



Enhanced Smoothing Technique for Indirect Optimization of Minimum-Fuel Low-Thrust Trajectories

Ehsan Taheri,* Ilya Kolmanovsky,† and Ella Atkins‡
 University of Michigan, Ann Arbor, Michigan 48109

DOI: 10.2514/1.G000379

Indirect methods used for solving optimal-control problems, when combined with proper initialization and homotopy approaches, remain attractive for space trajectory optimization, as they are able to achieve fast convergence to a solution of the necessary conditions. In this paper, the extended logarithmic-smoothing technique is revisited and integrated with an indirect method to efficiently generate minimum-fuel time-fixed low-thrust rendezvous trajectories. This approach is considered for three cases, in which equations of motion are expressed in terms of Cartesian, spherical, and modified equinoctial coordinates. In addition, the paper addresses the calculation of the Jacobian matrix of the constraints via an implementation of the state-transition-matrix approach, which avoids the discontinuities of the control along the trajectory. The application of the method to two interplanetary missions from Earth to Mars and to asteroid Dionysus is demonstrated. It is shown that, by exploiting the state transition matrix and the homotopy method, the optimal-control problem becomes amenable to numerical treatment. The numerical results are compared in terms of the percent of converged cases, mean values for final mass, number of iterations and function evaluations, accuracy in satisfying the constraints, and computational time.

Nomenclature

$A/B/D$	=	coefficient matrices for equations of motions in modified equinoctial/spherical coordinate systems
c	=	exhaust velocity, m/s
g_0	=	gravitational acceleration at sea level, m/s ²
I_{sp}	=	specific impulse, s
m	=	mass of spacecraft, kg
r	=	radial coordinate, km
\mathbf{r}	=	position vector, km
T_{max}	=	maximum thrust value, N
t	=	time, s
\mathbf{u}	=	unit thrust vector
\mathbf{u}_s	=	$[u_r, u_\theta, u_\phi]^T$ thrust acceleration in spherical coordinate system, m/s ²
\mathbf{v}	=	velocity vector, km/s
\mathbf{x}_m	=	$[p, f, g, h, k, L]^T$ vector of modified equinoctial orbital elements
\mathbf{x}_s	=	$[r, \theta, \phi, \dot{r}, \dot{\theta}, \dot{\phi}, m]^T$ state vector in spherical coordinate system
\mathbf{Z}	=	$[\mathbf{x}^T, m, \boldsymbol{\lambda}^T]^T$ combined vector of states and costates
γ	=	thrust-acceleration conversion factor, s ² /m
δ	=	engine-throttling input
ϵ	=	continuation parameter
θ	=	angular coordinate, rad
$\boldsymbol{\lambda}$	=	costate vector
$\boldsymbol{\lambda}_c$	=	$[\lambda_x, \lambda_y, \lambda_z, \lambda_{\dot{x}}, \lambda_{\dot{y}}, \lambda_{\dot{z}}]^T$ costate vector associated with the Cartesian coordinates
$\boldsymbol{\lambda}_m$	=	$[\lambda_p, \lambda_f, \lambda_g, \lambda_h, \lambda_k, \lambda_L]^T$ vector of costates associated with modified equinoctial elements
λ_r	=	costate associated with r

$\boldsymbol{\lambda}_s$	=	$[\lambda_r, \lambda_\theta, \lambda_\phi, \lambda_{\dot{r}}, \lambda_{\dot{\theta}}, \lambda_{\dot{\phi}}]^T$ vector of costates associated with the spherical states
λ_v	=	costate associated with v
μ	=	gravitational parameter, km ³ /s ²
σ	=	sum of the squared of the residuals at the final time
Φ	=	state transition matrix
ϕ	=	elevation angle, rad

Subscripts

c	=	Cartesian
f	=	final
i	=	initial
m	=	modified equinoctial elements
s	=	spherical
T	=	target body

I. Introduction

Successful application of ion thrusters for spacecraft attitude and position control [1], along with their use in interplanetary missions, such as Deep Space 1 [2], Dawn [3], and SMART-1 [4], has provided the motivation for a new direction in trajectory optimization. The advantage of using electrically powered spacecraft propulsion stems from the increased fuel efficiency, which is due to the high value of the specific impulse [5]. The low-thrust propulsion systems are now a proven technology, and a host of future missions may use them to reduce the mission cost and increase the final payload mass. However, the engines are thrusting over a longer period of time compared to the traditional high-thrust chemical engines that operate at a lower specific impulse. The continuous profile of the thrust makes the design of optimal low-thrust trajectories challenging.

Trajectory-optimization problems can be generally solved by either direct or indirect techniques [6]. Direct methods convert the optimal-control problem into a nonlinear parameter optimization [nonlinear programming (NLP)] problem with various transcription schemes (e.g., Hermite cubic, cubic polynomial, and orthogonal function approximations) applied to either states or controls, or both states and controls. In addition, an entire class of optimal-control methods exist, which exploit a quadratic localized version of Bellman's dynamic programming, referred to as differential dynamic programming [7–12]. The related static/dynamic control approach presented in [8] has been used by NASA to fly the Dawn spacecraft to the asteroids Vesta and Ceres. The use of direct methods has grown considerably in recent years, mainly because of the growth in

Received 29 December 2015; revision received 10 May 2016; accepted for publication 17 June 2016; published online 15 September 2016. Copyright © 2016 by the American Institute of Aeronautics and Astronautics, Inc. All rights reserved. Copies of this paper may be made for personal and internal use, on condition that the copier pay the per-copy fee to the Copyright Clearance Center (CCC). All requests for copying and permission to reprint should be submitted to CCC at www.copyright.com; employ the ISSN 0731-5090 (print) or 1533-3884 (online) to initiate your request.

*Postdoctoral Research Fellow, Department of Aerospace Engineering; etaheri@umich.edu. Member AIAA.

†Professor, Department of Aerospace Engineering; ilya@umich.edu. Member AIAA.

‡Associate Professor, Department of Aerospace Engineering; ematkins@umich.edu. Associate Fellow AIAA.

computing resources and the fact that direct methods appear to have better robustness with respect to the initial guess. The development of capable NLP solvers that could use the sparsity patterns of the resulting NLP problems along with their robustness has been a major factor in popularizing the application of the direct methods.

Indirect methods, on the other hand, are based on analytically deriving the necessary conditions for optimality based on the Pontryagin's maximum principle (PMP), which usually results in a two-point boundary-value problem (TPBVP). In the indirect methods, the introduction of the Lagrange multipliers (costates) doubles the number of the differential equations that have to be propagated along the trajectory. The major difficulty in using the indirect methods, though, is the sensitivity of the solution to the initial guesses for the unknown costates. In other words, the neighborhood of the true costate values from which the solver converges to the optimal solution is often small [6]. In addition, the derivation of the Hamiltonian and first-order necessary conditions, as well as the transversality conditions, can be tedious and is problem dependent. It is helpful to use symbolic computations and complex-number-based derivatives (see [13]) to avoid error-prone hand derivations and to automate the procedure. Since, in space applications the equations of motions (EOMs) are relatively low order, the indirect methods, when combined with proper initialization and homotopy approaches, remain attractive, as they are able to achieve fast convergence to a solution of the necessary conditions. They can also be effective in receding horizon control applications, in which the problem does not change significantly between the time instants at which the solution has to be recomputed. Both of the aforementioned direct and indirect methods aim at high-fidelity solution, but may be time consuming and not suitable for evaluating thousands of trajectories in the preliminary phase of the mission design. There have been various efforts and routes taken in overcoming the difficulties associated with the design of optimal spacecraft and airplane trajectories, for instance, by resorting to heuristic and/or evolutionary techniques [14–17]. In another approach, shape-based (SB) methods for low-thrust trajectory planning have been developed [18–24] that trade optimality for speed. The solution trajectory determined by the SB methods satisfies the EOMs, boundary conditions, and even the constraint on the thrust acceleration. In addition, the solution of the SB methods can be shown to serve as a good initial guess for both direct and indirect methods [23,24].

Another route to enhance the numerical convergence of indirect methods is by generating a better estimate of the unknown costates. In [25], the costates are determined analytically by simplifying the dynamics of the original problem. The simplified dynamics are obtained by ignoring the effect of the central body (setting gravitational parameter to zero) and assuming no fuel consumption ($\dot{m} = 0$). However, the approach is limited to low-thrust trajectories of less than 1 revolution. In [26], it is demonstrated that approximate values for the initial adjoint variables can be obtained by solving equations in the neighborhood of the initial time, in which the adjoint variables can be expressed by a first-order Taylor series expansion. The most extensively used approach is based on the adjoint-control transformation [27] for estimating the costates of the TPBVP arising in low-thrust trajectory design [28–31]. In essence, the initial steering angles of the control vector and their associated time derivatives are used to compute the estimate of initial costate values of the original problem.

The continuation (homotopy) method is another approach to improving the numerical convergence of the indirect methods [32,33]. With the homotopy approach, a sequence of subproblems is solved, dependent on a continuation parameter, in which the solution of an easier subproblem becomes an initial guess for the solution of the successive harder subproblem. In principle, the differences between subproblems can be in the fidelity of the dynamic model, the cost function, and the constraints. Continuation methods are ideal for those problems, in which the solutions of the easier subproblems are readily available. However, if one of the intermediate subproblems becomes infeasible, the method fails. The sensitivity of the problem to the changes in the continuation parameter is one of the reasons that the intermediate subproblems become infeasible. Any method that is

aimed at reducing the mentioned sensitivity is advantageous, because a smaller set of the subproblems has to be solved. The extended logarithmic-smoothing method [32] has originally been proposed to address some of the challenges of low-thrust trajectory optimization via homotopy, and in [32], it has been implemented for the case in which the EOMs are formulated in the inertial Cartesian coordinate system (CS). A small step size in the continuation parameter has been taken that results in a large set of subproblems.

In this work, we consider spacecraft EOMs expressed in various CSs to improve the performance of algorithms that are aimed at solving minimum-fuel fixed-time low-thrust rendezvous optimization problems. For this purpose, the extended logarithmic-smoothing method that exploits the continuation at the cost-function level is revisited when the dynamics are modeled using the modified equinoctial orbital elements (MEOEs) [34], and using the Cartesian and spherical coordinates. In addition, we investigate if it is possible to take advantage of the homogeneity of the minimum-fuel indirect optimal-control problem with respect to the costate vector and the idea of the normalization of the costate vector in [35] to improve the computational performance. Furthermore, the accuracy of the derivatives of the terminal constraints of the resulting TPBVP with respect to the design variables is a crucial factor in improving the convergence of the numerical solvers. The state transition matrix (STM) method [36] is often used for the calculation of the sensitivities, but it can be used only if the trajectory is continuous. However, in the minimum-fuel problems, the control input is typically discontinuous, and, as detailed in [31], the presence of discontinuities poses a major difficulty to the direct application of the STM method. One way to circumvent the discontinuity is, as suggested in [31], by reconstructing a sequence of partial derivatives across the existing discontinuities and stringing them together using the chain rule. This approach, however, requires detection of changes in the sign of the so-called switching function (SF) because the discontinuities occur at zero crossings of the SF. In this paper, we demonstrate that, by adopting the regularized formulation, the constraint sensitivities with respect to the design variables can be retrieved through the STM method without dealing with the switching phenomenon at the expense of getting denser matrices that appear in the STM relations. Accurate calculation of the constraint sensitivities enables us to solve a fewer number of subproblems by changing the continuation parameter at a greater rate. Finally, we demonstrate the application of the developed method on challenging interplanetary-orbital-transfer examples. It is interesting that, by exploiting the aforementioned ideas, we are able to generate solutions reliably using general-purpose solvers in MATLAB.

