



Multi-objective low-thrust spacecraft trajectory design using reachability analysis

Nikolaus Vertovec^{a,*}, Sina Ober-Blöbaum^b, Kostas Margellos^a

^a Department of Engineering Science, University of Oxford, Oxford OX1 3PJ, United Kingdom

^b Department of Mathematics, Paderborn University, Paderborn, 33100, Germany

ARTICLE INFO

Article history:

Received 21 April 2022

Revised 2 September 2022

Accepted 6 November 2022

Available online 17 November 2022

Recommended by Prof. T Parisini

Keywords:

Optimal control

Reachability analysis

Multi-objective optimization

Pareto optimality

Hamilton–Jacobi equations

ABSTRACT

One of the fundamental problems in spacecraft trajectory design is finding the optimal transfer trajectory that minimizes the propellant consumption and transfer time simultaneously. We formulate this as a multi-objective optimal control (MOC) problem that involves optimizing over the initial or final state, subject to state constraints. Drawing on recent developments in reachability analysis subject to state constraints, we show that the proposed MOC problem can be stated as an optimization problem subject to a constraint that involves the sub-level set of the viscosity solution of a quasi-variational inequality. We then generalize this approach to account for more general optimal control problems in Bolza form. We relate these problems to the Pareto front of the developed multi-objective programs. The proposed approach is demonstrated on two low-thrust orbital transfer problems around a rotating asteroid.

© 2022 The Author(s). Published by Elsevier Ltd on behalf of European Control Association. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>)

1. Introduction

Since the Galileo mission in 1991 we have seen a steady increase in proposed missions to asteroids and comets, as they might hold the key to many scientific questions including the origins of life on earth [3]. The Dawn mission to Vesta and Ceres proved the viability of low-thrust electric propulsion for asteroid exploration [1,32], and it is expected that many upcoming missions will rely on similar low-thrust propulsion. While there has been a significant study of interplanetary transfer trajectories using low-thrust propulsion, comparatively little research has been conducted on the trajectory design in the vicinity of asteroids. We investigate a spacecraft trajectory design problem around an asteroid, where the objective is to use minimal amounts of propellant to raise an orbit while keeping flight times as short as possible. This is a multi-objective optimal control (MOC) problem, whereby one seeks to find the optimal way a dynamical system can perform a certain task, while minimizing or maximizing a set of, usually contradictory and incommensurable, objective functions [30]. Conventional optimization techniques for spacecraft trajectory design often fall

into two categories, indirect methods, based on the calculus of variations, and direct methods, whereby the optimal control problem is reformulated as a nonlinear program. Direct approaches rely on parametrization and while a candidate solution is found, there are no guarantees on the optimality of the solution. Indirect methods, meanwhile rely on necessary analytic conditions for optimality using Lagrange multipliers. Yet while optimality of the obtained solutions may be guaranteed, indirect approaches, such as multiple shooting methods, rely on a good initial approximation of the optimal trajectory [39,40]. A third approach is dynamic programming, whereby the optimality conditions are formulated in continuous time based on the Hamilton–Jacobi–Bellman (HJB) equation [39], however, it is hampered by the so-called curse of dimensionality. Despite this, unlike direct approaches, optimality is guaranteed, and unlike indirect approaches, the solution does not rely on an initial approximation of the optimal trajectory. We expand on this third approach by taking advantage of recent developments in reachability analysis.

Reachability analysis aims to find the set of points from which a target can be reached within a given time, subject to constraints. It forms a fundamental part of the dynamics and control literature and has been used extensively for controller synthesis of complex systems [4,24,29]. In recent years we have seen considerable research being conducted into computing reachable sets using Hamilton–Jacobi (HJ) reachability analysis, whereby the reachable set is derived from the viscosity solution of a HJB equation

* Corresponding author.

E-mail addresses: nikolaus.vertovec@eng.ox.ac.uk (N. Vertovec), sinaober@math.uni-paderborn.de (S. Ober-Blöbaum), kostas.margellos@eng.ox.ac.uk (K. Margellos).

accounting also for the presence of state constraints. Such a HJB framework is presented in Bokanowski et al. [11], Margellos and Lygeros [25] with more general value problems bypassing previous regularity issues presented in Altarovici et al. [2]. In Fisac et al. [17] an extension of the HJB framework to time-varying targets and constraints is considered and in Désilles and Zidani [15] the approach is extended to multi-objective control problems. HJ reachability has also been successfully applied to various aerospace applications including air traffic control [26], the climbing problem of multi-stage launchers [7], payload optimization [8], as well as most recently to the complete model of the ascent problem of multi-stage launchers [9]. One of the advantages of using HJ reachability is that the optimal trajectory can easily be constructed once the reachable set has been computed. This makes HJ reachability attractive for problems that require computing trajectories for various different initial states.

For the spacecraft trajectory design problem considered in this paper, there are two possible formulations for minimizing the burnt propellant. The first assumes that the initial mass is a free optimization variable. This approach is common during mission design where the total required fuel budget is being calculated. To this end, we formulate the spacecraft trajectory design problem as a MOC problem and show that it can be equivalently stated as an optimization problem subject to a constraint that involves the sub-level set of a certain value function. The latter is shown to be the unique continuous viscosity solution of a quasi-variational inequality that involves a HJB equation. Such value functions have been defined in Bokanowski et al. [11], Margellos and Lygeros [25] to account for the presence of state constraints. This formulation allows characterizing the Pareto front of the formulated MOC problem and also facilitates its computation by means of available numerical tools.

The second formulation of the spacecraft trajectory design problem assumes a fixed initial mass and the objective is to maximize the remaining mass after completing a given orbital maneuver. This approach is more common when the maneuver needs to be added to a given mission and the available fuel is non-negotiable. This formulation had been previously investigated by the authors in Vertovec et al. [41]. To solve this second formulation of the spacecraft trajectory design problem, we draw on research from Désilles and Zidani [15] and extend our formulation of the MOC problem to introduce an auxiliary state, allowing us to account for arbitrary problems in Bolza form, and, together with appropriate normalization and approximations allowing for a reduction of the state space, greatly improving on the method presented in Vertovec et al. [41].

Thus our contributions can be summarized as

1. the formulation of an efficient constrained MOC problem for low-thrust spacecraft trajectory design that optimizes only over the set of admissible initial states and transition times,
2. the reduction of the state space through the use of appropriate approximations,
3. the expansion of the MOC problem to allow for a generalization of the proposed methodology for arbitrary multi-objective problems in Bolza form.

This paper is organized into six sections. Section 2 contains details regarding the derivations of the spacecraft dynamics as well as the definitions of the constraints pertaining to its behavior. In Section 3 the optimal control problem is formulated while Section 4 describes how the set of admissible initial states is derived from the viscosity solution of a quasi-variational inequality. Section 5 is dedicated to the numerical computation and case study of an orbital transfer around a rotating asteroid. Finally, Section 6 provides concluding remarks and directions for future work.

2. Mathematical description and physical modeling

2.1. Spacecraft equations of motion

We begin by modeling the dynamics of the spacecraft. The spacecraft thrust is defined in spherical coordinates as

$$\mathbf{u}(t) := [\boldsymbol{\alpha}(t), \boldsymbol{\delta}(t), \mathbf{T}(t)] \in \mathcal{U}, \quad (1)$$

where $\boldsymbol{\alpha}(t) \in [-\pi, \pi]$ is the incidence angle, $\boldsymbol{\delta}(t) \in [-\frac{\pi}{2}, \frac{\pi}{2}]$ is the sideslip angle and $\mathbf{T}(t) \in [0, T_{\max}]$ is the variable thrust, with T_{\max} denoting the maximal allowable thrust. The Cartesian transformation of the thrust vector is denoted by $\mathbf{u}_x(t)$, $\mathbf{u}_y(t)$ and $\mathbf{u}_z(t)$, respectively. The compact set $\mathcal{U} = [-\pi, \pi] \times [-\frac{\pi}{2}, \frac{\pi}{2}] \times [0, T_{\max}]$ is the set of possible control input values while $\mathbf{u} \in \mathcal{U}_{ad}$ denotes the control policy and \mathcal{U}_{ad} denotes the set of admissible policies which is the set of Lebesgue time measurable functions from $[-\infty, 0]$ to \mathcal{U} . Note that time here is considered to be non-positive to facilitate the reachability problem exposition in Section 3. Boldface notation is used to denote time varying functions such as trajectories and policies, while non-boldface notation is used to denote scalars and vectors. The equations of motion of the spacecraft around a rotating body can be expressed in 3-dimensional Euclidean space as a second-order ordinary differential equation (see eg., Jiang and Baoyin [21])

$$2\boldsymbol{\Omega}(t) \times \frac{d\mathbf{R}(t)}{dt} + \boldsymbol{\Omega}(t) \times (\boldsymbol{\Omega}(t) \times \mathbf{R}(t)) + \frac{dU(\mathbf{R}(t))}{d\mathbf{R}} + \frac{d\boldsymbol{\Omega}(t)}{dt} \times \mathbf{R}(t) - \frac{\mathbf{u}(t)}{m(t)} = -\frac{d^2\mathbf{R}(t)}{dt^2}, \quad (2)$$

where $\mathbf{R}(t)$ is the radius vector from the asteroid's center of mass to the particle, the first and second time derivatives of $\mathbf{R}(t)$ are with respect to the body-fixed coordinate system, $U(\mathbf{R}(t))$ is the gravitational potential of the asteroid and $\boldsymbol{\Omega}$ is the rotational angular velocity vector of the asteroid relative to inertial space. The term $2\boldsymbol{\Omega}(t) \times \frac{d\mathbf{R}(t)}{dt}$ describes the Coriolis forces, $\boldsymbol{\Omega}(t) \times (\boldsymbol{\Omega}(t) \times \mathbf{R}(t))$, the centrifugal forces and $\frac{d\boldsymbol{\Omega}(t)}{dt} \times \mathbf{R}(t)$ the Euler forces. We consider an asteroid rotating uniformly with constant magnitude ω around the z-axis. Therefore, the Euler forces can be neglected and we can express the rotation vector as $\boldsymbol{\Omega} := \omega \mathbf{e}_z$, where \mathbf{e}_z is the unit vector along the z-axis. Following Greenwood [18], the radius vector and its derivatives are given by

$$\mathbf{R}(t) := \begin{bmatrix} \mathbf{x}(t) \\ \mathbf{y}(t) \\ \mathbf{z}(t) \end{bmatrix}, \quad \frac{d\mathbf{R}(t)}{dt} = \begin{bmatrix} \mathbf{v}_x(t) \\ \mathbf{v}_y(t) \\ \mathbf{v}_z(t) \end{bmatrix}. \quad (3)$$

