Taylor and Francis, 1975. doi:10.1201/9781315137667.

[7] A. L. Kornhauser, P. M. Lion, Optimal deterministic guidance for bounded-thrust spacecrafts, Celestial mechanics 5 (1972) 261–281. doi:10.1007/BF01228429.

[8] Z. Chen, Neighboring optimal control for fixed-time multi-burn orbital transfers, Aerospace Science and Technology 61 (2017) 57–65. doi:10.1016/j.ast.2016.11.021.

[9] Z. Chen, S. Tang, Neighboring optimal control for open-time multiburn orbital transfers, Aerospace Science and Technology 74 (2018) 37–45. doi:10.1016/j.ast.2018.01.003.

[10] P. Di Lizia, R. Armellin, A. Morselli, F. Bernelli-Zazzera, High order optimal feedback control of space trajectories with bounded control, Acta Astronautica 94 (2014) 383–394. doi:10.1016/j.actaastro.2013.02.011.

[11] Y. Gao, Low-thrust nonlinear guidance by tracking mean orbital elements, Journal of Guidance, Control, and Dynamics 31 (2008) 1103–1110. doi:10.2514/1.31256.

[12] R. C. Huang, I. Hwang, M. J. Corless, Nonlinear algorithm for tracking interplanetary low-thrust trajectories, Journal of Guidance, Control, and Dynamics 35 (2012) 696–700. doi:10.2514/1.53001.

[13] O. Halbe, R. G. Raja, R. Padhi, Robust reentry guidance of a reusable launch vehicle using model predictive static programming, Journal of Guidance, Control, and Dynamics 37 (2013) 134–148. doi:10.2514/1.61615.

[14] K. Malanowski, H. Maurer, Sensitivity analysis for parametric control problems with control-state constraints, Computational Optimization and Applications 5 (1996) 253–283. doi:10.1007/BF00248267.

[15] H. J. Pesch, Real-time computation of feedback controls for constrained optimal control problems. part 1: neighbouring extremals, Optimal Control Applications and Methods 10 (1989) 129–145. doi:10.1002/oca.4660100205.

[16] H. J. Pesch, 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 10 (1989) 147–171. doi:10.1002/oca.4660100206.

[17] Z. Wang, M. J. Grant, Minimum-fuel low-thrust transfers for spacecraft: A convex approach, IEEE Transactions on Aerospace and Electronic Systems 54 (2018) 2274–2290. doi:10.1109/TAES.2018.2812558.

[18] Y. M. Agamawi, W. W. Hager, A. V. Rao, Mesh refinement method for solving bang-bang optimal control problems using direct collocation, 2019. arXiv:1905.11895.

[19] R. Furfaro, D. Mortari, Least-squares solution of a class of optimal space guidance problems via theory of connections, Acta Astronautica 168 (2020) 92–103. doi:10.1016/j.actaastro.2019.05.050.

[20] H. Johnston, E. Schiassi, R. Furfaro, D. Mortari, Fuel-efficient powered descent guidance on large planetary bodies via theory of functional connections, 2020. arXiv:2001.03572.

[21] R. W. Beard, G. N. Saridis, J. T. Wen, Galerkin approximations of the generalized hamilton-jacobi-bellman equation, Automatica 33 (1997) 2159–2177. doi:10.1016/S0005-1098(97)00128-3.

[22] C. Park, D. J. Scheeres, Determination of optimal feedback terminal controllers for general boundary conditions using generating functions, Automatica 42 (2006) 869–875. doi:10.1016/j.automatica.2006.01.015.

[23] S. R. Vadali, R. Sharma, Optimal finite-time feedback controllers for nonlinear systems with terminal constraints, Journal of Guidance, Control, and Dynamics 29 (2006) 921–928. doi:10.2514/1.16790.

[24] A. Heydari, S. Balakrishnan, Adaptive critic-based solution to an orbital rendezvous problem, Journal of Guidance, Control, and Dynamics 37 (2014) 344–350. doi:10.2514/1.60553.

[25] R. Sharma, G. W. York, Near optimal finite-time terminal controllers for space trajectories via SDRE-based approach using dynamic programming, Aerospace Science and Technology 75 (2018) 128–138. doi:10.1016/j.ast.2017.12.022.

[26] R. Padhi, M. Kothari, Model predictive static programming: a computationally efficient technique for suboptimal control design, International journal of innovative computing, information and control 5 (2009) 399–411.

[27] A. Maity, H. B. Oza, R. Padhi, Generalized model predictive static programming and angle-constrained guidance of air-to-ground missiles, Journal of Guidance, Control, and Dynamics 37 (2014) 1897–1913. doi:10.2514/1.G000038.

[28] H. B. Oza, R. Padhi, Impact-angle-constrained suboptimal model predictive static programming guidance of air-to-ground missiles, Journal of Guidance, Control, and Dynamics 35 (2012) 153–164. doi:10.2514/1.53647.

[29] B. Zhang, S. Tang, B. Pan, Multi-constrained suboptimal powered descent guidance for lunar pinpoint soft landing, Aerospace Science and Technology 48 (2016) 203–213. doi:10.1016/j.ast.2015.11.018.

[30] Y. Wang, H. Hong, S. Tang, Geometric control with model predictive static programming on SO (3), Acta Astronautica 159 (2019) 471–479. doi:10.1016/j.actaastro.2019.01.023.

[31] S. Mondal, R. Padhi, Angle-constrained terminal guidance using quasi-spectral model predictive static programming, Journal of Guidance, Control, and Dynamics 41 (2017) 783–791. doi:10.2514/1.G002893.

[32] C. M. Sakode, R. Padhi, Computationally efficient suboptimal control design for impulsive systems based on model predictive static programming, IFAC Proceedings Volumes 47 (2014) 41–46. doi:10.3182/20140313-3-IN-3024.00172.

[33] D. F. Lawden, Optimal trajectories for space navigation, Butterworths, London, 1963.

[34] H. Robbins, Optimality of intermediate-thrust arcs of rocket trajectories, AIAA Journal 3 (1965) 1094–1098. doi:10.2514/3.3060.

[35] Y. Gao, X. Li, Optimization of low-thrust many-revolution transfers and Lyapunov-based guidance, Acta Astronautica 66 (2010) 117–129. doi:10.1016/j.actaastro.2009.05.013.

[36] R. P. Russell, Primer vector theory applied to global low-thrust trade studies, Journal of Guidance, Control, and Dynamics 30 (2007) 460–472. doi:10.2514/1.22984.

[37] K. Sachan, R. Padhi, Waypoint constrained multi-phase optimal guidance of spacecraft for soft lunar landing, Unmanned Systems 7 (2019) 83–104. doi:10.1142/S230138501950002X.