The paper is organized as follows. Section II discusses the EOMs and the indirect approach to solve the minimum-fuel problem in the modified equinoctial CS. In addition, the original cost function and the modified one are explained. In Sec. III, the analytic approach of calculating the Jacobian matrix of the constraints that uses the STM method is explained. The details of implementation and sparsity patterns of the involved matrices for each CS are discussed in detail and compared. In addition, the details of implementing the complex (CX) derivative approach to be used within the STM method are explained. Section IV presents, discusses, and compares the results of applying the proposed method to two different test cases. Finally, Sec. V summarizes concluding remarks.

II. Problem Formulation

In this section, the governing EOMs are represented in the modified equinoctial CS, and the first-order necessary conditions of optimality are derived for the minimum-fuel problem. First, the original formulation of the minimum-fuel problem with a standard Mayer's form cost function is explained. Then, a modified cost function with the extended logarithmic function is introduced and the optimal control for throttling input is derived in terms of the SF. The same steps are followed for the Cartesian and spherical CSs, and are detailed in Appendices A and B.

A. Modified Equinoctial Orbital Elements

We consider an optimal-control problem for a spacecraft in an inverse-square gravitational field of a central body, in which the spacecraft is affected also by the acceleration of an onboard low-thrust propulsion system. The EOMs are expressed in the MEOEs CS, and the variation of mass is included. MEOE CS is suitable for analysis and optimization of low-thrust trajectories, as it allows the most general representation of circular, elliptic, and hyperbolic orbits without singularities at zero eccentricity and inclinations of 0 and 90 deg. Unlike the inertial Cartesian coordinates that are changing considerably over a revolution, the MEOEs are well behaved, varying slowly except for the true longitude L . We define the state vector of MEOEs (denoted by subscript m) as $\mathbf{x}_m \equiv [p, f, g, h, k, L]^T$, and their time rate of change as $\dot{\mathbf{x}}_m = f[\mathbf{x}_m(t), \mathbf{u}_m(t)]$. Let A be a mapping of $\mathbb{R}^6 \rightarrow \mathbb{R}^6 \times \mathbb{R}^3$ of the following form:

$$A(\mathbf{x}_m) = \begin{bmatrix} 0 & \frac{2p}{w} \sqrt{\frac{\mu}{p}} & 0 & 0 & 0 & 0 \\ \sqrt{\frac{\mu}{p}} \sin(L) & \sqrt{\frac{\mu}{p}} \frac{1}{w} [(w+1) \cos(L) + f] & -\sqrt{\frac{\mu}{p}} \frac{g}{w} [h \sin(L) - k \cos(L)] & \sqrt{\frac{\mu}{p}} \frac{s^2 \cos(L)}{2w} & \sqrt{\frac{\mu}{p}} \frac{s^2 \sin(L)}{2w} & \sqrt{\frac{\mu}{p}} \frac{1}{w} [h \sin(L) - k \cos(L)] \\ -\sqrt{\frac{\mu}{p}} \cos(L) & \sqrt{\frac{\mu}{p}} \frac{1}{w} [(w+1) \sin(L) + g] & \sqrt{\frac{\mu}{p}} \frac{f}{w} [h \sin(L) - k \cos(L)] & \sqrt{\frac{\mu}{p}} \frac{s^2 \cos(L)}{2w} & \sqrt{\frac{\mu}{p}} \frac{s^2 \sin(L)}{2w} & \sqrt{\frac{\mu}{p}} \frac{1}{w} [h \sin(L) - k \cos(L)] \\ 0 & 0 & 0 & \sqrt{\frac{\mu}{p}} \frac{s^2 \cos(L)}{2w} & \sqrt{\frac{\mu}{p}} \frac{s^2 \sin(L)}{2w} & \sqrt{\frac{\mu}{p}} \frac{1}{w} [h \sin(L) - k \cos(L)] \\ 0 & 0 & 0 & \sqrt{\frac{\mu}{p}} \frac{s^2 \cos(L)}{2w} & \sqrt{\frac{\mu}{p}} \frac{s^2 \sin(L)}{2w} & \sqrt{\frac{\mu}{p}} \frac{1}{w} [h \sin(L) - k \cos(L)] \\ 0 & 0 & 0 & \sqrt{\frac{\mu}{p}} \frac{s^2 \cos(L)}{2w} & \sqrt{\frac{\mu}{p}} \frac{s^2 \sin(L)}{2w} & \sqrt{\frac{\mu}{p}} \frac{1}{w} [h \sin(L) - k \cos(L)] \end{bmatrix} \quad (1)$$

and let \mathbf{b} be a mapping of $\mathbb{R}^6 \rightarrow \mathbb{R}^6 \times \mathbb{R}$ defined as

$$\mathbf{b} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \sqrt{\mu p} \left(\frac{w}{p}\right)^2 \end{bmatrix}^T \quad (2)$$

The dynamics that include the variation of mass can be written as

$$\dot{\mathbf{x}}_m = A \frac{T_{\max}}{m} \delta_m \mathbf{u}_m + \mathbf{b} \quad (3)$$

$$\dot{m} = -\frac{T_{\max}}{c} \delta_m \quad (4)$$

In these EOMs, $w = 1 + f \cos(L) + g \sin(L)$, $s^2 = 1 + h^2 + k^2$, μ is the gravitational parameter of the central body, m is the mass of the spacecraft, T_{\max} is the maximum available thrust, $0 \leq \delta \leq 1$ is the engine throttling, \mathbf{u}_m is the unit direction vector of the thrust force whose components are expressed in the local-vertical/local-horizontal orbital reference frame, and $c = I_{\text{sp}} g_0$ is the exhaust velocity. I_{sp} and g_0 are the specific impulse and the gravitational acceleration at sea level, respectively. The position and velocity vectors can be expressed in Cartesian CS in terms of the MEOEs [34].

B. Minimum-Fuel Optimal-Control Problem, Necessary Conditions, and TPBVP

In this section, we consider a minimum-fuel optimization of spacecraft rendezvous trajectory with a fixed transfer time. The cost function is expressed in Mayer's form in terms of minimizing the negative value of the mass remaining at the final time, that is

$$J = -m(t_f) \quad (5)$$

The control inputs being optimized are \mathbf{u}_m and δ_m . We proceed by forming the Hamiltonian

$$\begin{aligned} H &= \lambda_m^T f[\mathbf{x}_m(t), \mathbf{u}_m(t)] + \lambda_m \left(-\frac{T_{\max}}{c} \delta_m \right) \\ &= \lambda_m^T \left(A \frac{T_{\max}}{m} \delta_m \mathbf{u}_m + \mathbf{b} \right) - \lambda_m \left(\frac{T_{\max}}{c} \delta_m \right) \end{aligned} \quad (6)$$

in which $\lambda_m = [\lambda_p, \lambda_f, \lambda_g, \lambda_h, \lambda_k, \lambda_L]^T$ is the costate vector associated to the MEOEs, and λ_m is the costate associated with the mass. By expanding the terms in the Hamiltonian with the help of Eqs. (1) and (2), and rearranging the terms, we can express it as

$$H = \frac{T_{\max}}{m} \delta_m \lambda_m^T A \mathbf{u}_m + \lambda_m^T \mathbf{b} - \lambda_m \frac{T_{\max}}{c} \delta_m \quad (7)$$

Note that the Hamiltonian is a linear function of both the unit thrust direction vector \mathbf{u}_m and the throttling magnitude δ_m . Therefore, the weak form of the PMP is used to characterize the optimal thrust vector:

$$\mathbf{u}_m^* \in \arg \min_{\|\mathbf{u}_m\|=1} H \quad (8)$$

which corresponds to minimizing the following term in the Hamiltonian

$$\lambda_m^T A(\mathbf{x}) \mathbf{u}_m \rightarrow \min \quad (9)$$

The Hamiltonian is minimized when

$$\mathbf{u}_m^* = -\frac{A^T \lambda_m}{\|A^T \lambda_m\|} \quad (10)$$

Exploiting the weak form of the PMP, we can also characterize the optimal throttling-magnitude input as

$$\delta_m^* \in \arg \min_{0 \leq \delta_m \leq 1} H \quad (11)$$

From Eq. (7), to determine δ_m , we need to minimize the term:

$$-T_{\max} \left(\frac{\|A^T \lambda_m\|}{m} + \frac{\lambda_m}{c} \right) \delta_m \rightarrow \min \quad (12)$$

We can simplify our analysis by defining a SF to represent the term inside the parentheses as

$$\text{SF}_m = \frac{\|A^T \lambda_m\|}{m} + \frac{\lambda_m}{c} \quad (13)$$

Singular arcs exist for cases in which $\text{SF}_m = 0$ over a finite time interval, but they rarely occur in space trajectories (see Ref. [31]). Therefore, assuming there are no singular arcs in the solution, the optimal value of the throttling input, δ_m^* , depends on the sign of the SF_m :

$$\delta_m^* = \begin{cases} 1, & \text{if } SF_m > 0 \\ 0, & \text{if } SF_m < 0 \end{cases} \quad (14)$$

$$\dot{\lambda}_m = - \left[\frac{\partial H}{\partial x} \right]^T \quad (19)$$

Note that the ‘‘bang-bang’’ behavior of δ_m^* per Eq. (14) complicates the standard integration of the differential equations and subsequent numerical optimization due to nonsmoothness. This problem can be avoided by resorting to a continuation (homotopy) approach introduced at the cost-function level. Bertrand and Epenoy [32] describe several approaches to perturb the cost function. In this paper, the cost function is rewritten in terms of the consumed propellant mass and is modified by introducing a combination of logarithmic functions. Logarithmic functions can be used to provide a smooth approximation of the throttling profile, as demonstrated in detail in [32]. If the extended logarithmic function is used, the final Lagrange form of the optimal-control problem is written as

$$J = \frac{T_{\max}}{c} \int_{t_i}^{t_f} \{ \delta_m - \epsilon [-\delta_m \log(\delta_m) - (1 - \delta_m) \log(1 - \delta_m)] \} dt \quad (15)$$

in which $\epsilon = 0$ corresponds to the original problem. Thus, the new Hamiltonian is written as

$$H = \frac{T_{\max}}{c} \{ \delta_m - \epsilon [-\delta_m \log(\delta_m) - (1 - \delta_m) \log(1 - \delta_m)] \} + \frac{T_{\max}}{m} \delta_m \lambda_m^T \mathbf{A}(\mathbf{x}) \mathbf{u}_m + \lambda_m^T \mathbf{b} - \lambda_m \left(\frac{T_{\max}}{c} \delta_m \right) \quad (16)$$

