Annual Reviews in Control 52 (2021) 282–315

Contents lists available at ScienceDirect

Annual Reviews in Control

journal homepage: www.elsevier.com/locate/arcontrol

Review

Advances in trajectory optimization for space vehicle control

Danylo Malyuta [a,*], Yue Yu [b], Purnanand Elango [a], Behçet Açıkmeşe [a]

[a] William E. Boeing Department of Aeronautics and Astronautics, University of Washington, Seattle, WA 98195, USA
[b] Oden Institute for Computational Engineering and Sciences, The University of Texas at Austin, Austin, TX 78712, USA

ARTICLE INFO

Keywords:
Optimal control
Convex optimization
Model predictive control
Trajectory optimization
Rocket ascent
Atmospheric reentry
Rocket landing
Spacecraft rendezvous
Small body landing
Attitude reorientation
Orbit transfer and injection
Interplanetary trajectory

ABSTRACT

Space mission design places a premium on cost and operational efficiency. The search for new science and life beyond Earth calls for spacecraft that can deliver scientific payloads to geologically rich yet hazardous landing sites. At the same time, the last four decades of optimization research have put a suite of powerful optimization tools at the fingertips of the controls engineer. As we enter the new decade, optimization theory, algorithms, and software tooling have reached a critical mass to start seeing serious application in space vehicle guidance and control systems. This survey paper provides a detailed overview of recent advances, successes, and promising directions for optimization-based space vehicle control. The considered applications include planetary landing, rendezvous and proximity operations, small body landing, constrained attitude reorientation, endo-atmospheric flight including ascent and reentry, and orbit transfer and injection. The primary focus is on the last ten years of progress, which have seen a veritable rise in the number of applications using three core technologies: lossless convexification, sequential convex programming, and model predictive control. The reader will come away with a well-rounded understanding of the state-of-the-art in each space vehicle control application, and will be well positioned to tackle important current open problems using convex optimization as a core technology.

1. Introduction

Improvements in computing hardware and maturing software libraries have made optimization technology become practical for space vehicle control. The term computational guidance and control (CGC) was recently coined to refer to control techniques that are iterative in nature and that rely on the onboard computation of control actions (Lu, 2017; Tsiotras & Mesbahi, 2017).

This paper surveys *optimization-based* methods, which are a subset of CGC for space vehicles. We consider applications for launchers, planetary landers, satellites, and spacecraft. The common theme across all applications is the use of an optimization problem to achieve a control objective. Generally speaking, the goal is to solve:

$$\min_{\boldsymbol{x}} J(\boldsymbol{x}) \text{ s.t.} \tag{1a}$$

$$\boldsymbol{x} \in C, \tag{1b}$$

where $J : \mathbb{R}^n \to \mathbb{R}$ is a *cost function*, $C \subseteq \mathbb{R}^n$ is a *feasible set*, and $\boldsymbol{x}$ is an n-dimensional vector of *decision variables*. Optimization is a relevant area of study for modern space vehicle control for two reasons: effectiveness of formulation, and the (emerging) existence of efficient solution methods.

To answer why an optimization formulation is effective, consider the physical and operational constraints on the tasks that recent and future space vehicles aim to perform. Future launchers and planetary landers will require advanced entry, descent, and landing (EDL) algorithms to drive down cost via reusability, or to access scientifically interesting sites (Blackmore, 2016). Instead of landing in open terrain, future landers will navigate challenging environments such as volcanic vents and jagged blades of ice (Europa Study Team, 2012; Robertson, 2017; San Martín, Lee, & Wong, 2013). Meanwhile, human exploration missions will likely be preceded by cargo delivery, requiring landings to occur in close proximity (Dwyer-Cianciolo et al., 2019). Motivated by the presence of water ice, upcoming missions to the Moon will target its south pole (NASA Science, 2019), where extreme light-dark lighting conditions call for an automated sensor-based landing (Robinson, 2018). Even for robotic missions, new onboard technology such as vision-based terrain relative navigation requires the satisfaction of challenging constraints that couple motion and sensing. Regardless of whether one achieves the lowest cost (1a) or not, optimization is indeed one of the most compelling frameworks for finding feasible solutions in the presence of challenging constraints (Tsiotras & Mesbahi, 2017).

In orbit, foreseeable space missions will necessitate robotic docking for sample return, debris capture, and human load alleviation

* Corresponding author.
E-mail addresses: danylo@malyuta.name (D. Malyuta), yueyu@utexas.edu (Y. Yu), pelango@uw.edu (P. Elango), behcet@uw.edu (B. Açıkmeşe).

https://doi.org/10.1016/j.arcontrol.2021.04.013
Received 2 July 2020; Received in revised form 5 April 2021; Accepted 25 April 2021
Available online 2 November 2021

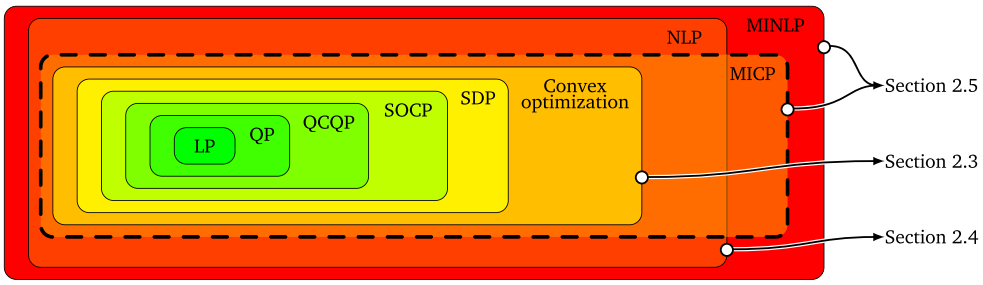


Fig. 1. Taxonomy of optimization problems. Going from inside to outside, each class becomes more difficult to solve. Roughly speaking, SOCP is currently the most general class that can be solved reliably enough to be deployed on a space system in short order.