The Coriolis and centrifugal forces (the first two terms in (2)) acting on the spacecraft are thus

$$2\boldsymbol{\Omega} \times \frac{d\mathbf{R}(t)}{dt} = \begin{bmatrix} -2\mathbf{v}_y(t) \\ 2\omega\mathbf{v}_x(t) \\ 0 \end{bmatrix}, \quad (4)$$

$$\boldsymbol{\Omega} \times (\boldsymbol{\Omega} \times \mathbf{R}(t)) = \begin{bmatrix} -\omega^2\mathbf{x}(t) \\ -\omega^2\mathbf{y}(t) \\ 0 \end{bmatrix}. \quad (5)$$

To model the current position, velocity, and available propellant, we define the state vector

$$\mathbf{r} := [x, y, z, v_x, v_y, v_z, \Delta m]^T \in \mathbb{R}^7, \quad (6)$$

where $\Delta m \in \mathbb{R}_{\geq 0}$ denotes the available propellant. The total spacecraft mass can be expressed as $m(t) = m_0 + \Delta m(t)$, where m_0 denotes the dry mass of the spacecraft. Following our derivations

from (2), we can formulate the dynamics of the spacecraft, $\dot{r} = f(r, u)$, as

$$f(r, u) = \begin{bmatrix} v_x \\ v_y \\ v_z \\ U_x(x, y, z) + \omega^2 x + 2\omega v_y + \frac{u_x}{m_0 + \Delta m} \\ U_y(x, y, z) + \omega^2 y - 2\omega v_x + \frac{u_y}{m_0 + \Delta m} \\ U_z(x, y, z) + \frac{u_z}{m_0 + \Delta m} \\ -\frac{\sqrt{u_x^2 + u_y^2 + u_z^2}}{v_{\text{exhaust}}} \end{bmatrix}, \quad (7)$$

where $v_{\text{exhaust}} \in \mathbb{R}_{\geq 0}$ is the exhaust velocity used to express the depletion of mass as propellant is burned, U_x , U_y and U_z are the derivatives of the gravitational potential in the direction of the unit vectors e_x , e_y and e_z , respectively, and where for brevity we neglect the time dependence by denoting $r = \mathbf{r}(t)$, $v_x = \mathbf{v}_x(t)$, and similarly for the other states.

2.2. State constraints

Since the dynamics of the spacecraft were derived for orbits in the vicinity of the asteroid, we need to enforce state constraints on x, y, z . We naturally also need to ensure that we bound the amount of propellant available. Assuming that the burnout mass of the spacecraft is the same as the dry mass, we set $m_{\min} := 0$ and $m_{\max} := m_{\text{propellant}}$ and impose $m_{\min} \leq \Delta m \leq m_{\max}$.

Due to particles ejected from the asteroid, we do not want to fall below a circular orbit with radius $\rho := \sqrt{x^2 + y^2 + z^2}$ of approximately $\rho_{\min} = 1$ km. Furthermore, in order for the two-body problem under discussion to be valid and the influence of other bodies in the solar system to be negligible, we need to stay within the sphere of influence (SOI) of the asteroid. The SOI can be approximated as in Seefelder [37] by $\rho_{\text{SOI}} \approx a \left(\frac{M_1}{M_2} \right)^{\frac{2}{3}}$, where a is the semi-major axis of the asteroid's orbit around the sun (1.5907×10^8 km), M_1 is the Mass of the asteroid (1.4091×10^{12} kg) and M_2 is the mass of the sun (1.9890×10^{30} kg). Therefore, the sphere of influence of the asteroid is approximately $\rho_{\max} = \rho_{\text{SOI}} \approx 8.74$ km. The set of states that satisfy the aforementioned restrictions is given by

$$\mathcal{K} := \{r \in \mathbb{R}^7 : \rho \in [\rho_{\min}, \rho_{\max}], m \in [m_{\min}, m_{\max}]\}.$$

The target orbit that we would like to transfer to is denoted by the closed target set $\mathcal{C} \subset \mathcal{K}$.

The initial orbit that we start at is denoted by the closed initial set $\mathcal{I} \subset \mathcal{K}$. Note that the initial and the target orbit restrict only the position and the velocity, but allow the mass to take any admissible value within $[m_{\min}, m_{\max}]$.

While Cartesian coordinates are useful for modeling the behavior of an object around a rotating body, since we restrict all admissible states to lie within the set \mathcal{K} , which constrains the radius ρ , it is more efficient to recast our problem in spherical coordinates. To this end, define a_ρ, a_θ, a_ψ as the transformations of

$$a_x := U_x(x, y, z) + \omega^2 x + 2\omega v_y,$$

$$a_y := U_y(x, y, z) + \omega^2 y - 2\omega v_x,$$

$$a_z := U_z(x, y, z).$$

The tangential velocity in the $x - y$ plane, v_t , and its perpendicular counterpart, v_\perp , can then be defined as follows:

$$\begin{bmatrix} v_t \\ v_\perp \end{bmatrix} = \begin{bmatrix} \rho \dot{\theta} \sin \psi \\ \rho \dot{\psi} \end{bmatrix}, \quad (8)$$

$$\begin{bmatrix} a_t \\ a_\perp \end{bmatrix} = \begin{bmatrix} \sin \psi [a_\theta \rho + \dot{\theta} v_\rho] + \dot{\theta} v_\perp \cos \psi \\ \rho \dot{\psi} + a_\psi \rho \end{bmatrix}. \quad (9)$$

Then we can restate the system dynamics in spherical coordinates as

$$r = [\rho, \theta, \psi, v_\rho, v_t, v_\perp, \Delta m]^T \in \mathbb{R}^7, \quad (10)$$

$$f(r, u) = \begin{bmatrix} v_\rho \\ \frac{v_t}{\rho \sin \psi} \\ \frac{v_\perp}{\rho} \\ a_\rho + \frac{f}{m_0 + \Delta m} \cos \alpha \\ a_t + \frac{f}{m_0 + \Delta m} \sin \alpha \sin \delta \\ a_\perp + \frac{f}{m_0 + \Delta m} \sin \alpha \cos \delta \\ -\frac{f}{v_{\text{exhaust}}} \end{bmatrix}. \quad (11)$$

With a slight abuse of notation, we also redefine \mathcal{I} , \mathcal{C} , and \mathcal{K} in spherical coordinates. Finally, we impose the following assumptions on the spacecraft dynamics.

Assumption 2.1. For every $r \in \mathcal{K}$ the set $\{f(r, u) : u \in \mathcal{U}\}$ is a compact convex subset of \mathbb{R}^7 .

Assumption 2.2. $f : \mathbb{R}^7 \times \mathcal{U} \rightarrow \mathbb{R}^7$ is bounded and there exists an $L_f > 0$ such that for every $u_1, u_2 \in \mathcal{U}$,

$$\|f(r_1, u_1) - f(r_2, u_2)\| \leq L_f \|r_1 - r_2\|.$$

Due to norm equivalence, the choice of norm is irrelevant and not further discussed. Using Assumptions 2.1 and 2.2, for any control policy $\mathbf{u} \in \mathcal{U}_{ad}$, any initial state $r_0 \in \mathcal{K}$ and transfer time $t_f > 0$, the system admits a unique, absolutely continuous solution on $[-t_f, 0]$ (see Sastry [34]).

3. Problem statement

3.1. Multi-objective optimal control problem

Having defined the system dynamics, we are now in a position to discuss how to find trajectories that start on an initial orbit, \mathcal{I} , and take the spacecraft to some final orbit, \mathcal{C} . Additionally, the objective is to keep the flight time and required propellant as small as possible. Thus, the multi-objective optimal control problem can be formulated as a minimization problem whereby the first goal is to minimize the required propellant, Δm , and the second is to minimize the required time for the orbit change, i.e., the transfer time, denoted by t_f . The trajectory, \mathbf{r} , which is the solution of (11), belongs to the Sobolev space $\mathbb{W}^{1,1}(\mathbb{R}^7)$. The set of trajectory-control pairs on $[-t_f, 0]$ starting at r_0 with transfer time t_f is denoted as:

$$\begin{aligned} \Pi_{r_0, t_f} &:= \{(\mathbf{r}, \mathbf{u}) : \dot{\mathbf{r}}(t) = f(\mathbf{r}(t), \mathbf{u}(t)), \quad \forall t \in [-t_f, 0]; \\ \mathbf{r}(-t_f) &= r_0\} \subset \mathbb{W}^{1,1}(\mathbb{R}^7) \times \mathcal{U}_{ad}. \end{aligned}$$

Note that as in Chen et al. [14] we adopt the convention that 0 denotes the terminal time hence the transfer time t_f denotes the time duration. Under Assumption 2.1 and by Filippov's Theorem [23, pg.121], we can conclude, that Π_{r_0, t_f} is compact.

Remark 3.1. For similar applications as the one discussed in this paper, where Assumption 2.1 might not hold, we refer to Bokanowski et al. [9] where convexification of the dynamics is considered in order to ensure that the set of absolutely continuous solutions of the problem is closed.

The set of admissible (in the sense of satisfying the state constraints) trajectory-control pairs on $[-t_f, 0]$ starting at r_0 with transfer time t_f is denoted as:

$$\Pi_{r_0, t_f}^{\mathcal{K}, \mathcal{C}} := \{(\mathbf{r}, \mathbf{u}) \in \Pi_{r_0, t_f} : \mathbf{r}(t) \in \mathcal{K}, \quad \forall t \in [-t_f, 0]; \mathbf{r}(0) \in \mathcal{C}\} \\ \subset \mathbb{W}^{1,1}(\mathbb{R}^7) \times \mathcal{U}_{ad}.$$

Finally, the set of admissible initial state and transfer time pairs is denoted as

$$\Gamma := \{(r_0, t_f) \in \mathbb{R}^7 \times [0, +\infty) \text{ such that } \Pi_{r_0, t_f}^{\mathcal{K}, \mathcal{C}} \neq \emptyset\}.$$

For a given initial state $r_0 \in \mathbb{R}^7$ and transfer time $t_f \in [0, +\infty)$, we can define the cost functions as $J_1(r_0, t_f) := \Delta m$ and $J_2(r_0, t_f) := t_f$, where Δm is the 7-th element of the state vector r_0 . The 2-dimensional objective function $J : \mathbb{R}^7 \times [0, +\infty) \rightarrow \mathbb{R}^2$ can then be written as

$$J(r_0, t_f) := [J_1(r_0, t_f), J_2(r_0, t_f)]^T. \quad (12)$$

We are now in a position to formulate the multi-objective optimal control problem under study as

$$\begin{aligned} & \text{minimize}_{(r_0, t_f) \in \mathcal{I} \times [0, \infty)} J(r_0, t_f) \\ & \text{subject to } (r_0, t_f) \in \Gamma \end{aligned} \quad (13)$$

3.2. Pareto optimality

The solution of (13) in general does not consist of a single isolated point, but rather a set of optimal compromises between the objectives J_1 and J_2 [27].