The optimal unit vector \mathbf{u}_m^* is still obtained through Eq. (10), but the SF of the new problem takes a different and more complicated form, as the Hamiltonian is not linear with respect to the throttling parameter. Using the strong form of PMP, we now obtain the optimal value of the throttling input as

$$\frac{\partial H}{\partial \delta_m} = 0 \rightarrow - \frac{T_{\max}}{c} \left[\epsilon \log \left(\frac{1 - \delta_m}{\delta_m} \right) - 1 \right] - T_{\max} \cdot SF_m = 0 \quad (17)$$

Equation (17) is solved to obtain the optimal value of the throttling parameter:

$$\delta_m^* = \frac{1}{1 + 10^{\frac{1 - SF_m \epsilon}{c}}} \quad (18)$$

which contains the SF of the unperturbed cost function, Eq. (13), as one of its terms. We note that, in the derivation of Eq. (18), we have canceled the T_{\max} terms in both terms of Eq. (17). For the second term, however, if any thrust-acceleration conversion factor is used in the numerical implementation, the same factor should be included in the SF. For instance, if the canonical units are used in the numerical simulation to scale the problem, it is necessary to convert the acceleration terms from the SI units of meter per square second to distance unit (DU) per square time unit (TU) using a conversion factor of the following form:

$$\gamma = \frac{TU^2}{DU \times 1000}$$

in which DU and TU denote the values for the distance and time units, respectively. In other words, in the first term of the SF, Eq. (13), the mass is in the denominator, which means that the term $\| \mathbf{A}^T \lambda_x \|^2$ is multiplied by acceleration and requires suitable conversion [see Eq. (12)]. Therefore, in the numerical simulations, the SF is computed using the following expression:

$$SF_m = \frac{\gamma \| \mathbf{A}^T \lambda_m \|^2}{m} + \frac{\lambda_m}{c}$$

in which γ is the thrust-acceleration conversion factor. We next derive the costates (adjoint) differential equations through

These equations are derived using the Symbolic toolbox of MATLAB. It turns out the mass costate differential equation is easy to derive. It has the following form:

$$\dot{\lambda}_m = - \frac{\partial H}{\partial m} = - \frac{T_{\max}}{m^2} \delta \| \mathbf{A}^T \lambda_m \|^2 \quad (20)$$

For a rendezvous problem, the boundary conditions can be written in the form of six equality constraints:

$$g[\mathbf{x}(t_f), t_f] = \begin{bmatrix} \mathbf{r}(t_f) - \mathbf{r}_T \\ \mathbf{v}(t_f) - \mathbf{v}_T \end{bmatrix} = \mathbf{0} \quad (21)$$

in which \mathbf{r}_T and \mathbf{v}_T are the target position and velocity vectors, respectively. Finally, the transversality conditions can be written as

$$\lambda_x(t_f) = \sum_{i=1}^{j=6} \alpha_i \frac{\partial g_i}{\partial \mathbf{x}} [\mathbf{x}(t_f), t_f] \quad (22)$$

However, we can only obtain the condition on the final value of the costate associated with the mass through the transversality condition:

$$\lambda_m(t_f) = 0 \quad (23)$$

The right-hand side of the costate differential equation associated with the mass state, Eq. (20), is always nonpositive, $\dot{\lambda}_m \leq 0$, and from the transversality conditions, $\lambda_m(t_f) = 0$, it follows that $\lambda_m(t_i) \geq 0$. Therefore, the state differential equations, Eq. (3); the costate differential equations, Eq. (19); along with the terminal constraints, Eqs. (21) and (23), constitute the TPBVP for the minimum-fuel problem. Any solution scheme developed for solving this problem requires an initial guess for the unknown initial costate values, $\lambda = [\lambda_m^T, \lambda_x^T]^T$. In a single-shooting scheme, the problem can be written as

$$\boldsymbol{\psi}[\lambda(t_i)] = \{ [\mathbf{r}(t_f) - \mathbf{r}_T]^T, [\mathbf{v}(t_f) - \mathbf{v}_T]^T, \lambda_m(t_f) \}^T = \mathbf{0} \quad (24)$$

in which $\lambda(t_i)$ denotes the vector of unknown initial costates.

The regularization approach, which is used in this section, is aimed at smoothing the problem, so that typical ordinary differential integrators can be used. By employing the homotopy approach, the continuation parameter ϵ is decreased to a sufficiently small but nonzero value. In our numerical experiments, this limiting value of $\epsilon = 1.0 \times 10^{-5}$ is sufficient for generating nearly perfect bang-bang thrust profiles.

The derivations of minimum-fuel relations in the Cartesian and spherical CSs for original costate vector are presented in Appendices A and B, respectively. Table 1 summarizes the optimal thrust unit vectors and throttling inputs corresponding to the Cartesian, modified equinoctial, and spherical CSs. Initially, we also considered the normalization strategy of the costate vector proposed in [35], and tried to use the transformation therein, taking advantages of the fact that the initial values of the costates are confined to lie on a unit eight-dimensional (8-D) hypersphere. The advantage is that the design variables are bounded (see [35]). However, the performance of the 8-D hypersphere normalized costate vector was not comparable to the

Table 1 Summary of the optimal thrust of Cartesian, MEOE, and spherical CSs

CS	Optimal unit vector \mathbf{u}^*	Optimal throttling input δ^*
Cartesian	$-\frac{\lambda_v}{\lambda_m}$	$\frac{1}{1 + 10^{\frac{1 - SF_m \epsilon}{c}}}$
MEOE	$-\frac{\mathbf{A}^T \lambda_m}{\ \mathbf{A}^T \lambda_m \ ^2}$	$\frac{1}{1 + 10^{\frac{1 - SF_m \epsilon}{c}}}$
Spherical	$-\frac{D \lambda_j \partial \phi}{\ D \lambda_j \partial \phi \ ^2}$	$\frac{1}{1 + 10^{\frac{1 - SF_m \epsilon}{c}}}$

other treatments of the initial costate vector, and for this reason, it is not included in the results.

C. Fixing Initial Value of the Mass Costate

Another common approach in dealing with initial values of the costates is to fix the value of one of the initial costates; this is equivalent to reducing the dimension of the unknown costate vector by 1. We fixed the value of the initial mass costate, $\lambda_m(t_i)$, to be equal to 1. Unlike the 8-D hypersphere normalization approach of the previous section, the latter approach does not bound the initial value of the costates, but in practice, it does help to scale the problem. For this reason, we include the results of this method in our comparison, denoted as “scaled-costate-vector” case.

III. Derivative Calculation

The highly nonlinear shooting problem defined in Eq. (24) is usually treated by using solvers that exploit Newton’s or quasi-Newton’s updates [37] that can be enhanced by using accurate values of the first- and second-order derivatives of the terminal constraints, Eq. (24), with respect to the design variables. The simplest method of calculating the derivatives is to use finite difference (FD) approach, which is also the least accurate [38,39]. Analytic derivatives are more accurate however, have limited applicability for two major reasons. The first issue is that, for complicated dynamics, the partial derivatives turn out to be long expressions, and that their calculation takes considerable time and ruins the speed advantage. In addition, for any new set of dynamics and constraints, the construction of the derivatives has to be repeated, which is a shortcoming from the perspective of the general-purpose programming. Another approach toward calculating the constraint derivatives is to use the so-called STM method through which the partial derivatives are mapped from one time instant to another time instant on a continuous trajectory [36]. The STM Φ comprises partial derivatives of components of final state vector with respect to components of the initial state vector:

$$\Phi(t, t_i) = \frac{\partial \mathbf{Z}(t)}{\partial \mathbf{Z}(t_i)} \quad (25)$$

and is obtained, in our problem, by integrating the following variational equation:

$$\dot{\Phi}(t, t_i) = \left[\frac{\partial \mathbf{F}}{\partial \mathbf{Z}} \right]_t \Phi(t, t_i), \Phi(t_i, t_i) = I_{14 \times 14} \quad (26)$$

in which $\dot{\mathbf{Z}} = \mathbf{F}(\mathbf{Z})$ is the complete set of state–costate differential equations, and $\mathbf{F} = [\dot{\mathbf{x}}_m^T, \dot{m}, \dot{\lambda}^T]^T$, in which $\mathbf{Z} = [\mathbf{x}_m^T, m, \lambda^T]^T$. Consequently, the dimension of the problem is increased by n^2 , in which n is the dimension of the involved states and costates, which, in our case, corresponds to $n = 14$. Calculation of the term $\partial \mathbf{F} / \partial \mathbf{Z}$ is the most important part in the STM method, along with the fact that we now have to propagate $n(n + 1) = 210$ differential equations. Note that the continuity of the differential equations is a necessary condition if we want to implement the STM method. The sparsity patterns of the matrix term, $\partial \mathbf{F} / \partial \mathbf{Z}$, for the Cartesian, spherical, and MEOE CSs are shown in Fig. 1 for the original costate vector. Note

that the sparsity pattern of the Cartesian representation is different from the one derived in [31], because the relation for the optimal throttling input δ^* is substituted into \mathbf{F} . As can be seen, the matrices are getting denser by having 62, 89, and 196 nonzero elements out of 196 elements in the Cartesian, spherical, and MEOE representations of the problem, respectively. The STM method is explained in [31] with a procedure to handle the discontinuities along a possible minimum-fuel trajectory that usually includes several legs of thrust and coast. However, through regularization of the cost function, our optimal-control input is always continuous, which avoids the former shortcoming at the price of increasing the complexity of the relations that appear in the term $\partial \mathbf{F} / \partial \mathbf{Z}$. This complexity is due to the fact that δ^* , in our formulation, becomes a function of the SF (see Table 1). The SF itself is a function of the states and costates [e.g., see Eq. (13)], and when δ^* is plugged into the differential equations, the relations become more complicated. In addition to the lengthy equations, for double-precision numbers, the smallest nonzero number that can be represented is 10^{-308} , and in our numerical implementation, floating-point underflow has been encountered at $\epsilon < 0.01$; this issue has to be handled by proper safeguarding. We found out that, by using conditional control statements over one of the intermediate variables (created by MATLAB function-generator toolbox), the issue is resolved without losing accuracy. In fact, any instances, in which a base value of 10 is raised to a certain power, need careful attention. Both the optimal throttling input δ^* and its derivative with respect to \mathbf{Z} result in such instances. The derivative of δ^* can be written as

$$\frac{\partial \delta^*}{\partial \mathbf{Z}} = -\delta^* (1 - \delta^*) L_n(10) \frac{\partial \eta}{\partial \mathbf{Z}}$$

in which $\eta = (1 - \text{SF} \times c) / \epsilon$, and L_n is the natural logarithm. The derivative of δ^* itself can be expressed in terms of δ^* . For all of CSs, we have been able to use the MATLAB symbolic toolbox to derive the STM partial derivatives symbolically, and we used the MATLAB automatic function generator to generate optimized codes. For the MEOE CS, we also considered using the CX derivative method [38] to calculate numerically the required partial derivatives. In essence, we still use the STM method; however, the evaluation of the complicated term $\partial \mathbf{F} / \partial \mathbf{Z}$ along the trajectory is achieved through the CX derivative approach. There are 14 elements in the design vector \mathbf{Z} , and at each time instant during the integration, we have to compute the term $\partial \mathbf{F} / \partial \mathbf{Z}$ by applying a small perturbation step to the imaginary direction of the desired independent element of the design vector (i.e., $z = z + i\alpha$, in which α is the size of the perturbation). For instance, the first derivative with respect to the first element of the \mathbf{Z} vector is computed through

$$\frac{\partial \mathbf{F}}{\partial z_1} \approx \frac{\text{Im}[F(z_1 + i\alpha, z_2, \dots, z_{14})]}{\alpha}$$

and constitutes the first column of the $\partial \mathbf{F} / \partial \mathbf{Z}$ term. This procedure has to be repeated for the other 13 elements to construct the full 14-by-14 matrix $\partial \mathbf{F} / \partial \mathbf{Z}$. The numerical results of using this method are explained in the next section.