(Woffinden & Geller, 2007). Early forms of the capability have been shown on the Japanese ETS-VII, European ATV, Russian Soyuz, US XSS-11, and US DART demonstrators. Most recently, the human-rated SpaceX Crew Dragon performed autonomous docking with the ISS, and the Orion Spacecraft is set to also feature this ability (D'Souza, Hannak, Spehar, Clark, & Jackson, 2007; Stephens et al., 2013). Further development in autonomous spacecraft rendezvous calls for smaller and cheaper sensors as well as a reduction in the degree of cooperation by the target spacecraft. This will require more flexibility in the chaser's autonomy, which is practically achievable using onboard optimization.

The above mission objectives suggest that future space vehicle autonomy will have to adeptly operate within a multitude of operational constraints. However, optimality usually stipulates operation near the boundary of the set of feasible solutions. In other words, the vehicle must activate its constraints (i.e., touch the constraint set boundary) at important or prolonged periods of its motion. By virtue of the feasible set $\mathcal{C}$ in (1b), optimization is one of the few suitable methods (and is perhaps the most natural one) to directly impose system constraints (Mayne, Rawlings, Rao, & Scokaert, 2000).

The benefit of an appropriate formulation, however, is limited if no algorithm exists to solve Problem (1) *efficiently*, which means quickly and utilizing few computational resources. Convex optimization has been a popular approach for formulating problems since it enables efficient solution methods. Fig. 1 illustrates a taxonomy of optimization problem *classes* or *families*, of which convex optimization is a part. The inner-most class in Fig. 1 is the linear program (LP). Next comes the quadratic program (QP), followed by the second-order cone program (SOCP). Troves of detail on each class may be found in many excellent optimization textbooks (Boyd & Vandenberghe, 2004; Nocedal & Wright, 1999; Rockafellar, 1970). Roughly speaking, SOCP is the most general class of problems that state-of-the-art algorithms can solve with high reliability and rigorous performance guarantees (Domahidi, Chu, & Boyd, 2013; Dueri, Açıkmeşe, Scharf, & Harris, 2017; Dueri, Zhang, & Açıkmeşe, 2014). Beyond SOCP, the semidefinite program (SDP) class enables optimization over the space of positive semidefinite matrices, which leads to many important robust control design algorithms (Boyd, Ghaoui, Feron, & Balakrishnan, 1994; Skogestad & Postlethwaite, 2005). SDP is the most general class of convex optimization for which off-the-shelf solvers are available, and many advances have been made in recent years towards more scalable and robust SDP solvers (Majumdar, Hall, & Ahmadi, 2020).

Although convex optimization can solve a large number of practical engineering problems, future space system requirements often surpass the flexibility of "vanilla" convex optimization. Solving nonconvex optimization problems will be required for many foreseeable space vehicles (Carson III et al., 2019). Thus, extending beyond SDP, we introduce three nonconvex problem classes.

First, one can abandon the convexity requirement, but retain function continuity, leading to the nonlinear program (NLP) class. Here the objective and constraint functions are continuous, albeit nonconvex. Alternatively, one could retain convexity but abandon continuity. This leads to the mixed-integer convex program (MICP) class, where binary variables are introduced to emulate discrete *switches*, such as those of valves, relays, or pulsing space thrusters (Achterberg, 2007; Achterberg & Wunderling, 2013; Dueri et al., 2017). Note that the MICP and NLP classes overlap, since some constraints admit both forms of expression. For example, the mixed-integer constraint:

$$x \in \{0\} \cup \{x \in \mathbb{R} : x \geq 1\}, \tag{2}$$

can be equivalently formulated as a nonlinear continuous constraint:

$$x(x-1) \geq 0, \;\; x \geq 0. \tag{3}$$

In the most general case, nonlinearity and discontinuity are combined to form the mixed-integer nonlinear program (MINLP) class. Since integer variables are nowhere continuous and the corresponding solution methods are of a quite different breed to continuous nonlinear programming, we reserve MINLP as the largest and toughest problem class. Algorithms for NLP, MICP, and MINLP typically suffer either from exponential complexity, a lack of convergence guarantees, or both (Malyuta & Açıkmeşe, 2020a). Nevertheless, the optimization community has had many successes in finding practical solution methods even for these most challenging problems (Achterberg & Wunderling, 2013; Szmuk, Reynolds, Açıkmeşe, Mesbahi and Carson III, 2019).

This paper stands in good company of numerous surveys on aerospace optimization. Betts (1998) presents an eloquent, albeit somewhat dated, treatise on trajectory optimization methods. Trélat (2012) provides a comprehensive survey of modern optimal control theory and indirect methods for aerospace problems, covering geometric optimal control, homotopy methods, and favorable properties of orbital mechanics that can be leveraged for trajectory optimization. Tsiotras & Mesbahi (2017) corroborate the importance of optimization in forthcoming space missions. Liu, Lu, & Pan (2017) survey the various appearances of lossless convexification and sequential convex programming in aerospace guidance methods. Eren et al. (2017) cover extensively the topic of model predictive control for aerospace applications, where Problem (1) is solved recursively to compute control actions. Mao, Dueri, Szmuk and Açıkmeşe (2018) survey three particular topics: lossless convexification, sequential convex programming, and solver customization for real-time computation. Shirazi, Ceberio, & Lozano (2018) provide a thorough discussion on the general philosophy and specific methods and solutions for in-space trajectory optimization. Recently, Song et al. (2020) surveyed optimization methods in rocket powered descent guidance with a focus on feasibility, dynamic accuracy, and real-time performance.

This paper contributes the most recent broad survey of convex optimization-based space vehicle control methods. We consider rockets for payload launch, rocket-powered planetary and small body landers, satellites, interplanetary spacecraft, and atmospheric entry vehicles. However, we do not cover some related topics like guidance of purely atmospheric vehicles (e.g., missiles and hypersonic aircraft), and control of satellite swarms, due to sufficiently unique distinctions. For a start in these areas, we refer the reader to Murillo & Lu (2010), Palumbo, Blauwkamp, & Lloyd (2010), Tewari (2011) and Zarchan

(2019) for hypersonic vehicle control, and Morgan et al. (2012), Rahmani et al. (2019), Scharf, Hadaegh, & Ploen (2003) and Tillerson, Inalhan, & How (2002) for swarm control.