Definition 3.2. A solution (r_0, t_f) is considered Pareto optimal if $\nexists(\hat{r}_0, \hat{t}_f) \in \Gamma$ such that $J(\hat{r}_0, \hat{t}_f) < J(r_0, t_f)$,

where a vector a is considered less than b (denoted $a < b$) if for every element a_i and b_i the relation $a_i < b_i$ holds. The relations $\leq, \geq, >$ are defined in an analogous way. Following Definition 3.2, a solution (r_0, t_f) is considered Pareto optimal if it is not possible to improve all its performance metrics $J_1(r_0, t_f), J_2(r_0, t_f)$ simultaneously. The set of Pareto optimal solutions is called the Pareto set \mathcal{P}_S , while its image is the Pareto front \mathcal{P}_F . Therefore, the solution of (13), i.e., the set of minimizing (r_0, t_f) pairs, is the desired Pareto set, while the cost function corresponding to the minimizing (r_0, t_f) pairs is the Pareto front.

To allow for mission designers to determine a compromise between minimizing required propellant and transfer times, we wish to compute the Pareto front. However, the unconventional constraint in (13) ensuring a solution (r_0, t_f) is feasible, prevents us from solving (13) with standard MOC problem solvers. Therefore, we will next discuss how we can recast the constraint $(r_0, t_f) \in \Gamma$ to a standard nonlinear inequality constraint, which will allow us to compute the Pareto front by means of conventional MOC problem solvers.

4. Solution to multi-objective optimal control problems

To find an equivalent formulation for the constraint $(r_0, t_f) \in \Gamma$, let $g(r)$ and $v(r)$ be two Lipschitz functions (with Lipschitz constants L_g and L_v , respectively) chosen such that

$$g(r) \leq 0 \iff r \in \mathcal{K}, \\ v(r) \leq 0 \iff r \in \mathcal{C}.$$

This can be achieved by choosing $g(r)$ and $v(r)$ as the signed distance to the set \mathcal{K} and \mathcal{C} , respectively.

Next, we consider the value function ω :

$$\omega(r_0, t_f) := \inf_{(\mathbf{r}, \mathbf{u}) \in \Pi_{r_0, t_f}} \left\{ v(\mathbf{r}(0)) \bigvee \max_{\tau \in [-t_f, 0]} g(\mathbf{r}(\tau)) \right\}, \quad (14)$$

where $a \bigvee b$ denotes $\max(a, b)$. We are now in a position to use the value function to decide if, for a given initial state and transfer

time, there exists a corresponding admissible trajectory. Thus we can introduce an equivalent formulation of (13).

Theorem 4.1. The constrained MOC problem, (13), is equivalent to

$$\begin{aligned} & \text{minimize}_{(r_0, t_f) \in \mathcal{I} \times [0, \infty)} J(r_0, t_f) \\ & \text{subject to } \omega(r_0, t_f) \leq 0. \end{aligned} \quad (15)$$

Proof. We show that $(r_0, t_f) \in \Gamma \iff \omega(r_0, t_f) \leq 0$.

Case A: Consider $(r_0, t_f) \in \Gamma$. For the sake of contradiction assume that $\omega(r_0, t_f) > 0$. This then implies that for all $(\mathbf{r}, \mathbf{u}) \in \Pi_{r_0, t_f}^{\mathcal{K}, \mathcal{C}}$ either $v(\mathbf{r}(0)) > 0 \iff \mathbf{r}(0) \notin \mathcal{C}$ or there exists $\tau \in [-t_f, 0]$ such that $g(\mathbf{r}(\tau)) > 0 \iff \mathbf{r}(\tau) \notin \mathcal{K}$. This contradicts the fact that $(r_0, t_f) \in \Gamma$ establishing that $(r_0, t_f) \in \Gamma$ implies $\omega(r_0, t_f) \leq 0$.

Case B: Consider $(r_0, t_f) \in \mathcal{I} \times [0, \infty)$, such that $\omega(r_0, t_f) \leq 0$. Under Assumption 2.1, applying Weierstrass' Theorem on the existence of minima for compact sets [2,6], we can conclude that the infimum over Π_{r_0, t_f} exists, and thus, $\omega(r_0, t_f) \leq 0$ implies the existence of a trajectory-control pair $(\mathbf{r}, \mathbf{u}) \in \Pi_{r_0, t_f}$, such that for all $t \in [-t_f, 0]$, $g(\mathbf{r}(t)) \leq 0$ and $v(\mathbf{r}(0)) \leq 0$. By definition of the function g and v , we thus have $\mathbf{r}(t) \in \mathcal{K}$ for all $t \in [-t_f, 0]$ and $\mathbf{r}(0) \in \mathcal{C}$, which in turns implies $(\mathbf{r}, \mathbf{u}) \in \Pi_{r_0, t_f}^{\mathcal{K}, \mathcal{C}}$. Therefore,

$$\omega(r_0, t_f) \leq 0 \implies (r_0, t_f) \in \Gamma,$$

thus concluding the proof. \square

Theorem 4.1 implies that the Pareto front can be computed from the solution of (15). To achieve this we discuss how to compute ω .

4.1. Value function computation

To begin to discuss how ω can be obtained, we introduce the Hamiltonian $H : \mathbb{R}^7 \times \mathbb{R}^7 \rightarrow \mathbb{R}$,

$$H(r, q) := - \min_{u \in \mathcal{U}} (q^T f(r, u)), \quad (16)$$

where $q \in \mathbb{R}^7$ is the costate vector.

Theorem 4.2. The value function ω is the unique continuous viscosity solution of the following quasi-variational inequality

$$\begin{cases} 0 = \max \{g(r) - \omega(r, t), \partial_t \omega + H(r, \nabla_r \omega)\} \\ \quad \text{for all } t \in [0, \infty), r \in \mathbb{R}^7, \\ \omega(r, 0) = (v(r) \bigvee g(r)) \text{ for all } r \in \mathbb{R}^7, \end{cases}$$

Since the Dynamic Programming Principle [6] holds for $h \in [0, t_f]$, $(r_0, t_f) \in \mathcal{K} \times \mathbb{R}$ with $t_f \geq 0$:

$$\omega(r_0, t_f) = \inf_{(\mathbf{r}, \mathbf{u}) \in \Pi_{r_0, t_f}} \left\{ \omega(\mathbf{r}(h - t_f), t_f - h) \bigvee \max_{s \in [-t_f, h - t_f]} g(\mathbf{r}(s)) \right\},$$

the proof of Theorem 4.2 follows standard arguments for viscosity solutions, as shown in Altarovici et al. [2], and Margellos and Lygeros [25]. Note that the infimum should be understood to be over the restriction of Π_{r_0, t_f} over $[-t_f, h - t_f]$.

In order to solve the quasi-variational inequality in Theorem 4.2, we employ a finite differences scheme. As in Bokanowski et al. [11], Margellos and Lygeros [25], a consequence following from Theorem 4.2 is the Lipschitz continuity of the value function.

Proposition 4.3. The value function ω is Lipschitz continuous.

The proof is similar to the more general proof of Proposition 4.9 that is introduced in the sequel. Proposition 4.3 allows us to make statements about the discrete-continuous error estimate, for which we refer to Theorem 5 in Bokanowski et al.

[11], as well as the convergence of the value function, for which we refer to Proposition 6 in Fialho and Georgiou [16].

The Hamiltonian admits an explicit form. To this end, consider the term

$$C(r, q) := q_1 v_\rho + q_2 v_t + q_3 v_\perp + q_4 a_\rho + q_5 a_t + q_6 a_\perp.$$

Then we can write the Hamiltonian as

$$H(r, q) := -\min_{u \in \mathcal{U}} \left(\frac{T}{m_0 + \Delta m} (q_4 \cos \alpha + \sin \alpha (q_5 \sin \delta + q_6 \cos \delta)) - q_7 \frac{T}{v_{\text{exhaust}}} \right) - C(r, q). \quad (17)$$

As T is always positive, the thrust angles can be optimized separately; see Appendix for more details. To this end, the derivation of α^* and δ^* follows a similar procedure as in Bokanowski et al. [9] and is listed in the Appendix. After applying the optimal thrust angles, the Hamiltonian becomes affine in T and we are able to find the optimal thrust magnitude. In particular, $H(r, q)$ becomes

$$H(r, q) = -\min_{T \in [0, T_{\max}]} \left(-\frac{T}{m_0 + \Delta m} \sqrt{q_4^2 + q_5^2 + q_6^2} - q_7 \frac{T}{v_{\text{exhaust}}} \right) - C(r, q) \quad (18)$$

$$\Rightarrow T^* := \begin{cases} T_{\max} & \text{if } \frac{q_7}{v_{\text{exhaust}}} + \frac{\sqrt{q_4^2 + q_5^2 + q_6^2}}{m_0 + \Delta m} \geq 0 \\ 0 & \text{otherwise} \end{cases}. \quad (19)$$

Remark 4.4. The singular control case, $\frac{q_7}{v_{\text{exhaust}}} + \frac{\sqrt{q_4^2 + q_5^2 + q_6^2}}{m_0 + \Delta m} = 0$, is negligible for the consideration of the optimal Hamiltonian. For the optimal control computation during the trajectory calculation, we have numerically investigated the occurrence of singular arcs and found no instances in which $\frac{\sqrt{q_4^2 + q_5^2 + q_6^2}}{m_0 + \Delta m} = -\frac{q_7}{v_{\text{exhaust}}}$ over an extended interval, thus we do not further consider the singular control case.

Finally, applying T^* and rewriting the minimum as the maximum of the negation of the associated function, the Hamiltonian takes the following analytic form

$$H(r, q) = -C(r, q) + \max \left(q_7 \frac{T_{\max}}{v_{\text{exhaust}}} + \frac{T_{\max}}{m_0 + \Delta m} \sqrt{q_4^2 + q_5^2 + q_6^2}, 0 \right).$$

4.2. Extension to problems in Bolza form

We will now generalize our approach to problems where the objective functions do not rely only on the initial state and are written in Bolza form. To achieve this we need to introduce auxiliary states. As in Altarovici et al. [2], and Désilles and Zidani [15], we show how problems in Bolza form are reformulated into Mayer form, and then show how problems in Mayer form are solved in a similar fashion as in Section 3. Consider the p -dimensional objective function defined as:

$$J_{\text{Bolza}}(\mathbf{r}, \mathbf{u}, t_f) := J_t(\mathbf{r}(0)) + \int_{-t_f}^0 J_r(\mathbf{r}(s), \mathbf{u}(s)) ds, \quad (20)$$

where J_t denotes the terminal cost and J_r denotes the running cost. We impose the following assumptions, as in Désilles and Zidani [15].