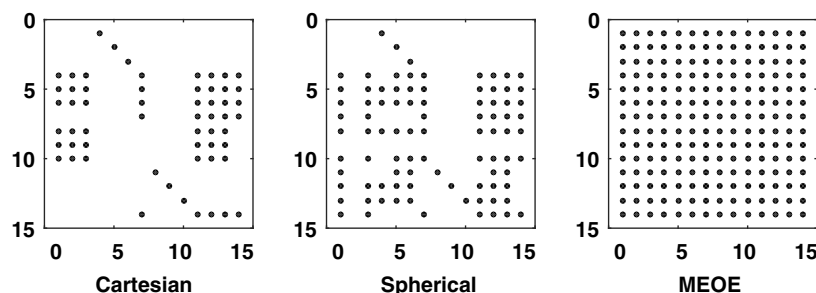


Fig. 1 Sparsity pattern of the term $\partial \mathbf{F} / \partial \mathbf{Z}$ in the STM method for various CSs.

IV. Results

In this section, the results of applying the extended logarithmic-smoothing technique to two time-fixed rendezvous optimal-control problems formulated in Cartesian, spherical, and MEOE CSs are explained. A grid of CSs has been considered, so that a reasonable comparison between the performances of various methods could be made. For the treatment of costate vector, we have considered the original as well as the scaled-costate versions. In addition, for each case, we compared the calculation of the sensitivities using the FD and STM methods. This corresponds to four cases for each CS and constitutes a set of 12 cases for each problem. Each row in the provided tables corresponds to the results of 50 randomly generated initial costates, and the numerical results summarize quantitative values, like the percent of converged cases, mean errors in the terminal position and velocity vectors, mean final mass, mean value for the number of iterations, mean value for the number of function evaluations, and mean simulation time. As specific examples, the Earth-to-Mars and Earth-to-Dionysus interplanetary rendezvous maneuvers are considered, in which the canonical units are used, such that 1 DU is equal to the astronomical unit (AU), and $2 \times \pi$ time unit (TU) is 1 year. In the numerical simulations, we used $\mu = 132,712,440,018 \text{ km}^3/\text{s}^2$, $g_0 = 9.8065 \text{ m/s}^2$, and $\gamma = \text{TU}^2/(1000 \times \text{DU}) = 168.6365 \text{ s}^2/\text{m}$. In addition, Table 2 summarizes the parameters of the spacecraft for each problem we used in the numerical simulations. The MATLAB `fsolve` function is used as the solver exploits the trust-region dogleg algorithm and is implemented similar to the method in [40]. Table 3 summarizes the settings used for each problem in the numerical simulations, in which TolX is the termination tolerance on the search step size. According to the description of the function, the tolerance called TolFun relates to both the size of the latest change in the sum of the squares of the residuals $\sigma = \Psi^T \Psi$ [see Eq. (24)] and the relative norm of the gradient of this sum of squares, respectively. The relative norm of the gradient is denoted as the “first-order-optimality” metric, and its value is used for determining the stopping condition. For instance, Table 4 shows the last two iterations of `fsolve` for the Earth-to-Dionysus problem solved in MEOE CS and using the CX derivative method. It is in the last iteration that the value of the first-order optimality, 8.32×10^{-10} , is less than the defined function tolerance TolFun = 1.0×10^{-7} . In addition to the first-order-optimality criterion, the solver checks a secondary criterion that the sum of the squared of the residuals σ to be lower than $\sqrt{\text{TolFun}}$. The satisfaction of the aforementioned criteria terminates the search, and the solver returns a feasibility flag. Note that the rate of convergence of the

Table 2 Spacecraft parameters used for each problem

Problem	m_0 , kg	T_{\max} , N	I_{sp} , s
Earth to Mars	1000	0.5	2000
Earth to Dionysus	4000	0.32	3000

Table 3 Parameter settings for the MATLAB `fsolve` solver

Problem	Maximum number		TolX	TolFun
	Iterations	Function calls		
Earth to Mars	300	300	1.0×10^{-14}	1.0×10^{-10}
Earth to Dionysus	1000	1000	1.0×10^{-14}	1.0×10^{-7}

Table 4 Last two iterations of MATLAB `fsolve` function for the problem of Earth to Dionysus with TolFun = 1.0×10^{-7}

Iteration number	σ	First-order optimality
90	3.80738×10^{-11}	7.05×10^{-5}
91	5.25934×10^{-21}	8.32×10^{-10}

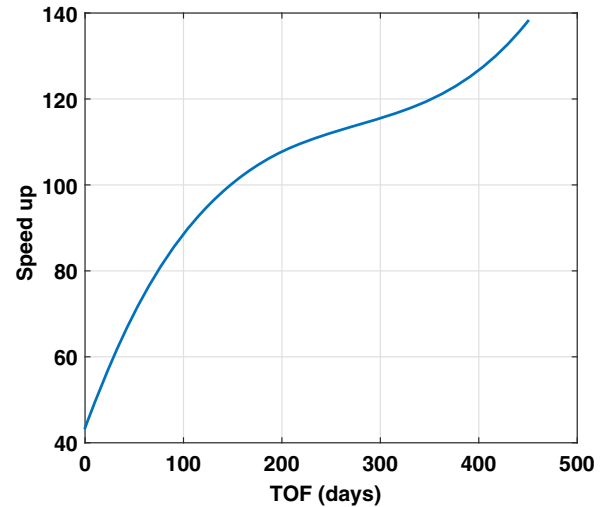


Fig. 2 Speedup factor using the compiled code vs. different values for the TOF.

gradient-based methods is quadratic [37], which is reflected in the accuracy of the values from iteration 90 to iteration 91. Note that, for the rendezvous problem of Earth to Dionysus, we relaxed the tolerance of the function (see Table 3) because its time of flight (TOF) of 9.6 years is considerably larger than that of the Earth-to-Mars problem. Each execution corresponds to solving a set of six subproblems in the homotopy approach used for solving Eq. (24) that starts from $\epsilon = 1$ down to $\epsilon = 1.0 \times 10^{-5}$ using 0.1 as the multiplier factor. The governing EOMs consist of 14 first-order differential equations that have to be propagated; but, when STM is used to calculate the sensitivity of the constraints with respect to the design variables, the number of differential equations increases to 210. To speed up the solution procedure, a compiled code based on a modified version of MATLAB `ode45` function is used, which is based on a variable step explicit Runge–Kutta formula, with absolute and relative tolerances set to 1.0×10^{-10} . The modified version is based on the MATLAB `ode45` code, except that the intermediate variables are removed and no longer are stored because the final values of states and costates are of our interest. Figure 2 shows the speedup factor for propagating the set of seven MEOE differential equations using a tangential thrust of 0.5 N acting on a spacecraft departing from Earth with $m_0 = 1000 \text{ kg}$ vs different values for the TOF. For instance, for a flight time of 350 days, the propagation is performed approximately 120 times faster. All of the numerical computations have been performed on an Intel Xeon 3.1 GHz PC with Windows XP and 4 GB of RAM. The reported computation times are calculated by using the MATLAB `tic-toc` command.

A. Earth-to-Mars Problem

In this section, the results of applying the extended logarithmic method to an Earth-to-Mars rendezvous problem are explained and compared. The boundary conditions are taken from [11], in which the position and velocity vectors of the initial and target bodies are

$$\mathbf{r}_i = [-140,699,693; -51,614,428; 980]^T \text{ (km)}$$

$$\mathbf{v}_i = [9.774596, -28.07828, 4.337725 \times 10^{-4}]^T \text{ (km/s)}$$

$$\mathbf{r}_T = [-172,682,023; 176,959,469; 7,948,912]^T \text{ (km)}$$

$$\mathbf{v}_T = [-16.427384, -14.860506, 9.21486 \times 10^{-2}]^T \text{ (km/s)}$$

and the TOF is 348.795 days. Table 5 summarizes the numerical results for various CSs and using different treatments of the costate vector along with two methods to calculate the sensitivities of the constraints with respect to the design variables (initial values of the costates). The set of initial costates is generated in the range of [0, 100] using the MATLAB `rand` function. The formulation of the

Table 5 Comparison of methods for the problem of Earth to Mars

CS	Costate treatment	Sensitivity calculation	Percent of converged cases	Δr_f , km	$\Delta v_f \times 10^{-8}$ km/s,	m_f , kg	Number of iterations	Number of function calls	Simulation time, s
Cartesian	Original	FD	64	0.494	6.638	603.93	52	412	0.766
		STM	100	0.439	4.74	603.93	42	65	1.3
	Scaled	FD	66	0.518	6.43	603.93	50	403	0.752
		STM	100	0.847	9.72	603.93	137	143	2.431
Spherical	Original	FD	96	0.49	39.9	603.93	53	448	1.483
		STM	98	0.143	1.86	603.93	83	89	3.311
	Scaled	FD	84	0.3	3.26	603.93	56	475	1.185
		STM	90	0.131	1.62	603.93	66	72	2.6
MEOE	Original	FD	96	0.194	2.13	603.93	44	389	0.641
		STM	100	0.096	0.093	603.93	43	49	4.62
	Scaled	FD	94	0.214	2.753	603.93	49	423	0.677
		STM	100	0.0977	0.95	603.93	55	61	5.0792

problem in various CSs results in the same final mass of $m_f = 603.93$ kg. The 64% of converged cases in the Cartesian CS is the lowest in comparison with the spherical and MEOE CSs. However, the effect of using accurate sensitivities is evident with an improvement in the percent of converged cases to 100%. The mean values for the final error in position and velocity vectors, Δr_f and Δv_f , are approximately the same. In all of the cases, it has been observed that the error in the final velocity vector is orders of magnitude smaller than the error in the final position vector. The mean value of the number of iterations and function evaluations has been decreased considerably when the STM method is used. On the other hand, the mean value of the simulation time for each converged simulation is approximately two times slower with the STM compared to the FD method. The scaled-costate vector treatment has the same statistical performance, except for an increase in the number of iterations and a slightly higher simulation time. Modeling the problem in the spherical CS has improved the percent of converged cases to 96% even when FD is used. The simulation time in the spherical CS case is higher than in the Cartesian CS case due to the nonlinearities in the EOMs. Again, the percent of converged cases has increased by using STM; but, it is not a significant improvement. Using the STM method, the overall number of iterations and function evaluations is reduced, whereas the simulation time is again greater than with the FD. The scaled-costate vector treatment has the quantitative performance of the original costate vector. Using the FD method, the percent of converged cases of the MEOE CS is already high with an improvement in the errors in the final position and velocity vectors, whereas the mean simulation time is slightly lower than the mean simulation time in the Cartesian CS case. With the