From the algorithmic perspective, our focus is on convex optimization-based methods for solving the full spectrum of optimization classes in Fig. 1. The motivation for focusing on convex methods comes from the great leaps in the reliability of convex solvers and the availability of flight heritage, which gives convex optimization a technology infusion advantage for future onboard and ground-based algorithms (Blackmore, 2016; Dueri et al., 2017). We nevertheless make side references to other important, but not convex optimization-based, algorithms throughout the text. Lastly, this paper discusses algorithms at a high level, and chooses to cover a large number of applications and methods in favor of providing deep technical detail for each algorithm. The goal, in the end, is to expose the reader to dominant recent and future directions in convex optimization-based space vehicle control research.

The paper is organized as follows. Section 2 covers general theory of important optimization methods used throughout spaceflight applications. Section 3 then surveys each space vehicle control application individually. Section 3.1 surveys powered descent guidance for planetary rocket landing. Section 3.2 discusses spacecraft rendezvous and proximity operations, followed by a discussion in Section 3.3 of its close cousin, small body landing. Constrained attitude reorientation is covered in Section 3.4. Section 3.5 surveys endo-atmospheric flight, including ascent and entry. Last but not least, orbit insertion and transfer are surveyed in Section 3.6. We conclude the paper with a perspective on what lies ahead for computational guidance and control. As such, Section 4 highlights some recent applications of machine learning to select problems. This final section also tabulates some of the optimization software tooling now available for getting started in optimization methods for spaceflight applications.

Notation. Binary numbers belong to the set $\mathbb{I} \triangleq \{0,1\}$. Vectors are written in bold, such as $\boldsymbol{x} \in \mathbb{R}^n$ versus $y \in \mathbb{R}$. The identity matrix is generally written as I, and sometimes as $I_n \in \mathbb{R}^{n\times n}$ in order to be explicit about size. The zero scalar, vector, or matrix is always written as 0, with its size derived from context. The vector of ones is written as $\mathbf{1}$, with size again derived from context. Starred quantities denote optimal values, for example x^* is the optimal value of x. We use $(\boldsymbol{a};\boldsymbol{b};\boldsymbol{c})$ to concatenate elements into a column vector, like in MATLAB. The symbol $\otimes$ denotes the Kronecker matrix product or quaternion multiplication, depending on context. The positive-part function $[\boldsymbol{x}]^+ \triangleq \max\{0,\boldsymbol{x}\}$ saturates negative elements of $\boldsymbol{x}$ to zero. Given a function $f(x(t),y(t),t)$, we simplify the argument list via the shorthand $f[t]$. Throughout the paper, we interchangeably use the terms "optimization" and "programming", courtesy of linear optimization historically being used for planning military operations (Wright, 2011). When we talk about "nonlinear programming", we mean more precisely "nonconvex programming". Convexity is now known to be the true separator of efficient algorithms, however this discovery came after linear programming already established itself as the dominant class that can be efficiently solved via the Simplex method (Rockafellar, 1993). Finally, "guidance" means "trajectory generation", while "navigation" means "state estimation".

2. Background on optimization methods

This section provides a broad overview of key algorithms for space vehicle trajectory optimization. The main focus is on methods that exploit convexity, since convex optimization is where state-of-the-art solvers provide the strongest convergence guarantees at the smallest computational cost (Boyd & Vandenberghe, 2004; Nocedal & Wright, 1999).

Our algorithm overview proceeds as follows. First, Section 2.1 introduces the general continuous-time optimal control problem. Then, Section 2.2 describes how the problem is discretized to yield a finite-dimensional problem that can be solved on a computer. Following this introduction, Sections 2.3–2.6 overview important algorithms for space vehicle trajectory optimization.

2.1. Optimal control theory

Optimal control theory is the bedrock of every trajectory optimization problem (Berkovitz, 1974; Pontryagin, Boltyanskii, Gamkrelidze, & Mishchenko, 1986). The goal is to find an optimal input trajectory for the following optimal control problem (OCP):

$$\min_{t_f,\boldsymbol{u}} L_f(\boldsymbol{x}(t_f),t_f) + \int_0^{t_f} L(\boldsymbol{x}(\tau),\boldsymbol{u}(\tau),\tau)\mathrm{d}\tau \text{ s.t.} \tag{4a}$$

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t),\boldsymbol{u}(t),t),\ \forall t \in [0,t_f], \tag{4b}$$

$$\boldsymbol{g}(\boldsymbol{x}(t),\boldsymbol{u}(t),t) \le 0,\ \forall t \in [0,t_f], \tag{4c}$$

$$\boldsymbol{b}(\boldsymbol{x}(0),\boldsymbol{x}(t_f),t_f) = 0. \tag{4d}$$

In Problem (4), $\boldsymbol{x} : [0,t_f] \to \mathbb{R}^{n_x}$ is the state trajectory and $\boldsymbol{u} : [0,t_f] \to \mathbb{R}^{n_u}$ is the input trajectory, while $t_f \in \mathbb{R}$ is the final time (i.e., the trajectory duration). The state evolves according to the dynamics $\boldsymbol{f} : \mathbb{R}^{n_x}\times\mathbb{R}^{n_u}\times\mathbb{R} \to \mathbb{R}^{n_x}$, and satisfies at all times a set of constraints defined by $\boldsymbol{g} : \mathbb{R}^{n_x}\times\mathbb{R}^{n_u}\times\mathbb{R} \to \mathbb{R}^{n_c}$. At the temporal boundaries, the state satisfies conditions provided by a boundary constraint $\boldsymbol{b} : \mathbb{R}^{n_x}\times\mathbb{R}^{n_x}\times\mathbb{R} \to \mathbb{R}^{n_b}$. The quality of an input trajectory is measured by a cost function consisting of a running cost $L : \mathbb{R}^{n_x}\times\mathbb{R}^{n_u}\times\mathbb{R} \to \mathbb{R}$ and a terminal cost $L_f : \mathbb{R}^{n_x}\times\mathbb{R} \to \mathbb{R}$.