Assumption 4.5. J_t is locally Lipschitz continuous on \mathbb{R}^7 with Lipschitz constant $L_t(R)$ for every neighborhood $R \subset \mathbb{R}^7$.

Assumption 4.6. J_r is continuous on $\mathbb{R}^7 \times \mathcal{U}$. Moreover, J_r is locally Lipschitz continuous on the first variable with Lipschitz constant $L_r(R)$ for every neighborhood $R \subset \mathbb{R}^7$.

Remark 4.7. To ease notation, we omit the dependents on the neighborhood for the Lipschitz constants and instead assume the existence of a global Lipschitz constant L_t and L_r , respectively.

Next, we define an auxiliary state $z \in \mathbb{R}^p$ as:

$$\begin{cases} \dot{\mathbf{z}}(s) = -J_r(\mathbf{r}(s), \mathbf{u}(s)), & \forall s \in [-t_f, 0] \\ \mathbf{z}(0) = z_0. \end{cases} \quad (21)$$

where z_0 becomes an optimization parameter and $\mathbf{z} \in \mathbb{W}^{1,1}(\mathbb{R}^p)$. The auxiliary state captures the cumulative running cost and thus is treated as an additional state. In the same manner we previously ensured a trajectory, \mathbf{r} , stayed within the set \mathcal{K} , we bound \mathbf{z} and ensure that the integrated running cost, added with the terminal cost, stays below some value z_0 . To capture all possible trajectories, we introduce the set

$$\mathcal{Z}_{r_0, t_f, z_0} := \left\{ (\mathbf{r}, \mathbf{u}, \mathbf{z}) : (\mathbf{r}, \mathbf{u}) \in \Pi_{r_0, t_f}; \dot{\mathbf{z}}(s) = -J_r(\mathbf{r}(s), \mathbf{u}(s)), \forall s \in [-t_f, 0]; \mathbf{z}(0) = z_0 \right\}, \quad (22)$$

and make the following assumption.

Assumption 4.8. For every $r \in \mathbb{R}^7$, the set

$$\left\{ \begin{bmatrix} f(r, u) \\ -J_r(r, u) \end{bmatrix} : u \in \mathcal{U}, \right\}$$

is a compact convex subset of $\mathbb{R}^7 \times \mathbb{R}^p$.

We now introduce the auxiliary value function ϑ :

$$\vartheta(r_0, t_f, z_0) := \inf_{(\mathbf{r}, \mathbf{u}, \mathbf{z}) \in \mathcal{Z}_{r_0, t_f, z_0}} \left\{ \bigvee_i [J_t^i(\mathbf{r}(0)) - z^i(-t_f)] \bigvee \nu(\mathbf{r}(0)) \bigvee \max_{s \in [-t_f, 0]} g(\mathbf{r}(s)) \right\}, \quad (23)$$

where $\bigvee_i x^i$ denotes the maximum element of the vector x . As with ω , the term $\max_{s \in [-t_f, 0]} g(\mathbf{r}(s))$ and $\nu(\mathbf{r}(0))$ ensures that any trajectory \mathbf{r} remains in \mathcal{K} and terminates in \mathcal{C} . The additional term $J_t^i(\mathbf{r}(0)) - z^i(-t_f)$ ensures that the integrated running cost, J_r , combined with the terminal cost, J_t , never grows larger than z_0 . Thus, in addition to ensuring that (\mathbf{r}, \mathbf{u}) are admissible trajectory control pairs, the sub-zero level set of ϑ bounds the terminal and integrated running cost. Therefore,

$$\vartheta(r_0, t_f, z_0) \leq 0 \iff \left[\exists (\mathbf{r}, \mathbf{u}) \in \Pi_{r_0, t_f}^{\mathcal{K}, \mathcal{C}}, J_{\text{Bolza}}(\mathbf{r}, \mathbf{u}, t_f) \leq z_0 \right]. \quad (24)$$

We are now in a position to introduce the generalized multi-objective optimal control problem for objective functions in Bolza form:

$$\begin{aligned} & \text{minimize} && z_0 \\ & \text{subject to} && \vartheta(r_0, t_f, z_0) \leq 0, \end{aligned} \quad (25)$$

where z_0 represents an upper bound for the term $J_{\text{Bolza}}(\mathbf{r}, \mathbf{u}, t_f)$, without explicit knowledge of \mathbf{r} or \mathbf{u} .

The generalized value function can again be obtained as the unique continuous viscosity solution of a quasi-variational inequality

$$\begin{cases} \max \left\{ g(r) - \vartheta(r, t, z), \partial_t \vartheta + H(r, \nabla_r \vartheta, \nabla_z \vartheta) \right\} = 0 \\ \quad \text{for all } t \in [0, \infty), r \in \mathbb{R}^7, z \in \mathbb{R}^p, \\ \vartheta(r, 0, z) = \bigvee_i [J_t^i(r) - z^i] \bigvee \nu(r) \bigvee g(r) \\ \quad \text{for all } r \in \mathbb{R}^7, \end{cases} \quad (26)$$

where the Hamiltonian is defined as

$$H(r, q_r, q_z) := \min_{u \in \mathcal{U}} \left(q_r^T f(r, u) - q_z^T J_r(r, u) \right).$$

Proposition 4.9. *The value function ϑ is Lipschitz continuous.*

The proof can be found in the Appendix. Proposition 4.9 can be used to show that a numerical solution of (26) (in the viscosity sense) can always be determined.

Under Assumptions 2.2, 4.5, 4.6 and 4.8, by Filippov's Theorem [23], the problem (23) admits an optimal solution, which implies the existence of an admissible \mathbf{r} and \mathbf{z} [15]. This yields the following relationship due to (21)

$$\bigvee_i [J_t^i(\mathbf{r}(0)) - \mathbf{z}^i(-t_f)] = \bigvee_i \left[J_t^i(\mathbf{r}(0)) + \int_{-t_f}^0 J_r^i(\mathbf{r}(s), \mathbf{u}(s)) ds - z_0^i \right]. \quad (27)$$

5. Numerical approximation and results

We will now discuss how the value functions can be obtained numerically, prior to discussing how the spacecraft trajectory design problem is solved. Following Proposition 4.3, a numerical solution to (26) can be found. To this end, we employ the Level Set Methods toolbox of Mitchell [28]. For the computation of ω , we use a Lax-Friedrichs Hamiltonian

$$\mathcal{H}(r, p^-, p^+) = H\left(r, \frac{p^- + p^+}{2}\right) - \sum_{k=1}^7 \frac{\alpha_k}{2} (p^+ - p^-), \quad (28)$$

where p^+ and p^- are the right and left derivatives computed using an appropriate fifth-order weighted essentially non-oscillatory (WENO) scheme. The Lax-Friedrichs Hamiltonian consists of an analytic expression of the Hamiltonian (derived previously), as well as a dissipation term, which is scaled by the dissipation coefficients α_k . The dissipation coefficients α_k needs to satisfy

$$\alpha_k \geq \left| \frac{\partial H}{\partial q_k} \right|. \quad (29)$$

Since we need α_k to serve as an upper bound, we consider the control input that maximizes the Hamiltonian:

$$\left| \frac{\partial H}{\partial q_k} \right| = \left| -\frac{\partial C(r, q)}{\partial q_k} + \frac{\partial}{\partial q_k} \max \left(q_7 \frac{T_{\max}}{v_{\text{exhaust}}} + \frac{T_{\max}}{m_0 + \Delta m} \sqrt{q_4^2 + q_5^2 + q_6^2}, 0 \right) \right| \quad (30)$$

$$\alpha_k = \begin{cases} |v_\rho| & k = 1 \\ \left| \frac{v_t}{\rho \sin \psi} \right| & k = 2 \\ \left| \frac{v_\perp}{\rho} \right| & k = 3 \\ \left| a_\rho - \max\left(0, \frac{T_{\max} q_3}{(m_0 + \Delta m) \sqrt{q_3^2 + q_4^2 + q_5^2}}\right) \right| & k = 4 \\ \left| a_t - \max\left(0, \frac{T_{\max} q_4}{(m_0 + \Delta m) \sqrt{q_3^2 + q_4^2 + q_5^2}}\right) \right| & k = 5 \\ \left| a_\perp - \max\left(0, \frac{T_{\max} q_5}{(m_0 + \Delta m) \sqrt{q_3^2 + q_4^2 + q_5^2}}\right) \right| & k = 6 \\ \frac{T_{\max}}{v_{\text{exhaust}}} & k = 7, \end{cases} \quad (31)$$

For a further discussion of the Lax-Friedrichs Hamiltonian and WENO scheme, we refer to Osher and Fedkiw [31], while for a discussion of the convergence of ω and the derivation of a necessary Courant-Friedrichs-Lewy condition, we refer to Bokanowski et al. [11], Hermosilla et al. [19], Mitchell [28].

5.1. Implementation

To illustrate the theoretical results of the previous sections, we consider a spacecraft on an initial near circular orbit around asteroid Castalia 4769. The goal is to compute an efficient transfer trajectory that raises the spacecraft to a stable orbit at an altitude of 6117.5 m above the asteroid. For the derivation of a stable orbit

Table 1
Normalization.

Scale	Values
Distance	Initial radius ρ_0
Velocity	2 m/s
Time	ρ_0/v_0
Mass	1 kg
Force	Maximum thrust T_{\max}

Table 2
Spatial grid configuration.