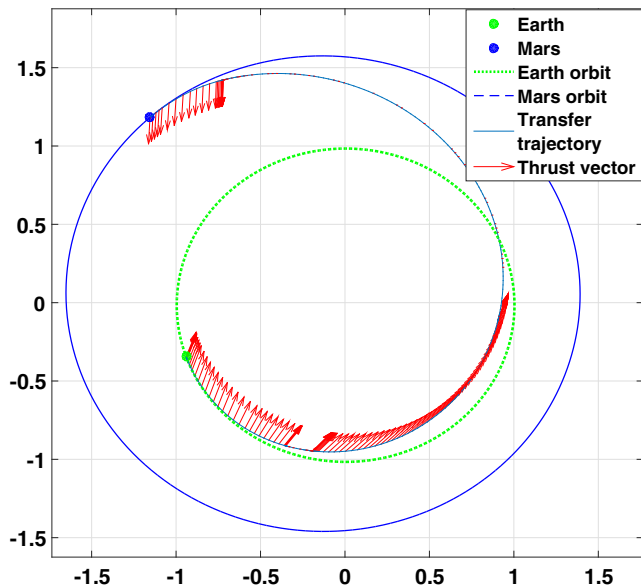


Fig. 3 Earth-to-Mars optimal trajectory in the inertial Cartesian CS.

Table 6 Keplerian orbital elements of asteroid Dionysus with respect to the sun

Dionysus	
a , AU	2.2
e	0.542
i , deg	13.6
Ω , deg	82.2
ω , deg	204.2
M , deg	114.4232
Epoch, MJD	53,400

STM method, the percent of converged cases is 100% and the final error in position vector is the best among all of the CSs. However, the STM method is approximately eight times slower than the FD method. For the MEOE CS, the computational efficiency of the algorithm degrades considerably due to the resulting lengthy symbolic relations. It is possible to further improve the code, which is generated by the MATLAB function-generator toolbox, for instance, by introducing intermediate variables to avoid redundant calculations; but, it is still may be slower than the FD method. As it is expected, for the cases that the solver is provided with analytic partial derivatives through the STM method, the solver converges in considerably fewer number of overall iterations and function evaluations because the accuracy of the derivatives has been improved.

Figure 3 shows the optimal trajectory for the Earth-to-Mars heliocentric orbital rendezvous problem, which is expressed in the inertial Cartesian CS. The spacecraft performs a maneuver that

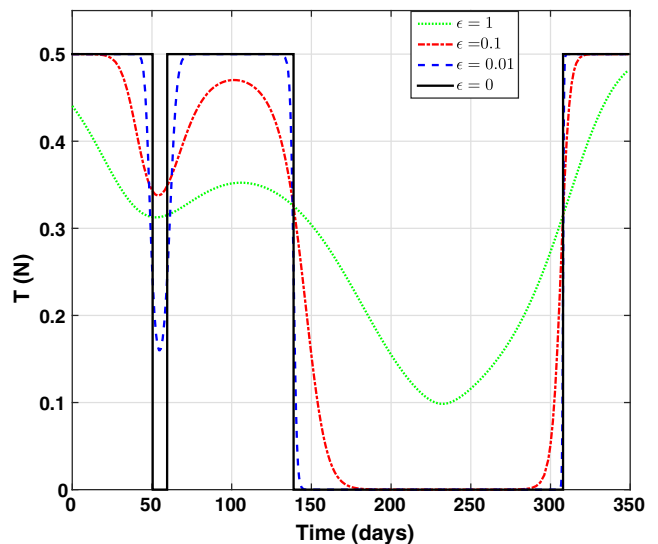


Fig. 4 Earth-to-Mars optimal thrust profile for different continuation parameters.

Table 7 Comparison of different modeling and sensitivity methods for the problem of Earth to Dionysus

CS	Costate treatment	Sensitivity calculation	Percent of converged cases	Δr_f , km	$\Delta v_f \times 10^{-8}$ km/s	m_f , kg	Number of iterations	Number of function calls	Simulation time, s
Cartesian	Original	FD	N/C	—	—	—	—	—	—
		STM	88	0.007	0.03	2403.94	120	126	11.4
	Scaled	FD	2	994.06	2721	2718.32	191	1002	6.9
Spherical	Original	STM	100	0.0056	0.02	2715.52	76	70	7.1
		FD	16	1232.4	3358	2718.32	163	941	5.7
	Scaled	STM	100	0.0006	0.002	2718.32	61	67	7.4
MEOE	Original	FD	N/C	—	—	—	—	—	—
		STM	100	0.0007	0.003	2718.32	60	66	9.47
	Scaled	FD	78	4.02	15.8	2718.32	62	532	3.7
	Original	STM	100	0.1	0.4	2718.32	67	73	19.5
		FD	74	2.66	10.4	2718.32	65	550	3.5
	Scaled	STM	100	0.1	0.4	2718.32	68	74	19.9

contains three thrust arcs. Figure 4 depicts the optimal profile of the thrust value for different continuation parameters. It shows that, by selecting a small value of the continuation parameter ϵ , the perfect bang-bang thrust-value profile is obtained using the extended logarithmic continuation. The spacecraft is on a long ballistic flight that takes about 150 days before the terminal activation of thrust occurs that sets the spacecraft on a suitable path for rendezvous. The switching between the engine on-and-off throttling positions is an attribute of minimum-fuel trajectories, which exploits the dynamics and avoids the excessive use of onboard propellant. It is important to note that the Earth-to-Mars problem is a relatively simple transfer, as the initial and final orbits have small values of eccentricity and inclination. For the problems with greater number of revolutions and greater change in the orbital elements, the difference between the cost functions and various models is expected to be greater, which is addressed in the next section.

B. Earth to Asteroid Dionysus

In this section, we try to solve a more difficult problem, in which the differences between the initial and final orbital elements are larger in comparison with the Earth-to-Mars problem. The orbital elements of asteroid Dionysus are taken from the Jet Propulsion Laboratory Solar System Dynamics[§] and can be found in Table 6, in which the epoch date is given as the Modified Julian Date (MJD). The target body is asteroid Dionysus with eccentricity and inclination values of 0.542 and 13.54 deg, respectively. The asteroid makes many close approaches to Earth, which makes it an appealing target for future missions. The spacecraft departs from Earth on 23 December 2012 and the mission takes 3534 days. Any low-thrust trajectory from Earth to the asteroid Dionysus requires considerable change in eccentricity and inclination values over several revolutions around the sun. Such kind of transfers, in fact, represents quite a tricky problem for a direct method. The large number of revolutions increases the dimension of the NLP problem, whereas the eccentricity of the orbit can cause similar problems. The Earth position and velocity vectors at the departure are

$$\mathbf{r}_i = [-3,637,871.081; 147,099,798.784; -2261.441]^T \text{ (km)}$$

$$\mathbf{v}_i = [-30.265097, -0.8486854, 0.0000505]^T \text{ (km/s)}$$

Table 7 summarizes the results of using various CSs and different sensitivity-calculation methods for the Earth-to-Dionysus problem. The set of initial costates is generated in the range of [0, 1] using the MATLAB rand function. The impact of the CS and the existence of accurate sensitivity information are noticeable in this problem. The Cartesian CS with the FD method and the original costate vector results in No Convergence (N/C). The performance of the scaled-costate treatment is not different with a negligible 2% of converged cases. The final position error is about 1000 km, which is a large value. Note that the final velocity error is also three orders of magnitudes

greater than in the previous cases. However, these numbers still satisfy the solver criteria, which are imposed in the canonical units. The simulation time of the scaled-costate treatment method is 6.9 s, which is greater than that of the Earth-to-Mars problem, which is due to the longer TOF in this problem. The mean value of the number of function evaluations is also large. However, when the STM method is used, the percent of converged cases changes significantly for both the original and scaled-costate versions to 88 and 100%, respectively. The final error in position and velocity is significantly lower. The increase in the simulation time (11.4 s) is due to the fact that there are several local solutions and the standard deviation of the simulation time is around 4 s. The solutions of the Cartesian CS with the STM and the original costate vector are explained in detail in the next paragraph. For the scaled-costate treatment method in the Cartesian CS, it seems that fewer numbers of iterations and function evaluations are required compared to the original costate vector. Likewise, the simulation time in the Cartesian CS with the scaled-costate treatment method and when STM is used is higher due to the greater number of iterations and function evaluations. For the spherical CS, the original costate vector with the FD method results in slightly better performance in the percent of the converged cases. Note that the simulation time when the spherical CS with the FD method is used is lower than that of the Cartesian with FD because of the fewer number of iterations and function evaluations. When the STM method is used, the percent of converged cases is 100% for both the original and scaled-costate treatment methods with a slight increase in the simulation time. The position and velocity errors with the STM are significantly lower than those with the FD method. The overall mean number of iterations and function evaluations are reduced. The MEOE CS with the FD method of both the original and scaled-costate treatment methods has the best convergence performance with 78 and 74%, respectively, of converged cases. The final position error is significantly lower than the other CSs. It is interesting that the simulation time for the MEOE CS with the FD method is lower than with the other CSs because of the lower number of iterations and function evaluations; these results suggest the suitability of the MEOEs for problems with many revolutions. The simulation time with the STM is the greatest among all of the methods, which is due to the lengthy expressions of the STM matrix. It is important to note that the final mass is the same for all cases except for two cases. The first one occurs in the Cartesian CS with the original costate vector, and the second one occurs in the Cartesian CS with the scaled-costate vector. In these two cases, the mean values of the final mass are not close to the optimal value. For all of the other cases, the standard deviation in the final mass was negligible, and we did not mention them; but, the standard deviations in the final mass of the Cartesian CS specifically for the original and scaled-costate vectors are approximately 276.68 and 11.2 kg, respectively. In fact, for the Cartesian CS with the STM and original costate vector, there are 43 converged cases out of the total 50 cases, and all of the converged cases satisfy the defined tolerances. All of the 43 solutions except two of them (which have significantly lower values for the final masses with 1531 and 1743 kg) are plotted in Fig. 5. There are five major levels of local solutions with different values for the final mass. Only

[§]Available online at <http://ssd.jpl.nasa.gov> [retrieved 23 December 2015].