Two aspects differentiate Problem (4) from a typical parameter optimization problem. First, the constraints encode a physical process governed by ordinary differential equations (ODEs) (4b). Second, due to the continuity of time, the input trajectory has an infinite number of design parameters. This makes Problem (4) a semi-infinite optimization problem that cannot be directly implemented on a computer. In the following subsections, we provide a brief overview of two approaches for solving this problem, called the direct and indirect methods. Roughly speaking, direct methods discretize Problem (4) and solve it as a parameter optimization problem, while indirect methods attempt to satisfy the necessary conditions of optimality.

2.1.1. Indirect methods

The maximum principle, developed since the 1960s, extends the classical calculus of variations and provides a set of necessary conditions of optimality for Problem (4) (Hartl, Sethi, & Vickson, 1995; Pontryagin et al., 1986). The maximum principle has found numerous aerospace applications (Longuski, Guzmán, & Prussing, 2014).

The *indirect* family of optimization methods solves the necessary conditions of optimality, which involves a two-point boundary value problem (TPBVP) corresponding to the state and costate dynamics and their boundary conditions. Traditionally, this is solved by a single- or multiple-shooting method. One limitation of these methods is the requirement to specify in advance the time intervals over which constraint (4c) is active (Betts, 1998). Other issues that hinder onboard implementation include poor convergence stemming from a sensitivity to the initial guess, and long computation time.

Despite these challenges, the indirect approach is often the only practical solution method when aspects like numerical sensitivity and trajectory duration rule out direct methods. Low-thrust trajectory optimization, discussed in Section 3.6, is a frequent candidate for the indirect approach since the low thrust-to-weight ratios and long trajectory durations (from weeks to years) create extreme numerical challenges when formulated as a parameter optimization problem.

Most indirect methods in aerospace literature solve only the *necessary* conditions of optimality for Problem (4). However, nonlinear optimization problems can have stationary points that are not local minima, such as saddle points and local maxima. This has prompted interest in using second-order conditions of optimality to ensure that the solution is indeed a local minimum (Cesari, 1983). At the turn of the century, researchers showed how second-order information can be incorporated in orbit transfer applications (Jo & Prussing, 2000). In the last decade, further work used second-order optimality conditions for

orbit transfer and constrained attitude reorientation problems (Caillau, Daoud and Gergaud, 2012; Picot, 2012).

A promising modern indirect method family relies on homotopy in order to solve the TPBVP (Di Lizia, Armellin, Morselli, & Bernelli-Zazzera, 2013; Pan, Lu, Pan, & Ma, 2016; Pan & Pan, 2020; Pan, Pan, & Lu, 2019; Rasotto, Armellin, & Lizia, 2015; Taheri & Junkins, 2019; Trélat, 2012). Homotopy aims to address the aforementioned challenges of slow convergence, initial guess quality, and active constraint specification. The core idea is to describe the problem as a family of problems parametrized by a *homotopy parameter* $\kappa \in [0,1]$, such that the original problem is recovered for $\kappa = 1$, and the problem for $\kappa = 0$ is trivially solved. For example, consider solving a non-trivial root-finding problem:

$$\boldsymbol{F}(\boldsymbol{y}) = 0, \tag{5}$$

where $\boldsymbol{y} \in \mathbb{R}^n$ and $\boldsymbol{F} : \mathbb{R}^n \to \mathbb{R}^n$ is a smooth mapping. A (linear) homotopy method will have us define the following homotopy function:

$$\boldsymbol{\Gamma}(\boldsymbol{y}, \kappa) \triangleq \kappa \boldsymbol{F}(\boldsymbol{y}) + (1-\kappa)\boldsymbol{G}(\boldsymbol{y}) = 0, \tag{6}$$

where $\boldsymbol{G} : \mathbb{R}^n \to \mathbb{R}^n$ is a smooth function that has a known or easily computable root $\boldsymbol{y}_0 \in \mathbb{R}^n$. Popular choices are $\boldsymbol{G}(\boldsymbol{y}) = \boldsymbol{F}(\boldsymbol{y}) - \boldsymbol{F}(\boldsymbol{y}_0)$, called *Newton homotopy*, and $\boldsymbol{G}(\boldsymbol{y}) = \boldsymbol{y} - \boldsymbol{y}_0$, called *fixed-point homotopy*. In nonlinear homotopy, the function $\boldsymbol{\Gamma}(\boldsymbol{y}, \kappa)$ is a nonlinear function of κ, but otherwise similar relationships continue to hold.

The locus of points $(\boldsymbol{y}, \kappa)$ where (6) holds is called a *zero curve* of the root-finding problem. Success of the homotopy approach relies on the zero curve being continuous in κ on the interval $\kappa \in [0,1]$, albeit possibly discontinuous in $\boldsymbol{y}$. Unfortunately, the existence of such a curve is not guaranteed except for a few restricted problems (Pan, Pan, & Zhang, 2018; Watson, 2002). In general, the loci of points satisfying (6) may include bifurcations, escapes to infinity, and limit points. Furthermore, the solution at $\kappa = 1$ may not be unique.

Nevertheless, homotopy methods have been developed to successfully traverse the $\kappa \in [0,1]$ interval when a zero curve does exist. The essence of the homotopy approach is to judiciously update an intermediate solution $(\boldsymbol{y}_k, \kappa_k)$ so as to follow a κ-continuous zero curve from $\boldsymbol{y}_0$ to $\boldsymbol{y}_K$, where $\boldsymbol{\Gamma}(\boldsymbol{y}_K, 1) = \boldsymbol{F}(\boldsymbol{y}_K) = 0$ and K is the final iteration counter. At each iteration, some methods use a Newton-based root finding approach (Pan et al., 2016), while others rely solely on numerical integration (Caillau, Cots and Gergaud, 2012). For further details on the homotopy approach, we refer the reader to Pan et al. (2016).