	ρ	v_ρ	v_t	Δm
Points	50	40	40	32
Spacing	0.0045	0.0416	0.0088	0.0067
Minimum	0.8067	-0.2495	-1.4154	-0.0533
Maximum	1.0270	1.3722	-1.0704	0.1533

around Castalia 4769 we refer to Kulumani and Lee [22] and references therein. The gravity of Castalia 4769 was modeled by means of a spherical harmonic expansion as discussed in Hudson and Ostro [20], Scheeres et al. [35,36]. Even though the proposed theoretical framework allows us to tackle problems of any state dimension, the available numerical tools and computational power limit us to only study the lower-dimensional planar case of the application. We, therefore, omit the states ψ and v_\perp to consider only

$$r = [\rho, \theta, v_\rho, v_t, \Delta m]^T \in \mathbb{R}^5. \quad (32)$$

To avoid ill-conditioning when solving the HJB equation, the state vector is normalized using the constants introduced in Table 1. This results in the following dynamics:

$$f(r, u) = \begin{bmatrix} v_\rho \\ v_t \\ a_\rho + \frac{cT}{m_0 + \Delta m} \cos \alpha \\ a_t + \frac{cT}{m_0 + \Delta m} \sin \alpha \\ -\frac{cT}{v_{\text{exhaust}}} \end{bmatrix}, \quad (33)$$

where $c = T_{\max} \rho_0 / (m_0 V_0^2)$ is a normalization constant. ρ_0 , m_0 , and v_0 denote the initial radius, mass, and velocity of the initial orbit, respectively.

The optimal control policy and trajectory $(\mathbf{r}, \mathbf{u}) \in \Pi_{r_0, t_f}^{\mathcal{K}, C}$ can be constructed efficiently using the numerical approximation of ω . For a given $N \in \mathbb{N}$ we consider the time step $h = \frac{1}{N}$ and a uniform grid of $[-t_f, 0]$ with spacing $s^k = \frac{k}{N}$. Let us define the state $\{r^k\}_{k=0}^N$ and control $\{u^k\}_{k=0}^{N-1}$ for the numerical approximation of the optimal trajectory and control policy. Setting r_0 as the initial orbit, we proceed by iteratively computing the control value

$$u^k(r^k) \in \underset{u \in \mathcal{U}}{\operatorname{argmin}} \omega(r^k + hf(r^k, u), s^k) \bigvee g(r^k).$$

For a given $\omega(r^k, s^k)$, this is done by numerically taking the partial derivatives along each grid direction to estimate the costate vector, q , and then determining the optimal control values as the minimizer of the Hamiltonian, H . After u^k is determined we compute r^{k+1} using the Matlab ode113 function, a variable-order Adams-Bashforth-Moulton method of order 1 to 13 [38], and increment k . For the implementation, we discretized the interval $[-t_f, 0]$ using $N = 4000$ grid points.

As shown in Fig. 1, when considering orbits further than 4 km away from the surface of the asteroid, the variation of the gravitational acceleration along θ becomes negligible. It is, therefore, possible to approximate the gravitational terms in spherical coordinates as

$$\begin{aligned} U_\rho(\rho, \theta) &\approx U_\rho(\rho) \\ U_\theta(\rho, \theta) &\approx 0. \end{aligned}$$

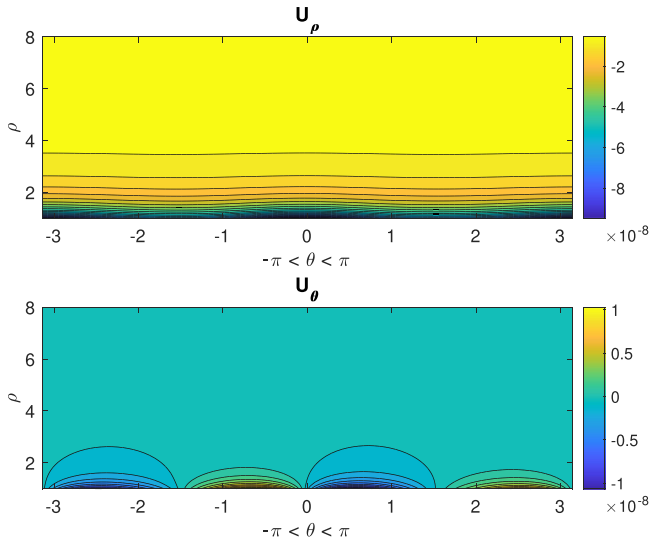


Fig. 1. Gravitational acceleration comparing U_ρ and U_θ around Castalia 4769. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

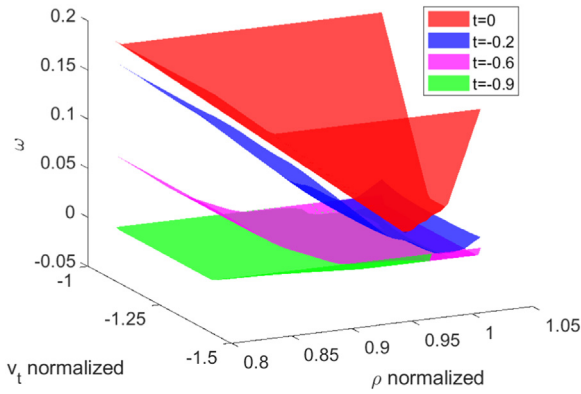


Fig. 2. The evolution of the value function, ω , projected in the $\rho - v_t$ plane for $v_\rho = 0$ and $\Delta m = 100$ g. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Using this approximation makes a_ρ and a_t independent of θ . This allows us to omit a grid dimension while numerically solving the quasi-variational inequality in [Theorem 4.2](#), greatly reducing the computational cost. Thus the final set of states used for the computation of the value function is

$$r = [\rho, v_\rho, v_t, \Delta m]^T \in \mathbb{R}^4. \quad (34)$$

During the calculation of the optimal trajectory, θ can easily be reconstructed by forward integrating the dynamics at each time step s^k , i.e.,

$$\theta^{k+1} = \theta^k + \int_{s^k}^{s^{k+1}} \frac{v_t^k}{\rho^k} ds. \quad (35)$$

5.2. Simulation results

The spacecraft is modeled with 750 kg of dry mass, 600 mN of maximum thrust, and an exhaust velocity of 40 km/s. Using an initial orbit with radius 5.1 km and tangential velocity of -2.4 m/s, we are able to compute the numerical approximation of ω using the spatial grid described in [Table 2](#) in combination with a temporal grid using $N = 1000$ grid points. The propagation of the zero level set over time is shown in [Figs. 2](#) and [3](#). Since the temporal grid for the calculation of the value function is more coarse

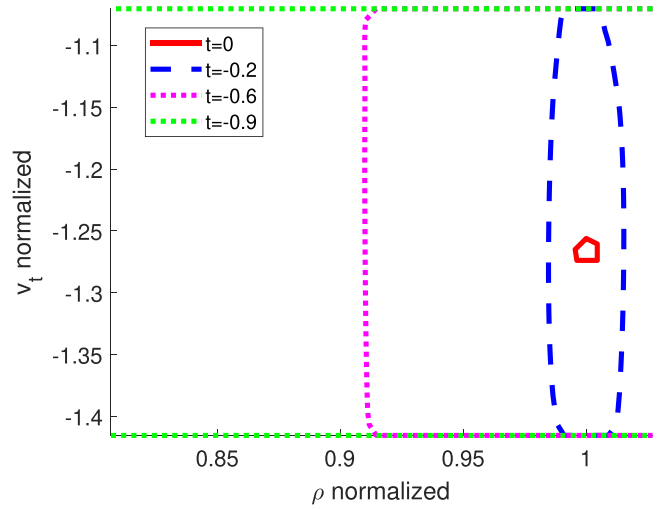


Fig. 3. The evolution of the zero level set of the value function, ω , projected in the $\rho - v_t$ plane for $v_\rho = 0$ and $\Delta m = 100$ g. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 3

Comparison of the accuracy, CPU time, and memory usage as the size of the grid changes.

	Low	Medium	High
Grid Size	[24 18 18 12]	[32 24 24 16]	[50 40 40 32]
CPU Time	20 min	44 min	9.28 h
Memory to store ω	374 MB	1181 MB	10.25 GB
$ \rho_{\text{final}} - \rho_{\text{target}} $	82.72 m	64.72 m	40.83 m
$ v_{\rho, \text{final}} - v_{\rho, \text{target}} $	$6.80e-5 \frac{m}{s}$	$4.26e-5 \frac{m}{s}$	$8.31e-7 \frac{m}{s}$
$ v_{t, \text{final}} - v_{t, \text{target}} $	$2.40e-4 \frac{m}{s}$	$4.01e-4 \frac{m}{s}$	$2.41e-4 \frac{m}{s}$

than that used for the trajectory calculation, we need to interpolate the value function while computing the final trajectory. The final computed trajectory, for an initial propellant mass of 24.89 g and transfer time 2757 s is shown in [Fig. 4](#). The asteroid rendering for [Fig. 4](#) was computed as in [Kulumani and Lee \[22\]](#).

The accuracy of the final orbit is within 41 m of the target orbit. Calculating ω took 9 h using a 3 GHz 8-Core Intel Core i7-9700 processor running Matlab with an extension of the Level Set Methods toolbox [\[13\]](#). The run time and accuracy can be significantly improved upon when using optimized code such as [Bokanowski et al. \[10\]](#) or [Chen et al. \[12\]](#). To show how the accuracy of the solution, the memory usage as well as the CPU time varies with the grid size, we recompute the value function and the trajectory with coarser spatial grids, as presented in [Table 3](#). We use only single-precision arrays to store the value function, yet utilized double-precision arrays for all numerical calculations on the value function. The subscript *final* denotes the values of the final point of the computed trajectory, i.e. $\mathbf{r}(\mathbf{0})$, while *target* refers to the target orbit used to define \mathcal{C} and subsequently initialize the computation of the value function. Once ω is computed, it is incorporated into [\(15\)](#), which is solved using Matlab's `paretosearch` function. Solving the MOC problem took 120 s and the resulting Pareto front is shown in [Fig. 5](#).

A comparison of the thrust magnitude of the smoothed optimal control policy is shown in [Fig. 6](#). As can be seen, the time-optimal solution uses near continuous thrust to reach the target, at the cost of using a large amount of fuel. The control policy of the trajectory presented in [Fig. 4](#) meanwhile, has noticeable cruising phases where no fuel is consumed. As expected, the thrust magnitude of both policies follows a bang-bang structure.

To illustrate the results of problems in Bolza form, we consider the case of optimizing the remaining propellant in oppose to the

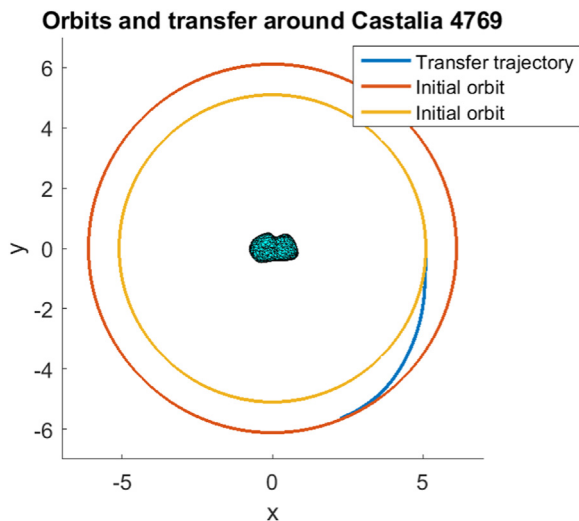


Fig. 4. Initial orbit and transfer trajectory to a circular orbit ≈ 1 km further away from the asteroid. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 4 Spatial grid configuration for the Bolza problem.