six of the solutions belong to the highest level with a final mass of $m_f = 2718.33$ kg. Figures 6a–6d show the representative solutions of the lowest level up to the level with $m_f = 2672$ kg, respectively. The thrust arcs are getting closer to the optimal profile, in which the engine is active for a certain period of the perihelion passage of the intermediate elliptical orbits and is off for the remainder of time until the next perihelion passage. This pattern goes on for four revolutions with a gradual increase in the lengths of the active thrust arcs until the last revolution. The last revolution consists of two separate thrust arcs,

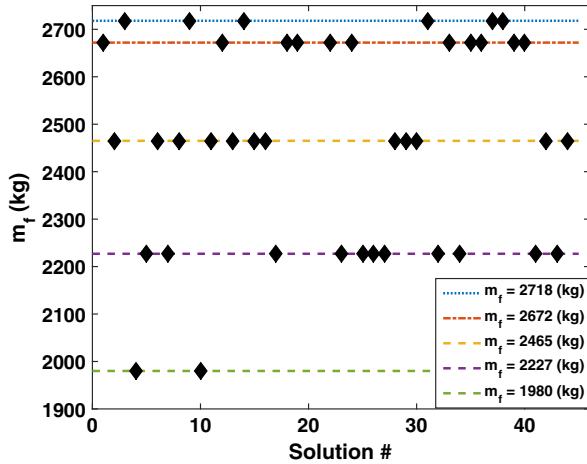


Fig. 5 Earth-to-Dionysus solution levels using the Cartesian CS with STM.

one during the perihelion passage followed immediately by a coast arc, and the second one to bring the spacecraft within the same inclination of the Dionysus orbit. The final thrust arc occurs at the farthest distance with respect to the sun and the lowest magnitude of velocity to make the plane change maneuver less expensive. Note that the spacecraft approaches the target asteroid on a zero-thrust (ballistic) arc. For the

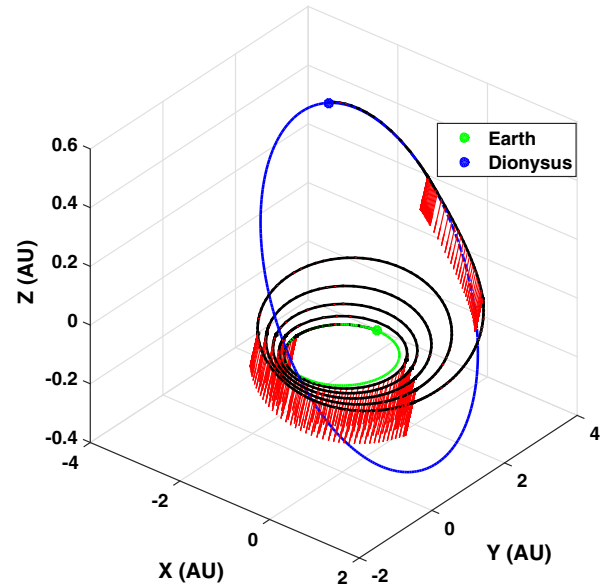


Fig. 7 Earth-to-Dionysus optimal trajectory with $m_f = 2718.32$ kg.

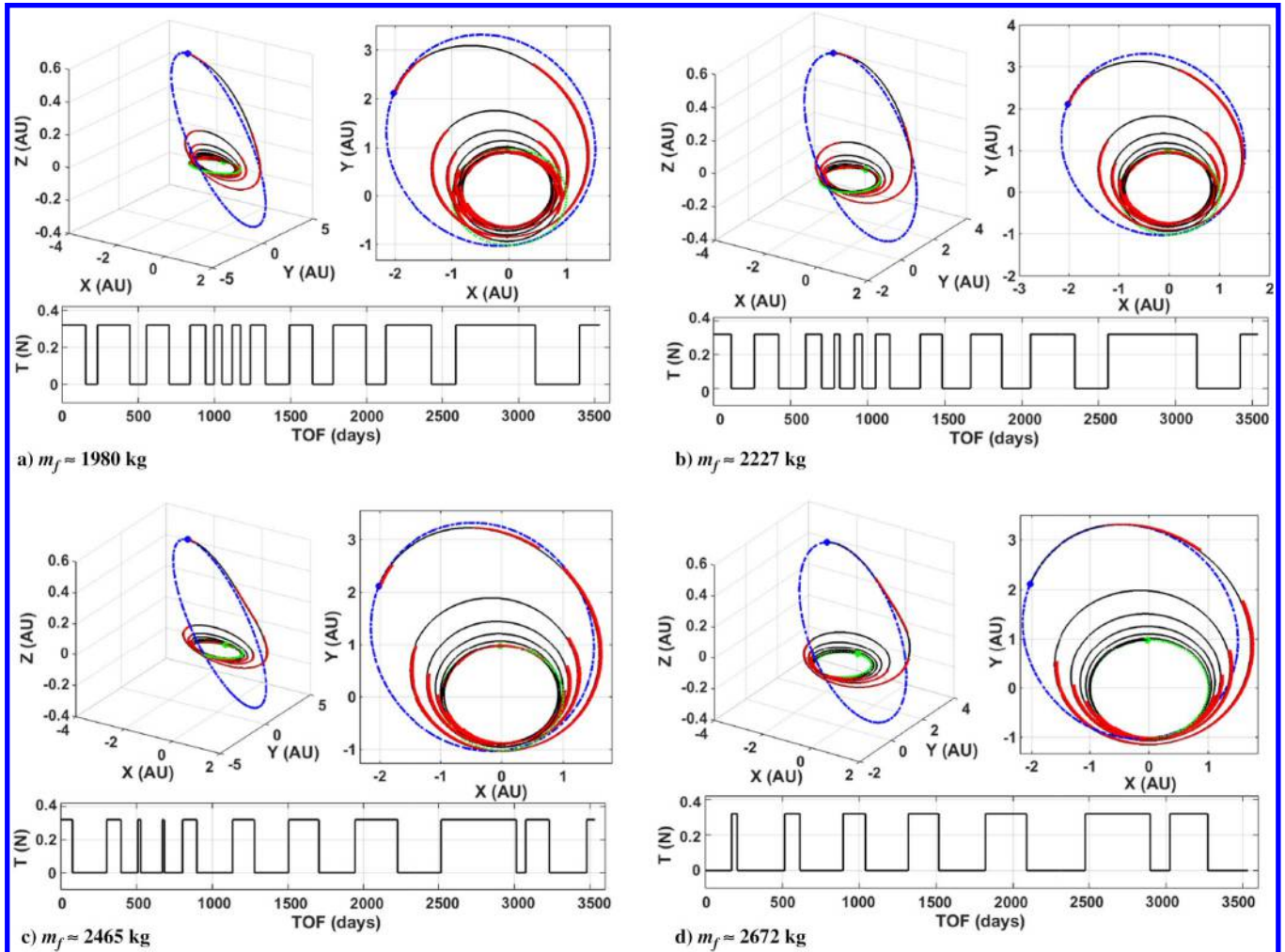


Fig. 6 Local optimal Earth-to-Dionysus trajectories when modeled in Cartesian CS with STM.

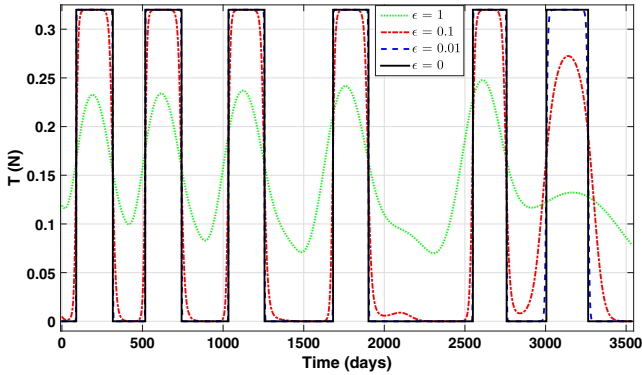


Fig. 8 Earth-to-Dionysus optimal thrust profile for different values of ϵ .

Cartesian CS with the STM method and scaled-costate vector, all of the solutions lie at the optimal level, except for only three solutions that lie on the solution level with $m_f = 2672$ kg, which in turn leads to a slight decrease in the mean value of the final mass.

Figure 7 depicts the optimal trajectory in the inertial Cartesian CS, which includes five revolutions around the sun. The optimal thrust profile of the Earth-to-Dionysus trajectory is shown in Fig. 8 for different values of the continuation parameter. Note that, with $\epsilon = 0.01$, the profile of the thrust magnitude is almost indistinguishable from that of $\epsilon = 0$, except for the last four switches. This fact can be used in problems, in which it is necessary to evaluate the feasibility of thousand low-thrust sequences in the preliminary phase. In the preliminary phase, it is not necessary to reduce the continuation parameter to a near-zero value, and instead, the sequence of subproblems could be solved until $\epsilon = 0.01$ is reached. For the problem of Earth-to-Dionysus transfer, the difference between the final mass for $\epsilon = 0.01$ and that of the optimal solution with $\epsilon = 1.0 \times 10^{-5}$ is less than 0.004%. For the sake of performance comparison, we considered the utilization of the CX derivative method to be used within the STM method [see Eq. (26)] for the evaluation of the $\partial F/\partial Z$ term. The MEOE CS and the

results are shown in Table 8. The performance of the CX method is similar to the symbolic STM method (in which $\partial F/\partial Z$ is derived symbolically), except for the simulation time, which is approximately 60% greater than the simulation time of the symbolic STM method. The CX derivative method is a quite time-consuming procedure, knowing that the expressions in $F = [\dot{x}^T, \dot{m}, \dot{\lambda}^T]^T$ are already lengthy even after the utilization of the MATLAB symbolic function generation with its optimization flag turned on to generate an efficient code. Note that the accurate calculation of the derivatives enables us to solve a smaller set of subproblems by greatly reducing the continuation parameter ϵ , for instance, by multiplying the continuation parameter by 0.01 or 0.001 instead of the original value of 0.1. For instance, we have been able to achieve the final optimal solution using the MEOE CS by solving only three subproblems corresponding to $\epsilon = 1, 0.01$, and 1.0×10^{-5} , respectively. Table 9 summarizes the results when fewer subproblems are solved. There is a 100% convergence while the simulation time is significantly lower compared to the solving of six subproblems. The results of the considered problems suggest that the MEOE CS with the FD method is the fastest and the most robust algorithm for the preliminary analysis. If more accurate solutions are required, the more accurate method for the calculation of the sensitivities has to be used, in which the STM method is a good candidate.