2.1.2. Direct methods

Direct methods offer a compelling alternative where one discretizes Problem (4) and solves it as a parameter optimization problem via numerical optimization. The resulting solution in the convex case is usually very close to the optimal continuous-time one. As discussed in the next section, the solution can satisfy (4b) exactly if an exact discretization method is used (Szmuk, Reynolds, & Açıkmeşe, 2018). The optimization step is most often performed by a primal–dual interior point method (IPM), for which a considerable software ecosystem now exists thanks to 40 years of active development (Forsgren, Gill, & Wright, 2002; Nocedal & Wright, 1999; Wright, 2005). Some of this software is listed in Section 4.1.

Thanks to this expansive software ecosystem, and the large research community actively working on numerical optimization algorithms, direct methods may be considered as the most popular approach today. Their ability to "effortlessly" handle constraints like (4c) makes them particularly attractive (Betts, 1998). In the remainder of this paper, our main focus is on direct methods that use convex optimization.

Nevertheless, as mentioned in the previous section, indirect methods are still relevant for problems that exhibit peculiarities such as extreme numerical sensitivity. It must further be emphasized that some of the best modern algorithms have resulted from the *combined* use of an indirect and a direct approach. Typically an indirect method, and in particular the necessary conditions of optimality, can be used to discover the solution structure, which informs more efficient customized algorithm design for a direct solution method. We will discuss how this "fusion" approach was taken for powered descent guidance and atmospheric entry applications in Sections 2.3 and 3.5.2 respectively. Last but not least, the maximum principle can sometimes be used to find the analytic globally optimal solution for problems where no direct method can do so (e.g., nonlinear problems). In this case, an indirect method can provide a reference solution against which one can benchmark a direct method's performance (Reynolds, Malyuta, Mesbahi, Açıkmeşe and Carson III, 2020; Sundström & Guzzella, 2009). In summary, indirect and direct methods play complementary roles: the former is a good ground-truth and analysis tool, while the latter is preferred for real-time onboard implementation.

2.2. Discretization

To be solvable on a computer, the semi-infinite Problem (4) must generally be reduced to a finite-dimensional problem. This is done by the process of *discretization*, where the goal is to convert the differential constraint (4b) into a finite-dimensional algebraic constraint. This is especially important for the family of direct methods discussed in Section 2.1.2, which rely on discretization to solve Problem (4) as a parameter optimization problem.

Generally, discretization is achieved by partitioning time into a grid of N nodes and fixing a basis for the state signal, the control signal, or both (Malyuta et al., 2019). The following subsections discuss three popular approaches: an exact discretization based on zeroth-order hold (Section 2.2.2), an approximate discretization based on the classic Runge–Kutta method (Section 2.2.3), and a pseudospectral discretization that is either global or adaptive (Section 2.2.4). We frame the discussion in terms of three salient features: (1) sparsity of the discrete representation of the dynamics (4b), (2) the mechanics of obtaining a continuous-time trajectory from the discrete representation, and (3) the connection, if any, between the discrete solution and the optimal costates of the original Problem (4) derived via the maximum principle. Our goal is to give the reader enough insight into discretization to appreciate the algorithmic choices for spaceflight applications in Section 3. For a more thorough discussion, we defer to the specialized papers (Agamawi & Rao, 2020; Betts, 1998, 2010; Conway, 2011; Kelly, 2017; Malyuta et al., 2019; Phogat, Chatterjee, & Banavar, 2018a; Rao, 2010; Ross & Karpenko, 2012).

2.2.1. Example dynamical system

To ground our coverage of discretization in a concrete application, let us restrict (4b) to a linear time-invariant (LTI) system of the form:

$$\dot{\boldsymbol{x}}(t) = \tilde{A}\boldsymbol{x}(t) + \tilde{B}\boldsymbol{u}(t). \tag{7}$$

Discretization for (7) is easily generalized to handle linearized dynamics of a nonlinear system (like (4b)) about a reference state-input trajectory $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{u}}) : \mathbb{R} \to \mathbb{R}^{n_x} \times \mathbb{R}^{n_u}$. To do so, replace $\tilde{A}$ and $\tilde{B}$ with:

$$\tilde{A}(t) = \nabla_{\boldsymbol{x}} \bar{\boldsymbol{f}}[t], \tag{8a}$$

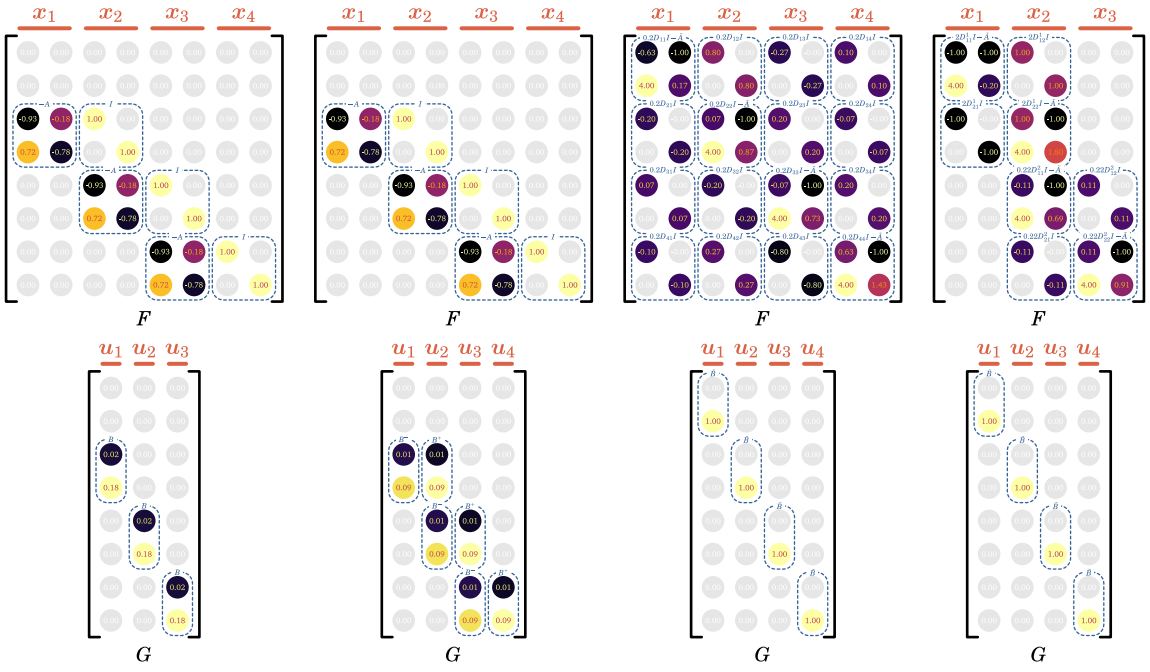
$$\tilde{B}(t) = \nabla_{\boldsymbol{u}} \bar{\boldsymbol{f}}[t], \tag{8b}$$