	ρ	v_ρ	v_t	Δm	z
Points	32	24	24	32	32
Spacing	0.007	0.069	0.016	0.0067	0.0067
Minimum	0.805	-0.277	-1.446	-0.0533	-0.1533
Maximum	1.029	1.317	-1.068	0.1533	0.0533

Table 5 Comparison of the accuracy, cpu time, and memory usage for the second formulation in Bolza form.

	Bolza-formulation
Grid Size	[32 24 24 32 32]
CPU Time	32 h
Memory to store ω	15.2 GB
$ \rho_{\text{final}} - \rho_{\text{target}} $	66.14 m
$ v_{\rho, \text{final}} - v_{\rho, \text{target}} $	$1.8e - 6 \frac{m}{s}$
$ v_{t, \text{final}} - v_{t, \text{target}} $	$0.0018 \frac{m}{s}$

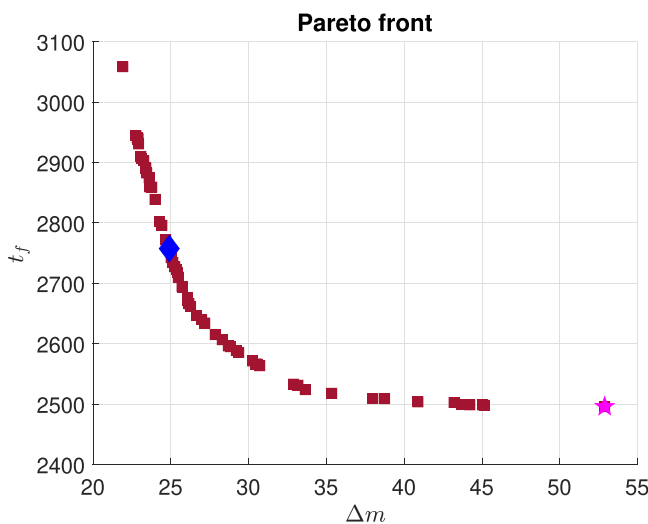


Fig. 5. Pareto front of the objective functions $J_1 = \Delta m$ in g and $J_2 = t_f$ in seconds. The trajectory in Fig. 4 is derived using the point marked by the blue diamond, while the time-optimal solution used in Fig. 6 uses the point denoted by the magenta pentagram. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

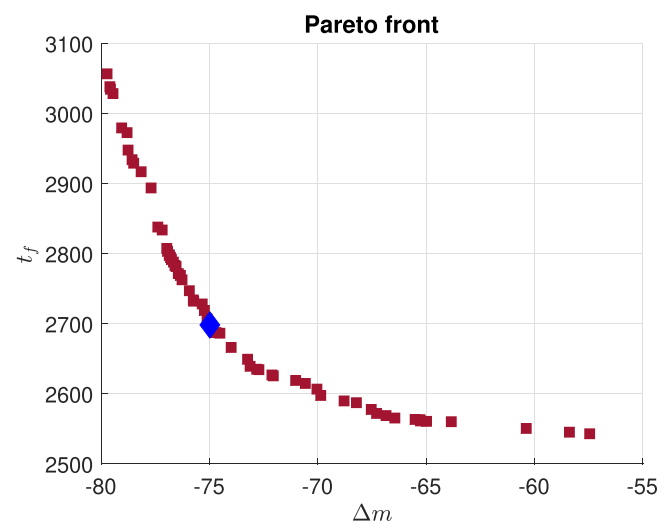
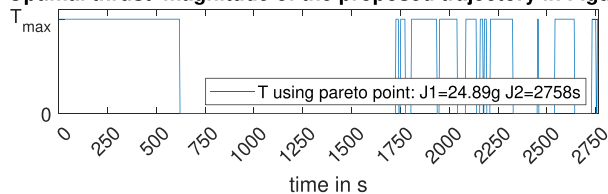


Fig. 7. Pareto front of the objective functions $J_1 = z$ in g and $J_2 = t_f$ in seconds, for the Bolza problem. The point used for trajectory calculations is marked by the blue diamond. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Optimal thrust magnitude of the proposed trajectory in Figure 4



Optimal thrust magnitude of the minimum time trajectory

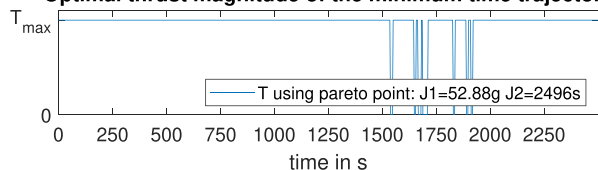


Fig. 6. A comparison of the thrust magnitude of the smoothed optimal control policy of the time-optimal solution and the trajectory presented in Fig. 4.

initial propellant. Therefore, let us fix the initial propellant to 100 g. We use a similar setup as in Section 5, with the addition of the auxiliary state, z , defined as a terminal cost. We consider the uniform spaced grid over r and z , defined in Table 4. Storing the value function using single precision requires 15.2 GB of memory. Since the objective is to maximize the remaining propellant, the optimization problem needs to minimize $-\Delta m$. Therefore, for a given final state $r_f \in \mathbb{R}^7$ and transfer time $t_f \in [0, +\infty)$, we define the cost functions as $J_1(r_f, t_f) := -\Delta m$ and $J_2(r_f, t_f) := t_f$, where Δm denotes the 7th element of the state vector r_f (the mass in our case). The 2-dimensional objective function $J : \mathbb{R}^7 \times [0, +\infty) \rightarrow \mathbb{R}^2$ can then be written as

$$J(r_f, t_f) := [J_1(r_f, t_f), J_2(r_f, t_f)]^T. \tag{36}$$

The resulting Pareto front is shown in Fig. 7. Calculating the reachable set took approximately 32 h of CPU time. Using the reachable set, calculating the Pareto front took approximately 258 s. As expected, the Pareto front looks similar to that of Fig. 5, yet not identical due to numerical inaccuracy and change in the initial mass, resulting in modified dynamics.

Using an initial propellant mass of 100 g, z of -75 g, and transfer time of 2698 s, the transfer orbit is computed in the same way as before. As expected, the transfer trajectory is similar to the tra-

jectory shown in Fig. 4, and the accuracy of the transfer orbit is within 66m of the target orbit. For comparison to the first approach, the accuracy of the second approach is shown in Table 5. From comparing the Pareto front, it can be seen that both methods have a minimum transfer time of around 2500s as well as a minimum propellant requirement of just over 20 g.

6. Conclusion

We have presented a novel method of using the value function of a quasi-variational inequality to compute the decision space of multi-objective optimization problems. The feasibility and effectiveness of the proposed approach was demonstrated by applying it to the problem of low-thrust trajectory design. The approach is applicable to arbitrary multi-objective optimization problems where the control variable is required to lie within a reachable set.

Since the Hamiltonian becomes affine with respect to the thrust magnitude, once the thrust angles have been fixed, future research aims to exploit this fact by integrating classification based approaches [33]. Furthermore, utilizing approximations of the reachable set as in Bansal and Tomlin [5], as well as decomposing the reachable sets as in Chen et al. [14], seems promising.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Proposition Appendix A.1

Proposition A.1. Under Assumption 2.2, any two trajectories \mathbf{r} and $\hat{\mathbf{r}}$ reconstructed from f , with $\mathbf{r}(-t_f) = r_0$ and $\hat{\mathbf{r}}(-t_f) = \hat{r}_0$, respectively, are such that $\|\mathbf{r}(\tau) - \hat{\mathbf{r}}(\tau)\| \leq \|r_0 - \hat{r}_0\| e^{(t_f + \tau)L_f}$ for all $\tau \in [-t_f, 0]$.

Proof. Let $r_0, \hat{r}_0 \in \mathbb{R}^7$ be two initial states, and $t_f \in [0, \infty)$. For the same t_f , we choose two trajectory control pairs $(\mathbf{r}, \mathbf{u}) \in \Pi_{r_0, t_f}$ and $(\hat{\mathbf{r}}, \hat{\mathbf{u}}) \in \Pi_{\hat{r}_0, t_f}$. Then by Carathéodory's existence of solutions [34], the following relation holds:

$$\begin{aligned} \|\mathbf{r}(-t) - \hat{\mathbf{r}}(-t)\| &\leq \|r_0 - \hat{r}_0\| + \int_{-t_f}^{-t} \|f(\mathbf{r}(s), \mathbf{u}(s)) - f(\hat{\mathbf{r}}(s), \hat{\mathbf{u}}(s))\| ds \\ &\leq \|r_0 - \hat{r}_0\| + L_f \int_{-t_f}^{-t} \|\mathbf{r}(s) - \hat{\mathbf{r}}(s)\| ds \\ &\leq \|r_0 - \hat{r}_0\| e^{(t_f - t)L_f}, \end{aligned}$$

where the second inequality is due to Assumption 2.2, while the last inequality is due to the Bellman-Gronwall Lemma [34]. \square

Appendix B. Proof of Proposition 4.9

Proof. Fix $(r_0, z_0), (\hat{r}_0, \hat{z}_0) \in \mathbb{R}^7 \times \mathbb{R}^p$, $t_f \in [0, \infty)$ and let $\epsilon > 0$. We choose $(\hat{\mathbf{r}}, \hat{\mathbf{u}}, \hat{\mathbf{z}}) \in \mathcal{Z}_{\hat{r}_0, t_f, \hat{z}_0}$, such that

$$\vartheta(\hat{r}_0, t_f, \hat{z}_0) \geq \bigvee_i [J_t^i(\hat{\mathbf{r}}(0)) - \hat{\mathbf{z}}^i(-t_f)] \bigvee v(\hat{\mathbf{r}}(0)) \bigvee \max_{s \in [-t_f, 0]} g(\hat{\mathbf{r}}(s)) - \epsilon.$$