V. Conclusions

The paper has presented the results of investigating several methods to facilitate the numerical solution of minimum-fuel time-fixed low-thrust rendezvous trajectories. The extended logarithmic-smoothing method has been applied to EOMs formulated in terms of the spherical and in terms of MEOEs. In addition, for each model, two variants of the initial costate design vector have been considered (i.e., the original set of costates and the set of scaled-costate vector). In all cases, the optimal-control problem reduces to a nonlinear TPBVP, which may be solved by resorting to the gradient-based methods. These methods show significantly better convergence and robustness if accurate values of the sensitivities of the terminal constraints with respect to the design variables are available. The effect of using accurate sensitivity information is demonstrated by using two methods (i.e., the standard FD method and the STM method). The discontinuity of the control trajectory, which can impede the use of the STM method, can be successfully avoided by modifying the cost function by a continuation parameter. The discontinuity in the differential equations is avoided, however, at the expense of getting denser matrices, used for the calculation of the STMs. It has been shown that the sparsity pattern diminishes as the dynamics in the spherical CS has been modeled and almost totally vanishes for the dynamics when expressed in terms of the MEOEs. This happens due to both the existence of highly coupled nonlinear terms of the differential equations and the fact that the control is now expressed in terms of the states and costates. In the implementation, the floating-point underflow was first encountered at continuation parameter $\epsilon \leq 0.01$ and has been handled by a control statement over one of the auxiliary variables that is created by the MATLAB function-generator toolbox. For the case of the modified equinoctial CS, the complex-number-based derivative method has been tried, which turns out to be slower than the symbolically derived expressions for the evaluation of the STM. The proposed method based on the introduction of an extended logarithmic method with modeling of the dynamics in various CSs has been applied successfully on two different time-fixed rendezvous interplanetary transfers. The results were compared against each other in terms of the percent of converged cases, mean error in the final position and velocity vectors, mean iteration and function evaluations, and mean simulation time. The regularization combined with the STM method improved the performance of the solver for all of the cases in terms of the percent of the converged cases. The results also indicate that, for problems with higher number of revolutions like the Earth to asteroid Dionysus, the modified equinoctial elements exhibit a significantly better convergence. It is shown that for the modified equinoctial CS that the authors are able to solve fewer subproblems to further reduce the

Table 8 Performance of using the CX derivative method for sensitivity calculation of the problem of Earth to Dionysus with $\alpha = 1.0 \times 10^{-16}$

Characteristic	Model/Value
CS	MEOE
Costate treatment	Original
Sensitivity calculation	CX
Percent of converged cases	100
Δr_f , km	0.1
$\Delta v_f \times 10^{-8}$, km/s	0.4
m_f , kg	2718.32
Number of iterations	65
Number of function calls	71
Simulation time, s	31.72

Table 9 Performance of MEOE CS with STM with the solution of three subproblems corresponding to $\epsilon = 1, 0.01$, and 1.0×10^{-5} , respectively

Characteristic	Model/Value
CS	MEOE
Costate treatment	Original
Sensitivity calculation	STM
Percent of converged cases	100
Δr_f , km	3.08
$\Delta v_f \times 10^{-8}$, km/s	15.04
m_f , kg	2718.32
Number of iterations	53
Number of function calls	57
Simulation time, s	2.8

computational time of the solution without affecting the percent of converged cases. The results suggest that the modified equinoctial model with the FD method is the fastest algorithm for the preliminary analysis. The proposed combination of methods that relies on general-purpose solvers, such as MATLAB fsolve, requires a relatively short computation time and is suitable for broad trajectory search in the preliminary design.

Appendix A: Inertial Cartesian CS

The EOMs of a spacecraft, which is under the influence of a single attracting body, can be written in the Cartesian CS (Subscript c is used for the control inputs.) as

$$\ddot{\mathbf{r}} = -\frac{\mu}{\|\mathbf{r}\|^3}\mathbf{r} + \frac{T_{\max}}{m}\delta_c\mathbf{u}_c \quad (\text{A1})$$

$$\dot{m} = -\frac{T_{\max}}{c}\delta_c \quad (\text{A2})$$

in which \mathbf{r} is the position vector, μ is the gravitational parameter of the attracting body, T_{\max} is the maximum thrust of the propulsive system of the spacecraft, m is the mass of the spacecraft, $0 \leq \delta_c \leq 1$ is the throttling input, \mathbf{u}_c is the unit vector of the thrust, and $c = I_{sp}g_0$ is the exhaust velocity. The second-order vectorial differential equation can be written as two first-order differential equations:

$$\dot{\mathbf{r}} = \mathbf{v} \quad (\text{A3})$$

$$\dot{\mathbf{v}} = -\frac{\mu}{r^3}\mathbf{r} + \frac{T_{\max}}{m}\delta_c\mathbf{u}_c \quad (\text{A4})$$

$$\dot{m} = -\frac{T_{\max}}{c}\delta_c \quad (\text{A5})$$

Considering the modified cost function, Eq. (15), the Hamiltonian is written as

$$H = \frac{T_{\max}}{c}\{\delta_c - \epsilon[-\delta_c \log(\delta_c) - (1 - \delta_c) \log(1 - \delta_c)]\} \quad (\text{A6})$$

$$+ \lambda_r^T \mathbf{v} + \lambda_v^T \left(-\frac{\mu}{\|\mathbf{r}\|^3}\mathbf{r} + \frac{T_{\max}}{m}\delta_c\mathbf{u}_c \right) + \lambda_m \left(-\frac{T_{\max}}{c}\delta_c \right) \quad (\text{A7})$$

The weak form of the PMP is used to characterize the optimal thrust vector:

$$\mathbf{u}_c^* \in \arg \min_{\|\mathbf{u}_c\|=1} H \quad (\text{A8})$$

The Hamiltonian is minimized when

$$\mathbf{u}_c^* = -\frac{\lambda_v}{\lambda_r} \quad (\text{A9})$$

Exploiting the weak form of the PMP, we can also characterize the optimal throttling-magnitude input:

$$\delta_c^* \in \arg \min_{0 \leq \delta_c \leq 1} H \quad (\text{A10})$$

We can solve the preceding equation to obtain the optimal value of the throttling parameter:

$$\delta_c^* = \frac{1}{1 + 10^{\frac{1-SF_c}{\epsilon}}} \quad (\text{A11})$$

in which the Cartesian SF, SF_c , is the SF of the unperturbed cost function and is given in the following relation:

$$SF_c = \frac{\lambda_v}{m} + \frac{\lambda_m}{c} \quad (\text{A12})$$

Appendix B: Spherical CS

Defining the state and control vectors as $\mathbf{x}_s \equiv [r, \theta, \phi, \dot{r}, \dot{\theta}, \dot{\phi}, m]^T$ and $\mathbf{u}_s = [u_r, u_\theta, u_\phi]^T$, the dynamics, in a spherical CS (Subscript s is used for control inputs.) is written in the following form:

$$\dot{\mathbf{x}} = \begin{bmatrix} \Gamma \\ \mathbf{B} + \frac{T_{\max}}{m}\delta_s\mathbf{D}\mathbf{u}_s \\ -\frac{T_{\max}}{c}\delta_s \end{bmatrix} \quad (\text{B1})$$

in which $\Gamma = [\dot{r}, \dot{\theta}, \dot{\phi}]^T$, and \mathbf{B} and \mathbf{D} are defined as

$$\mathbf{B} = \begin{bmatrix} r[\dot{\theta} \cos(\phi)]^2 + r\dot{\phi}^2 - \frac{\mu}{r^2} \\ -2\frac{\dot{\theta}\dot{r}}{r} + 2\dot{\theta}\dot{\phi} \tan(\phi) \\ -2\frac{\dot{\phi}\dot{r}}{r} - \dot{\theta}^2 \sin(\phi) \cos(\phi) \end{bmatrix}, \quad \mathbf{D} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{r \cos(\phi)} & 0 \\ 0 & 0 & \frac{1}{r} \end{bmatrix} \quad (\text{B2})$$

We can form the Hamiltonian:

$$H = \frac{T_{\max}}{c}\{\delta_s - \epsilon[-\delta_s \log(\delta_s) - (1 - \delta_s) \log(1 - \delta_s)]\} \quad (\text{B3})$$

$$+ \lambda_{r\theta\phi}^T \Gamma + \lambda_{\dot{r}\dot{\theta}\dot{\phi}}^T \left(\mathbf{B} + \frac{T_{\max}}{m}\delta_s\mathbf{D}\mathbf{u}_s \right) + \lambda_m \left(-\frac{T_{\max}}{c}\delta_s \right) \quad (\text{B4})$$

in which $\lambda_{r\theta\phi} = [\lambda_r, \lambda_\theta, \lambda_\phi]^T$ and $\lambda_{\dot{r}\dot{\theta}\dot{\phi}} = [\lambda_{\dot{r}}, \lambda_{\dot{\theta}}, \lambda_{\dot{\phi}}]^T$. The weak form of the PMP is used to characterize the optimal thrust vector:

$$\mathbf{u}_s^* \in \arg \min_{\|\mathbf{u}_s\|=1} H \quad (\text{B5})$$

The Hamiltonian is minimized when

$$\mathbf{u}_s^* = -\frac{\mathbf{D}\lambda_{\dot{r}\dot{\theta}\dot{\phi}}}{\|\mathbf{D}\lambda_{\dot{r}\dot{\theta}\dot{\phi}}\|} \quad (\text{B6})$$

Note that $\mathbf{D} = \mathbf{D}^T$. Exploiting the weak form of the PMP, we can also characterize the optimal throttling-magnitude input:

$$\delta_s^* \in \arg \min_{0 \leq \delta_s \leq 1} H \quad (\text{B7})$$

We can solve the preceding equation to obtain the optimal value of the throttling parameter:

$$\delta_s^* = \frac{1}{1 + 10^{\frac{1-SF_s}{\epsilon}}} \quad (\text{B8})$$

in which the spherical SF, SF_s , is the SF of the unperturbed cost function and is given in the following relation:

$$SF_s = \frac{\|\mathbf{D}\lambda_{\dot{r}\dot{\theta}\dot{\phi}}\|}{m} + \frac{\lambda_m}{c} \quad (\text{B9})$$