$$\bar{\boldsymbol{f}}[t] = \boldsymbol{f}(\bar{\boldsymbol{x}}(t), \bar{\boldsymbol{u}}(t), t), \tag{8c}$$

and add a residual term $\boldsymbol{r}(t) = \bar{\boldsymbol{f}}[t] - \tilde{A}(t)\bar{\boldsymbol{x}}(t) - \tilde{B}(t)\bar{\boldsymbol{u}}(t)$ to the right-hand side of (7). Note that in this case, (7) generally becomes a linear time-varying (LTV) system. While the zeroth-order hold method as presented in Section 2.2.2 requires linear dynamics, the Runge–Kutta and pseudospectral methods in their full generality can in fact handle the unmodified nonlinear dynamics (4b).

As a particular example of (7), we will consider a simple mass–spring–damper system with $n_x = 2$ states and $n_u = 1$ input:

$$\ddot{r}(t) + 2\zeta\omega_n \dot{r}(t) + \omega_n^2 r(t) = m^{-1} f(t), \tag{9}$$



(a) ZOH (16). (b) RK4 (22). (c) Global CGL (28). (d) Adaptive CGL (30).

Fig. 2. Matrix sparsity patterns for the linear dynamics equation $FX = GU$ for the mass–spring–damper system in Section 2.2.1 using each of the discretization methods discussed in Section 2.2. A salient feature of the ZOH and RK4 methods is their relative sparsity compared to pseudospectral methods. The adaptive collocation in (d) increases sparsity by subdividing time into multiple intervals within which separate collocations are applied, and which are coupled only through continuity at their interface (in this figure, at x_2). The non-zero element colors merely serve to visually separate the elements by their value (larger values correspond to warmer colors). (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

where r is the position, m is the mass, ζ is the damping ratio, ω_n is the natural frequency, and f is the force (input). We set $m = 1$ kg, $\zeta = 0.2$, and $\omega_n = 2$ rad s^{-1}. Furthermore, consider a staircase input signal where $f(t) = 1$ for $t \in [0, t_{step})$ and $f(t) = 0$ for $t \geq t_{step}$. We shall use $t_{step} = 1$ s. The initial condition is $r(0) = \dot{r}(0) = 0$. The simulation source code for this example is publicly available.[1]

The dynamics (9) can be written in the form (7) by using the state $\boldsymbol{x} = (r; \dot{r}) \in \mathbb{R}^2$, the input $u = f \in \mathbb{R}$, and the Jacobians:

$$\tilde{A} = \begin{bmatrix} 0 & 1 \\ -\omega_n^2 & -2\zeta\omega_n \end{bmatrix}, \quad \tilde{B} = \begin{bmatrix} 0 \\ m^{-1} \end{bmatrix}. \tag{10}$$

2.2.2. Zeroth-order hold

Zeroth-order hold (ZOH) is a discretization method that assumes the input to be a staircase signal on the temporal grid. ZOH is called an exact discretization method because, if the input satisfies this staircase property, then the discrete-time system state will exactly match the continuous-time system state at the temporal grid nodes. In practice, ZOH is a highly relevant discretization type because off-the-shelf actuators in most engineering domains, including spaceflight, output staircase commands (Scharf, Açıkmeşe, Dueri, Benito, & Casoliva, 2017).

Optimization routines that use ZOH typically consider a uniform temporal grid, although the method generally allows for arbitrarily distributed grid nodes:

$$t_k = \frac{k-1}{N-1} t_f, \ k = 1, \dots, N. \tag{11}$$

The input trajectory is then reduced to a finite number of inputs $\boldsymbol{u}_k \in \mathbb{R}^{n_u}$, $k = 1, \dots, N-1$, that define the aforementioned staircase signal:

$$\boldsymbol{u}(t) = \boldsymbol{u}_k, \ \forall t \in [t_k, t_{k+1}), \ k = 1, \dots, N-1. \tag{12}$$

[1] Visit https://github.com/dmalyuta/arc_2020_code.

It then becomes possible to find the explicit update equation for the state across any $[t_k, t_{k+1}]$ time interval, using standard linear systems theory (Antsaklis & Michel, 2006):

$$\boldsymbol{x}_{k+1} = A\boldsymbol{x}_k + B\boldsymbol{u}_k, \tag{13a}$$

$$A = \Phi(\Delta t_k, 0), \ B = A \int_0^{\Delta t_k} \Phi(\tau, 0)^{-1} \mathrm{d}\tau \tilde{B}, \tag{13b}$$

where $\Delta t_k = t_{k+1} - t_k$, $\boldsymbol{x}_k \equiv \boldsymbol{x}(t_k)$ and $\Phi(\cdot, t_k) : \mathbb{R} \to \mathbb{R}^{n_x \times n_x}$ is the *state transition matrix*. Since we assumed the system to be LTI, we have $\Phi(t, t_k) = \mathrm{e}^{\tilde{A}(t-t_k)}$ where e is the matrix exponential. If the system is LTV, the state transition matrix can be computed by integrating the following ODE:

$$\dot{\Phi}(t, t_k) = \tilde{A}(t)\Phi(t, t_k), \ \Phi(t_k, t_k) = I. \tag{14}$$

As it was said before, ZOH is an exact discretization method if the input behaves according to (12). The reason behind this becomes clear by inspecting (13), which *exactly* propagates the state from one time step to the next. This is different from forward Euler discretization, where the update is:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \Delta t_k(\tilde{A}\boldsymbol{x}_k + \tilde{B}\boldsymbol{u}_k), \tag{15}$$

For a general non-staircase input signal, however, there is a subtle connection between ZOH and forward Euler discretization. The former does a zeroth-order hold on the input signal, and integrates the state exactly, while the latter does a zeroth-order hold on the output signal (i.e., the time derivative of the system state). Thus, unlike in forward Euler discretization, state propagation for ZOH discretization cannot diverge for a stable system.