By definition of ϑ , for any $(\mathbf{r}, \mathbf{u}) \in \Pi_{r_0, t}$, this yields the following relation

$$\begin{aligned} \vartheta(r_0, t_f, z_0) - \vartheta(\hat{r}_0, t_f, \hat{z}_0) &\leq \bigvee_i [J_t^i(\mathbf{r}(0)) - \mathbf{z}^i(-t_f)] \\ &\quad \bigvee v(\mathbf{r}(0)) \bigvee \max_{s \in [-t_f, 0]} g(\mathbf{r}(s)) \\ &\quad - \bigvee_i [J_t^i(\hat{\mathbf{r}}(0)) - \hat{\mathbf{z}}^i(-t_f)] \bigvee v(\hat{\mathbf{r}}(0)) \bigvee \max_{s \in [-t_f, 0]} g(\hat{\mathbf{r}}(s)) + \epsilon. \end{aligned}$$

Let $\kappa \in [-t_f, 0]$ be such that

$$g(\mathbf{r}(\kappa)) = \max_{s \in [-t_f, 0]} g(\mathbf{r}(s)).$$

We then have

$$\begin{aligned} \vartheta(r_0, t_f, z_0) - \vartheta(\hat{r}_0, t_f, \hat{z}_0) &\leq \bigvee_i [J_t^i(\mathbf{r}(0)) - \mathbf{z}^i(-t_f)] \\ &\quad \bigvee v(\mathbf{r}(0)) \bigvee g(\mathbf{r}(\kappa)) \\ &\quad - \bigvee_i [J_t^i(\hat{\mathbf{r}}(0)) - \hat{\mathbf{z}}^i(-t_f)] \bigvee v(\hat{\mathbf{r}}(0)) \bigvee g(\hat{\mathbf{r}}(\kappa)) + \epsilon. \end{aligned}$$

Using Proposition Appendix A.1, we distinguish the following cases.

$$\text{Case A: } g(\mathbf{r}(\kappa)) \geq \bigvee_i [J_t^i(\mathbf{r}(0)) - \mathbf{z}^i(-t_f)] \bigvee v(\mathbf{r}(0))$$

$$\begin{aligned} \vartheta(r_0, t_f, z_0) - \vartheta(\hat{r}_0, t_f, \hat{z}_0) &\leq \\ g(\mathbf{r}(\kappa)) - \bigvee_i [J_t^i(\hat{\mathbf{r}}(0)) - \hat{\mathbf{z}}^i(-t_f)] \bigvee v(\hat{\mathbf{r}}(0)) \bigvee g(\hat{\mathbf{r}}(\kappa)) + \epsilon \\ &\leq g(\mathbf{r}(\kappa)) - g(\hat{\mathbf{r}}(\kappa)) + \epsilon \leq L_g e^{(t_f + \kappa)L_f} \|r_0 - \hat{r}_0\| + \epsilon, \end{aligned}$$

where the last inequality is due to the fact that g is Lipschitz continuous.

$$\text{Case B: } v(\mathbf{r}(0)) \geq \bigvee_i [J_t^i(\mathbf{r}(0)) - \mathbf{z}^i(-t_f)] \bigvee g(\mathbf{r}(\kappa))$$

$$\begin{aligned} \vartheta(r_0, t_f, z_0) - \vartheta(\hat{r}_0, t_f, \hat{z}_0) &\leq \\ v(\mathbf{r}(0)) - \bigvee_i [J_t^i(\hat{\mathbf{r}}(0)) - \hat{\mathbf{z}}^i(-t_f)] \bigvee v(\hat{\mathbf{r}}(0)) \bigvee g(\hat{\mathbf{r}}(\kappa)) + \epsilon \\ &\leq v(\mathbf{r}(0)) - v(\hat{\mathbf{r}}(0)) + \epsilon \leq L_v e^{t_f L_f} \|r_0 - \hat{r}_0\| + \epsilon \end{aligned}$$

$$\text{Case C: } \bigvee_i [J_t^i(\mathbf{r}(0)) - \mathbf{z}^i(-t_f)] \geq g(\mathbf{r}(\kappa)) \bigvee v(\mathbf{r}(0))$$

Recall (27), then under Assumption 2.2, any two trajectories \mathbf{z} and $\hat{\mathbf{z}}$ reconstructed from J_r with z_0 and \hat{z}_0 , respectively, are bounded within a given time interval $[-t, 0]$. To see this,

$$\begin{aligned} \|\mathbf{z}(-t) - \hat{\mathbf{z}}(-t)\| &\leq \|z_0 - \hat{z}_0\| + \int_{-t}^0 \|J_r(\mathbf{r}(s), \mathbf{u}(s)) - J_r(\hat{\mathbf{r}}(s), \hat{\mathbf{u}}(s))\| ds \\ &\leq \|z_0 - \hat{z}_0\| + L_r \int_{-t}^0 \|\mathbf{r}(s) - \hat{\mathbf{r}}(s)\| ds \\ &\leq \|z_0 - \hat{z}_0\| + L_r \int_{-t}^0 \|r_0 - \hat{r}_0\| e^{(t_f + s)L_f} ds \\ &\leq \|z_0 - \hat{z}_0\| + \|r_0 - \hat{r}_0\| L_r e^{t_f L_f} \frac{1 - e^{-tL_f}}{L_f}, \end{aligned}$$

where the third inequity is due to Proposition Appendix A.1, and the last one follows by performing the integration.

Next, let $j \in [1, \dots, p]$ be such that

$$J_t^j(\mathbf{r}(0)) - \mathbf{z}^j(-t_f) = \bigvee_i [J_t^i(\mathbf{r}(0)) - \mathbf{z}^i(-t_f)].$$

Then it follows, that

$$\begin{aligned} \vartheta(r_0, t_f, z_0) - \vartheta(\hat{r}_0, t_f, \hat{z}_0) &\leq \bigvee_i [J_t^i(\mathbf{r}(0)) - \mathbf{z}^i(-t_f)] \\ &\quad - \bigvee_i [J_t^i(\hat{\mathbf{r}}(0)) - \hat{\mathbf{z}}^i(-t_f)] \bigvee v(\hat{\mathbf{r}}(0)) \bigvee g(\hat{\mathbf{r}}(\kappa)) + \epsilon \\ &\leq \bigvee_i [J_t^i(\mathbf{r}(0)) - \mathbf{z}^i(-t_f)] \\ &\quad - \bigvee_i [J_t^i(\hat{\mathbf{r}}(0)) - \hat{\mathbf{z}}^i(-t_f)] + \epsilon \\ &\leq [J_t^j(\mathbf{r}(0)) - \mathbf{z}^j(-t_f)] - [J_t^j(\hat{\mathbf{r}}(0)) - \hat{\mathbf{z}}^j(-t_f)] + \epsilon \\ &\leq [J_t^j(\mathbf{r}(0)) - J_t^j(\hat{\mathbf{r}}(0))] - [\mathbf{z}^j(-t_f) - \hat{\mathbf{z}}^j(-t_f)] + \epsilon \end{aligned}$$

By Proposition Appendix A.1 and under Assumption 4.5

$$[J_t^j(\mathbf{r}(0)) - J_t^j(\hat{\mathbf{r}}(0))] \leq L_t \|r_0 - \hat{r}_0\| e^{t_f L_f}.$$

Finally, this yields the relationship

$$\begin{aligned} & \left[J_t^j(\mathbf{r}(0)) - J_t^j(\hat{\mathbf{r}}(0)) \right] - \left[\mathbf{z}^j(-t_f) - \hat{\mathbf{z}}^j(-t_f) \right] + \epsilon \\ & \leq \left\| z_0 - \hat{z}_0 \right\| + \left\| r_0 - \hat{r}_0 \right\| \left[L_t e^{t_f L_f} + L_r \frac{e^{t_f L_f} - 1}{L_f} \right] + \epsilon. \end{aligned}$$

Thus in every case, on an interval $[0, t_f]$ there exists a set of constants C_r and C_z , such that

$$\vartheta(r_0, t_f, z_0) - \vartheta(\hat{r}_0, t_f, \hat{z}_0) \leq C_r \left\| r_0 - \hat{r}_0 \right\| + C_z \left\| z_0 - \hat{z}_0 \right\| + \epsilon$$

The same argument conducted with (r_0, t_f, z) and $(\hat{r}_0, t_f, \hat{z})$ reversed establishes that

$$\vartheta(\hat{r}_0, t_f, \hat{z}_0) - \vartheta(r_0, t_f, z_0) \leq C_r \left\| r_0 - \hat{r}_0 \right\| + C_z \left\| z_0 - \hat{z}_0 \right\| + \epsilon.$$

Since ϵ is arbitrary, we conclude that

$$\left| \vartheta(\hat{r}_0, t_f, \hat{z}_0) - \vartheta(r_0, t_f, z_0) \right| \leq C_r \left\| r_0 - \hat{r}_0 \right\| + C_z \left\| z_0 - \hat{z}_0 \right\|,$$

thus concluding the proof. \square

The proof for ω is similar to that of ϑ and will therefore be omitted in the interest of space.

Appendix C. Derivation of the optimal thrust angles

Notice that since the applied thrust, T , is always positive, the term

$$(q_4 \cos \alpha + \sin \alpha (q_5 \sin \delta + q_6 \cos \delta))$$

in the Hamiltonian can be minimized separately from T .

To this end, we introduce the auxiliary variables

$$\chi(\delta) := \sqrt{q_5^2 + q_6^2} \cos(\delta - \arctan \frac{q_5}{q_6}),$$

$$A(\delta) := \sqrt{q_4^2 + \chi(\delta)^2},$$

using the trigonometric identity

$$a \cos x + b \sin x = R \cos(x - \arctan \frac{b}{a}),$$

with $R = \sqrt{a^2 + b^2}$. First optimizing over α , and subsequently over δ (notice that this sequential minimization is exact since $A(\delta) \geq 0$) results in

$$\begin{aligned} & \min_{\alpha, \delta \in [-\pi, \pi] \times [-\frac{\pi}{2}, \frac{\pi}{2}]} (q_4 \cos \alpha + \sin \alpha (q_5 \sin \delta + q_6 \cos \delta)) \\ & = \min_{\delta \in [-\frac{\pi}{2}, \frac{\pi}{2}]} A(\delta) \min_{\alpha \in [-\pi, \pi]} \cos(\alpha - \arctan \frac{\chi(\delta)}{q_4}). \end{aligned}$$

Thus, the optimal thrust angles are given by

$$\alpha^*(\delta) := \pm\pi + \arctan \frac{\chi(\delta)}{q_4}.$$

Since $\cos(\alpha^*(\delta) - \arctan \frac{\chi(\delta)}{q_4}) = -1$, after applying $\alpha^*(\delta)$, it follows that

$$\delta^* \in \arg \min_{\delta \in [-\frac{\pi}{2}, \frac{\pi}{2}]} A(\delta) = \arctan \frac{q_5}{q_6} \pm \pi.$$

Subsequently,

$$q_4 \cos \alpha^* + \sin \alpha^* (q_5 \sin \delta^* + q_6 \cos \delta^*) = -\sqrt{q_4^2 + q_5^2 + q_6^2}. \quad (\text{C.1})$$

Substituting (C.1) into (17) results in (18) which depends in an affine fashion on T .