References

- [1] Choueiri, E. Y., "New Dawn for Electric Rockets," *Scientific American*, Vol. 300, No. 2, 2009, pp. 58–65.
doi:10.1038/scientificamerican0209-58
- [2] Rayman, M. D., and Williams, S. N., "Design of the First Interplanetary Solar Electric Propulsion Mission," *Journal of Spacecraft and Rockets*, Vol. 39, No. 4, 2002, pp. 589–595.
doi:10.2514/2.3848.
- [3] Rayman, M. D., Fraschetti, T. C., Raymond, C. A., and Russell, C. T., "Dawn: A Mission in Development for Exploration of Main Belt Asteroids Vesta and Ceres," *Acta Astronautica*, Vol. 58, No. 11, 2006, pp. 605–616.
doi:10.1016/j.actaastro.2006.01.014
- [4] Racca, G., Marini, A., Stagnaro, L., Van Dooren, J., Di Napoli, L., Foing, B., Lumb, R., Volp, J., Brinkmann, J., and Grünagel, R. et al., "SMART-1 Mission Description and Development Status," *Planetary and Space Science*, Vol. 50, No. 14, 2002, pp. 1323–1337.
doi:10.1016/S0032-0633(02)00123-X
- [5] Keaton, P. W., *Low-Thrust Rocket Trajectories*, Los Alamos National Lab. Rept. LA-10625-MS, Los Alamos, 2002.
- [6] Betts, J. T., "Survey of Numerical Methods for Trajectory Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 2, 1998, pp. 93–207.
doi:10.2514/2.4231
- [7] Müller, W., "JACOBSON, DH and DQ MAYNE: Differential Dynamic Programming. Modern Analytic and Computational Methods in Science and Mathematics, No. 24. American Elsevier Publ. Co., Inc., New York 1970. XVI, 208 S., 17 Abb., Dfl. 51.50," *Biometrische Zeitschrift*, Vol. 15, No. 5, 1973, pp. 363–364.
- [8] Whiffen, G. J., and Sims, J., "Application of a Novel Optimal Control Algorithm to Low-Thrust Trajectory Optimization," *Proceedings of the 11th Annual AAS/AIAA Space Flight Mechanics Meeting*, AIAA, Reston, VA, 2001, pp. 1525–1540.
- [9] Colombo, C., Vasile, M., and Radice, G., "Optimal Low-Thrust Trajectories to Asteroids Through an Algorithm Based on Differential Dynamic Programming," *Celestial Mechanics and Dynamical Astronomy*, Vol. 105, Nos. 1–3, 2009, pp. 75–112.
- [10] Lantoine, G., and Russell, R. P., "A Hybrid Differential Dynamic Programming Algorithm for Constrained Optimal Control Problems. Part 2: Application," *Journal of Optimization Theory and Applications*, Vol. 154, No. 2, 2012, pp. 418–442.
doi:10.1007/s10957-012-0038-1
- [11] Lantoine, G., and Russell, R. P., "A Hybrid Differential Dynamic Programming Algorithm for Constrained Optimal Control Problems. Part 1: Theory," *Journal of Optimization Theory and Applications*, Vol. 154, No. 2, 2012, pp. 382–417.
- [12] Olympio, J. T., "A Continuous Implementation of a Second-Variation Optimal Control Method for Space Trajectory Problems," *Journal of Optimization Theory and Applications*, Vol. 158, No. 3, 2013, pp. 687–716.
- [13] Dargent, T., "Automatic Minimum Principle Formulation for Low Thrust Optimal Control in Orbit Transfers Using Complex Numbers," *21st International Symposium on Space Flights Dynamics*, Toulouse, France, Sept. 2009.
- [14] Vasile, M., Minisci, E., and Locatelli, M., "Analysis of Some Global Optimization Algorithms for Space Trajectory Design," *Journal of Spacecraft and Rockets*, Vol. 47, No. 2, 2010, pp. 334–344.
doi:10.2514/1.45742
- [15] Abdelkhalik, O., and Gad, A., "Dynamic-Size Multiple Populations Genetic Algorithm for Multigravity-Assist Trajectory Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 35, No. 2, 2012, pp. 520–529.
doi:10.2514/1.54330
- [16] Englander, J. A., Conway, B. A., and Williams, T., "Automated Mission Planning via Evolutionary Algorithms," *Journal of Guidance, Control, and Dynamics*, Vol. 35, No. 6, 2012, pp. 1878–1887.
doi:10.2514/1.54101
- [17] Kamyar, R., and Taheri, E., "Aircraft Optimal Terrain/Threat-Based Trajectory Planning and Control," *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 2, 2014, pp. 466–483.
doi:10.2514/1.61339
- [18] Vasile, M., De Pascale, P., and Casotto, S., "On the Optimality of a Shape-Based Approach Based on Pseudo-Equinoctial Elements," *Acta Astronautica*, Vol. 61, No. 1, 2007, pp. 286–297.
doi:10.1016/j.actaastro.2007.01.017
- [19] Novak, D. M., and Vasile, M., "Improved Shaping Approach to the Preliminary Design of Low-Thrust Trajectories," *Journal of Guidance, Control, and Dynamics*, Vol. 34, No. 1, 2011, pp. 128–147.
doi:10.2514/1.50434
- [20] Gondelach, D., and Noomen, R., "Hodographic-Shaping Method for Low-Thrust Interplanetary Trajectory Design," *Journal of Spacecraft and Rockets*, Vol. 52, No. 3, 2015, pp. 728–738.
doi:10.2514/1.A32991
- [21] Taheri, E., and Abdelkhalik, O., "Shape Based Approximation of Constrained Low-Thrust Space Trajectories Using Fourier Series," *Journal of Spacecraft and Rockets*, Vol. 49, No. 3, 2012, pp. 535–546.
doi:10.2514/1.A32099
- [22] Abdelkhalik, O., and Taheri, E., "Approximate On-Off Low-Thrust Space Trajectories Using Fourier Series," *Journal of Spacecraft and Rockets*, Vol. 49, No. 5, 2012, pp. 962–965.
doi:10.2514/1.A32307
- [23] Taheri, E., and Abdelkhalik, O., "Fast Initial Trajectory Design for Low-Thrust Restricted-Three-Body Problems," *Journal of Guidance, Control, and Dynamics*, Vol. 38, No. 11, 2015, pp. 2146–2160.
doi:10.2514/1.G000878
- [24] Taheri, E., and Abdelkhalik, O., "Initial Three-Dimensional Low-Thrust Trajectory Design," *Advances in Space Research*, Vol. 57, No. 3, 2016, pp. 889–903.
doi:10.1016/j.asr.2015.11.034
- [25] Thorne, J. D., and Hall, C. D., "Approximate Initial Lagrange Costates for Continuous-Thrust Spacecraft," *Journal of Guidance, Control, and Dynamics*, Vol. 19, No. 2, 1996, pp. 283–288.
doi:10.2514/3.21616
- [26] Yan, H., and Wu, H., "Initial Adjoint-Variable Guess Technique and Its Application in Optimal Orbital Transfer," *Journal of Guidance, Control, and Dynamics*, Vol. 22, No. 3, 1999, pp. 490–492.
doi:10.2514/2.7631
- [27] Dixon, L., and Bartholomew-Biggs, M. C., "Adjoint-Control Transformations for Solving Practical Optimal Control Problems," *Optimal Control Applications and Methods*, Vol. 2, No. 4, 1981, pp. 365–381.
doi:10.1002/oca.4660020405
- [28] Kluever, C. A., and Pierson, B. L., "Optimal Earth-Moon Trajectories Using Nuclear Electric Propulsion," *Journal of Guidance, Control, and Dynamics*, Vol. 20, No. 2, 1997, pp. 239–245.
doi:10.2514/2.4058
- [29] Ranieri, C. L., and Ocampo, C. A., "Optimization of Roundtrip, Time-Constrained, Finite Burn Trajectories via an Indirect Method," *Journal of Guidance, Control, and Dynamics*, Vol. 28, No. 2, 2005, pp. 306–314.
doi:10.2514/1.5540
- [30] Senent, J., Ocampo, C., and Capella, A., "Low-Thrust Variable-Specific-Impulse Transfers and Guidance to Unstable Periodic Orbits," *Journal of Guidance, Control, and Dynamics*, Vol. 28, No. 2, 2005, pp. 280–290.
doi:10.2514/1.6398
- [31] 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.
doi:10.2514/1.22984
- [32] Bertrand, R., and Epenoy, R., "New Smoothing Techniques for Solving Bang-Bang Optimal Control Problems—Numerical Results and Statistical Interpretation," *Optimal Control Applications and Methods*, Vol. 23, No. 4, 2002, pp. 171–197.
doi:10.1002/oca.709
- [33] Sullo, N., and Ceriotti, M., "A Homotopy-Based Method for Optimization of Hybrid High-Low Thrust Trajectories," *65th International Astronautical Congress, IAC 2014*, Paper IAC-14-C1.8.6, Sept.–Oct. 2014.
- [34] Betts, J. T., *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*, *Advances in Design and Control*, 2nd ed., Soc. for Industrial and Applied Mathematics, Philadelphia, PA, 2010, pp. 265–266.
- [35] Jiang, F., Baoyin, H., and Li, J., "Practical Techniques for Low-Thrust Trajectory Optimization with Homotopic Approach," *Journal of Guidance, Control, and Dynamics*, Vol. 35, No. 1, 2012, pp. 245–258.
doi:10.2514/1.52476
- [36] Battin, R. H., *An Introduction to the Mathematics and Methods of Astrodynamics*, AIAA Education Series, AIAA, New York, 1987, Chap. 9.
doi:10.2514/4.861543
- [37] Nocedal, J., and Wright, S., *Numerical Optimization*, Springer Science+Business Media, New York, 2006, pp. 30–66.
- [38] Martins, J. R., Sturdza, P., and Alonso, J. J., "The Complex-Step Derivative Approximation," *ACM Transactions on Mathematical Software*, Vol. 29, No. 3, 2003, pp. 245–262.
doi:10.1145/838250.838251
- [39] Lantoine, G., "A Methodology for Robust Optimization of Low-Thrust Trajectories in Multi-Body Environments," Ph.D. Dissertation, Georgia Inst. of Technology, Atlanta, GA, 2010.
- [40] Moré, J. J., Garbow, B. S., and Hillstom, K. E., "User Guide for MINPACK-1," No. ANL-80-74, CM-P00068642, Argonne National Lab., Argonne, IL, 1980.

This article has been cited by:

1. Fanghua Jiang, Gao Tang, Junfeng Li. 2017. Improving Low-Thrust Trajectory Optimization by Adjoint Estimation with Shape-Based Path. *Journal of Guidance, Control, and Dynamics* **40**:12, 3282-3289. [[Citation](#)] [[Full Text](#)] [[PDF](#)] [[PDF Plus](#)]
2. Zhemin Chi, Hongwei Yang, Shiyu Chen, Junfeng Li. 2017. Homotopy method for optimization of variable-specific-impulse low-thrust trajectories. *Astrophysics and Space Science* **362**:11. . [[Crossref](#)]
3. Zhengfan Zhu, Qingbo Gan, Xin Yang, Yang Gao. 2017. Solving fuel-optimal low-thrust orbital transfers with bang-bang control using a novel continuation technique. *Acta Astronautica* **137**, 98-113. [[Crossref](#)]
4. Kshitij Mall, Michael James Grant. 2017. Epsilon-Trig Regularization Method for Bang-Bang Optimal Control Problems. *Journal of Optimization Theory and Applications* **174**:2, 500-517. [[Crossref](#)]
5. Abolfazl Shirazi, Josu Ceberio, Jose A. Lozano. Evolutionary algorithms to optimize low-thrust trajectory design in spacecraft orbital precession mission 1779-1786. [[Crossref](#)]
6. Ehsan Taheri, Nan I. Li, Ilya Kolmanovsky. 2017. Co-state initialization for the minimum-time low-thrust trajectory optimization. *Advances in Space Research* **59**:9, 2360-2373. [[Crossref](#)]
7. Kui Zeng, Yunhai Geng, Baolin Wu. 2017. Shape-based analytic safe trajectory design for spacecraft equipped with low-thrust engines. *Aerospace Science and Technology* **62**, 87-97. [[Crossref](#)]