The incremental update Eq. (13a) can be written in "stacked form" to expose how the discrete dynamics are in fact a system of linear equations. To begin, note that writing (13a) for $k = 1, \dots, N-1$ is mathematically equivalent to:

$$\boldsymbol{X} = \left[\begin{array}{c|c} I_{n_x} \ 0 & 0 \\ \hline I_{N-1} \otimes A & 0 \end{array}\right] \boldsymbol{X} + \left[\begin{array}{c} 0 \\ \hline I_{N-1} \otimes B \end{array}\right] \boldsymbol{U} \triangleq \mathsf{A}\boldsymbol{X} + \mathsf{B}\boldsymbol{U}, \tag{16}$$

where $\boldsymbol{X} = (\boldsymbol{x}_1; \boldsymbol{x}_2; \dots; \boldsymbol{x}_N) \in \mathbb{R}^{Nn_x}$ is the stacked state, $\boldsymbol{U} = (\boldsymbol{u}_1; \boldsymbol{u}_2; \dots; \boldsymbol{u}_{N-1}) \in \mathbb{R}^{(N-1)n_u}$ is the stacked input, and $\otimes$ denotes the Kronecker product. Zeros in (16) denote blocks of commensurate dimensions. Clearly, we can then write the discrete dynamics as:

$$F\boldsymbol{X} = G\boldsymbol{U} \text{ where } F \triangleq I - \mathsf{A},\ G \triangleq \mathsf{B}. \tag{17}$$

The sparsity pattern for (17) using the mass–spring–damper system (9) with $N = 4$ and $t_f = 0.6$ s is shown in Fig. 2a. Both F and G consist largely of zeros, which has important consequences for customizing optimization routines that exploit this sparsity to speed up computation (Dueri et al., 2017; Malyuta et al., 2019).

An initial value problem (IVP) using a fixed $\boldsymbol{x}_1$ can be solved either by recursively applying (13a), or by solving (17):

$$\boldsymbol{X}_{2:} = F_{2:}^{\dagger}\left(-F_1\boldsymbol{x}_1 + G\boldsymbol{U}\right), \tag{18}$$

where F_1 represents the first n_x columns of F, $F_{2:}$ represents the remaining columns, and $\boldsymbol{X}_{2:} = (\boldsymbol{x}_2; \dots; \boldsymbol{x}_N)$. We use the left pseudoinverse of $F_{2:}$ and note that the solution is unique since $F_{2:}$ has a trivial nullspace. Fig. 3 shows an example of applying (13a) to the mass–spring–damper system (9) using $t_f = 10$ s and $N = 51$. Because the input step at $t_{step} = 1$ s falls exactly at a time node, the discretization is exact.

To connect ZOH discretization back to Problem (4), it is now possible to write the problem as a finite-dimensional nonlinear parameter optimization:

$$\min_{t_f, U} L_f(\boldsymbol{x}_N, t_f) + \frac{t_f}{N-1}\sum_{k=1}^{N-1} L(\boldsymbol{x}_k, \boldsymbol{u}_k, t_k) \text{ s.t.} \tag{19a}$$

$$F\boldsymbol{X} = G\boldsymbol{U}, \tag{19b}$$

$$\boldsymbol{g}(\boldsymbol{x}_k, \boldsymbol{u}_k, t_k) \leq 0,\ \forall k = 1, \dots, N-1, \tag{19c}$$

$$\boldsymbol{b}(\boldsymbol{x}_1, \boldsymbol{x}_N, t_N) = 0. \tag{19d}$$

A few remarks are in order about Problem (19). First, the constraint (19b) exactly satisfies the original dynamics (4b) under the ZOH assumption. Second, the optimal solution is only approximately optimal for Problem (4) due to an inexact running cost integration in (19a) and a finite $(N-1)$-dimensional basis with which the ZOH input is constructed. Third, the path constraints (4c) are satisfied pointwise in time via (19c), in other words there is a possibility of inter-sample constraint violation, which can nevertheless sometimes be avoided (Açıkmeşe, Scharf, Blackmore, & Wolf, 2008). Finally, in many important special cases Problem (19) is convex, and a multitude of algorithms exploit this fact, as shall be seen throughout this paper.

2.2.3. Runge–Kutta discretization

The classic Runge–Kutta (RK4) discretization method can be viewed as an advanced version of the forward Euler method (15). Unlike ZOH, which explicitly assumes a staircase input signal, RK4 is a general numerical integration method that can be applied to any state and control signals. As such, it is an inexact discretization method like forward Euler, albeit a much more accurate one. In particular, if we use the uniform temporal grid (11), then the accumulated RK4 integration error shrinks as $\mathcal{O}(N^{-4})$, whereas forward Euler integration error shrinks at the much slower rate $\mathcal{O}(N^{-1})$ (Betts, 2010; Butcher, 2016).