References

- [1] M.J. Abrahamson, A. Ardito, D. Han, R. Haw, B. Kennedy, N. Mastrodemos, S. Nandi, R. Park, B. Rush, A. Vaughan, Dawn orbit determination team: trajectory modeling and reconstruction processes at Vesta, *Adv. Astronaut. Sci.* 148 (February) (2013) 2271–2290.
- [2] A. Altarovic, O. Bokanowski, H. Zidani, A general Hamilton–Jacobi framework for non-linear state-constrained control problems, *ESAIM - Control, Optim. Calc. Var.* 19 (2) (2013) 337–357, doi:10.1051/cocv/2012011.
- [3] P.G. Antreasian, F.T. Nicholson, P.H. Kallemeyn, S. Bhaskaran, R.J. Haw, P. Halamek, Galileo orbit determination for the Ida encounter, *Adv. Astronaut. Sci.* 87 (2) (1994) 1027–1048.
- [4] J.P. Aubin, J. Lygeros, M. Quincampoix, S. Sastry, N. Seube, Impulse differential inclusions: a viability approach to hybrid systems, *IEEE Trans. Autom. Control* 47 (1) (2002) 2–20, doi:10.1109/9.981719.
- [5] S. Bansal, C.J. Tomlin, Deepreach: a deep learning approach to high-dimensional reachability, *CoRR abs/2011.02082*(2020).
- [6] M. Bardi, I. Capuzzo-Dolcetta, *Optimal Control and Viscosity Solutions of Hamilton–Jacobi–Bellman Equations*, Birkhäuser Basel, 1997, doi:10.1007/978-0-8176-4755-1.
- [7] O. Bokanowski, E. Bourgeois, A. Désilles, H. Zidani, Global optimization approach for the climbing problem of multi-stage launchers, 2015. Working paper or preprint, <https://hal-ensta-paris.archives-ouvertes.fr/hal-01113819>.
- [8] O. Bokanowski, E. Bourgeois, A. Désilles, H. Zidani, Payload optimization for multi-stage launchers using HJB approach and application to a SSO mission, *IFAC-PapersOnLine* 50 (1) (2017) 2904–2910, doi:10.1016/j.ifacol.2017.08.647.
- [9] O. Bokanowski, E. Bourgeois, A. Désilles, H. Zidani, Global optimization approach for the ascent problem of multi-stage launchers, in: H.G. Bock, W. Jäger, E. Kostina, H.X. Phu (Eds.), *Modeling, Simulation and Optimization of Complex Processes HPSC 201*, Springer International Publishing, Cham, 2021, pp. 1–42.
- [10] O. Bokanowski, A. Désilles, H. Zidani, J. Zhao, ROC-HJ solver, <https://uma-ensta-paris.fr/soft/ROC-HJ/>.
- [11] O. Bokanowski, N. Forcadell, H. Zidani, Reachability and minimal times for state constrained nonlinear problems without any controllability assumption, *SIAM J. Control Optim.* 48 (7) (2010) pp.4292–4316.
- [12] M. Chen, J. Fisac, C. Tomlin, Berkeley efficient API in C++ for level set methods, <https://github.com/HJReachability/beacsls>.
- [13] M. Chen, S. Herbert, S. Bansal, C. Tomlin, Optimal control helper toolbox, <https://github.com/HJReachability/helperOC>.
- [14] M. Chen, S.L. Herbert, M.S. Vashishtha, S. Bansal, C.J. Tomlin, Decomposition of reachable sets and tubes for a class of nonlinear systems, *IEEE Trans. Autom. Control* 63 (11) (2018), doi:10.1109/TAC.2018.2797194.
- [15] A. Désilles, H. Zidani, Pareto front characterization for multiobjective optimal control problems using Hamilton–Jacobi approach, *SIAM J. Control Optim.* 57 (6) (2019) 3884–3910, doi:10.1137/18M1176993.
- [16] I.J. Fialho, T.T. Georgiou, Worst case analysis of nonlinear systems, *IEEE Trans. Autom. Control* 44 (6) (1999) 1180–1196, doi:10.1109/9.769372.
- [17] J.F. Fisac, M. Chen, C.J. Tomlin, S.S. Sastry, Reach-avoid problems with time-varying dynamics, targets and constraints, in: *Proceedings of the 18th International Conference on Hybrid Systems: Computation and Control*, in: HSCC '15, Association for Computing Machinery, New York, NY, USA, 2015, p. 1120, doi:10.1145/2728606.2728612.
- [18] D.T. Greenwood, *Principles of Dynamics*, second ed., Prentice-Hall, Englewood Cliffs, N.J., 1988.
- [19] C. Hermosilla, P.R. Wolenski, H. Zidani, The mayer and minimum time problems with stratified state constraints, *Set-Valued Var. Anal.* 26 (3) (2018) 643–662, doi:10.1007/s11228-017-0413-z.
- [20] R.S. Hudson, S.J. Ostro, Shape of asteroid 4769 Castalia (1989 PB) from inversion of radar images, *Science* (1994), doi:10.1126/science.263.5149.940.
- [21] Y. Jiang, H. Baoyin, Orbital mechanics near a rotating asteroid, *J. Astrophys. Astron.* (2014), doi:10.1007/s12036-014-9259-z.
- [22] S. Kulamani, T. Lee, Low-thrust trajectory design using reachability sets near asteroid 4769 Castalia, in: *AIAA/AAS Astrodynamics Specialist Conference*, 2016, doi:10.2514/6.2016-5376.
- [23] D. Liberzon, *Calculus of Variations and Optimal Control Theory: A Concise Introduction*, Princeton University Press, 2011.
- [24] J. Lygeros, C. Tomlin, S. Sastry, Controllers for reachability specifications for hybrid systems, *Automatica* 35 (3) (1999) 349–370, doi:10.1016/S0005-1098(98)00193-9.
- [25] K. Margellos, J. Lygeros, Hamilton–Jacobi formulation for reach-avoid differential games, *IEEE Trans. Autom. Control* 56 (8) (2011) 1849–1861, doi:10.1109/TAC.2011.2105730.
- [26] K. Margellos, J. Lygeros, Toward 4-D trajectory management in air traffic control: a study based on monte carlo simulation and reachability analysis, *IEEE Trans. Control Syst. Technol.* 21 (5) (2013) 1820–1833, doi:10.1109/TCST.2012.2220773.
- [27] K. Miettinen, *Nonlinear Multiobjective Optimization*, International Series in Operations Research & Management Science, vol. 12, Springer US, Boston, MA, 1998, doi:10.1007/978-1-4615-5563-6.
- [28] I.M. Mitchell, The flexible, extensible and efficient toolbox of level set methods, *J. Sci. Comput.* 35 (2–3) (2008) 300–329, doi:10.1007/s10915-007-9174-4.
- [29] I.M. Mitchell, A.M. Bayen, C.J. Tomlin, A time-dependent Hamilton–Jacobi formulation of reachable sets for continuous dynamic games, *IEEE Trans. Autom. Control* 50 (7) (2005) 947–957, doi:10.1109/TAC.2005.851439.
- [30] S. Ober-Blöbaum, M. Ringkamp, G. zum Felde, Solving multiobjective optimal control problems in space mission design using discrete mechanics and reference point techniques, in: *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, 2012, pp. 5711–5716, doi:10.1109/CDC.2012.6426285.
- [31] S. Osher, R. Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces*, Applied Mathematical Sciences, Springer New York, 2003.
- [32] D.W. Pacher, Low-thrust orbit transfer design for Dawn operations at Vesta, *Adv. Astronaut. Sci.* 140 (818) (2011) 1213–1232.

- [33] V. Rubies-Royo, D. Fridovich-Keil, S. Herbert, C.J. Tomlin, A classification-based approach for approximate reachability, in: 2019 International Conference on Robotics and Automation (ICRA), 2019, pp. 7697–7704, doi:[10.1109/ICRA.2019.8793919](https://doi.org/10.1109/ICRA.2019.8793919).
- [34] S. Sastry, *Nonlinear Systems : Analysis, Stability, and Control, Interdisciplinary Applied Mathematics, vol. 10*, Springer, New York ; London, 1999.
- [35] D.J. Scheeres, J.K. Miller, D.K. Yeomans, The orbital dynamics environment of 433 eros: a case study for future asteroid missions, *BBA - Bioenergetics* (1987), doi:[10.1016/0005-2728\(87\)90102-2](https://doi.org/10.1016/0005-2728(87)90102-2).
- [36] D.J. Scheeres, S.J. Ostro, R.S. Hudson, R.A. Werner, Orbits close to asteroid 4769 Castalia, *Icarus* 121 (1) (1996) 67–87, doi:[10.1006/icar.1996.0072](https://doi.org/10.1006/icar.1996.0072).
- [37] W. Seefelder, Lunar Transfer Orbits Utilizing Solar Perturbations and Ballistic Capture, *Luft- und Raumfahrt, Utz, Wiss.*, 2002. https://books.google.co.uk/books?id=Nvg_vYHePt0C.
- [38] L.F. Shampine, M.W. Reichelt, The MATLAB ODE suite, *SIAM J. Sci. Comput.* 18 (1) (1997) 1–22, doi:[10.1137/S1064827594276424](https://doi.org/10.1137/S1064827594276424).
- [39] A. Shirazi, J. Ceberio, J.A. Lozano, Spacecraft trajectory optimization: a review of models, objectives, approaches and solutions, *Prog. Aerosp. Sci.* 102 (2018) 76–98, doi:[10.1016/j.paerosci.2018.07.007](https://doi.org/10.1016/j.paerosci.2018.07.007).
- [40] O. von Stryk, R. Bulirsch, Direct and indirect methods for trajectory optimization, *Ann. Oper. Res.* 37 (1992) 357–373, doi:[10.1007/BF02071065](https://doi.org/10.1007/BF02071065).
- [41] N. Vertovec, S. Ober-Blöbaum, K. Margellos, Multi-objective minimum time optimal control for low-thrust trajectory design, in: 2021 European Control Conference (ECC), 2021, pp. 1975–1980, doi:[10.23919/ECC54610.2021.9654919](https://doi.org/10.23919/ECC54610.2021.9654919).