Directly from the definition of the RK4 method, we can write the following state update equation:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \Delta t_k(\boldsymbol{k}_1 + 2\boldsymbol{k}_2 + 2\boldsymbol{k}_3 + \boldsymbol{k}_4)/6, \tag{20a}$$

$$\boldsymbol{k}_1 = \tilde{A}\boldsymbol{x}_k + \tilde{B}\boldsymbol{u}_k, \tag{20b}$$

$$\boldsymbol{k}_2 = \tilde{A}(\boldsymbol{x}_k + 0.5\Delta t_k\boldsymbol{k}_1) + \tilde{B}\boldsymbol{u}_{k+1/2}, \tag{20c}$$

$$\boldsymbol{k}_3 = \tilde{A}(\boldsymbol{x}_k + 0.5\Delta t_k\boldsymbol{k}_2) + \tilde{B}\boldsymbol{u}_{k+1/2}, \tag{20d}$$

$$\boldsymbol{k}_4 = \tilde{A}(\boldsymbol{x}_k + \Delta t_k\boldsymbol{k}_3) + \tilde{B}\boldsymbol{u}_{k+1}, \tag{20e}$$

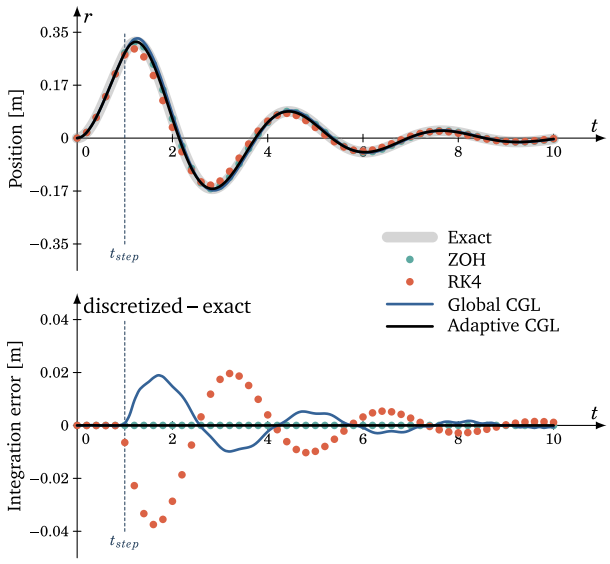


Fig. 3. Simulation of a mass–spring–damper system using four discretization types, whose matrix sparsity patterns are shown in Fig. 2. In each case, the simulation final time is $t_f = 10$ s and a temporal grid of $N = 51$ is used. The pseudospectral simulations are drawn as continuous curves by using (24) to recover a continuous-time trajectory from a finite set of states at the temporal grid nodes.

where $\boldsymbol{u}_{k+1/2} = 0.5(\boldsymbol{u}_k + \boldsymbol{u}_{k+1})$. By reshuffling terms in (20), we can write the state update in a similar form to (13a):

$$\boldsymbol{x}_{k+1} = A\boldsymbol{x}_k + B^-\boldsymbol{u}_k + B^+\boldsymbol{u}_{k+1}, \tag{21}$$

where $\{A, B^-, B^+\}$ are constructed from $\{I, \tilde{A}, \tilde{B}\}$ according to (20). Taking inspiration from (16), (21) can be written in stacked form:

$$\boldsymbol{X} = \left[\begin{array}{c|c} \begin{matrix} I_{n_x} & 0 \\ I_{N-1} \otimes A & \end{matrix} & \begin{matrix} 0 \\ 0 \end{matrix} \end{array}\right]\boldsymbol{X} + \left[\begin{array}{c} 0 \\ \hline \left(I_{N-1} \otimes [B^- \ B^+]\right)E \end{array}\right]\boldsymbol{U}, \tag{22}$$

where the matrix E combines columns in order to share the input values $\boldsymbol{u}_k$ for $2 \leq k \leq N-1$ (i.e., the internal time grid nodes):

$$E = \text{blkdiag}\left\{I_{n_u}, I_{N-2} \otimes \begin{bmatrix} I_{n_u} \\ I_{n_u} \end{bmatrix}, I_{n_u}\right\}. \tag{23}$$

Unlike ZOH, RK4 actually uses the input value at the Nth time node, hence there is one extra degree-of-freedom (DoF) leading to a slightly larger stacked input, $\boldsymbol{U} = (\boldsymbol{u}_1; \boldsymbol{u}_2; \dots; \boldsymbol{u}_N) \in \mathbb{R}^{Nn_u}$. By defining A and B according to (22), the discrete dynamics take the same form as (17). In this case, the sparsity pattern for the same mass–spring–damper example, using $N = 4$ and $t_f = 0.6$ s, is shown in Fig. 2b. Like ZOH, RK4 yields a sparse representation of the dynamics.

Like ZOH, an IVP using the discretized dynamics can be solved either by recursively applying (21), or via a pseudoinverse like (18). An example is shown in Fig. 3. Clearly, RK4 is an inexact discretization method. In this case, the interpolated input $\boldsymbol{u}_{k+1/2}$ in (20c)–(20d) is erroneous just after the input step at $t_{step} = 1$ s. Increasing the grid resolution will quickly decrease the integration error, at the expense of a larger linear system (17).

When discretized with RK4, Problem (4) looks much like Problem (19), except $\boldsymbol{u}_N$ is an extra decision variable and the user may also choose RK4 to integrate the running cost in (4a). However, a subtle but very important difference is that the solution to the discretized problem generally no longer *exactly* satisfies the original continuous-time dynamics. Thus, although RK4 may be computationally slightly cheaper than ZOH (especially for LTV dynamics, since it does not require computing integrals like (13b)), it is used less often than ZOH or pseudospectral methods discussed next.

2.2.4. Pseudospectral discretization

A key property of ZOH discretization from Section 2.2.2 is that it does not parametrize the state signal. As a result, numerical integration is required to recover the continuous-time state trajectory from the solution of Problem (19). In trajectory optimization literature, this is known as explicit simulation or *time marching* (Rao, 2010). An alternative to this approach is to approximate the state trajectory upfront by a function that is generated from a finite-dimensional basis of polynomials:

$$\boldsymbol{x}(t) = \sum_{i=1}^{N} \boldsymbol{x}_i \phi_i\big(\tau(t)\big),\ t \in [0, t_f], \quad \phi_i(\tau) \triangleq \prod_{\substack{j=1 \\ j \neq i}}^{N} \frac{\tau - \tau_j}{\tau_i - \tau_j}, \tag{24}$$

where $\tau = 2t_f^{-1}t - 1$ and $\phi_i : [-1, 1] \to \mathbb{R}$ are known as Lagrange interpolating polynomials of degree $N-1$. Note that the polynomials are defined on a normalized time interval. Since the temporal mapping is bijective, we can equivalently talk about either t or τ.

Given a temporal grid, the Lagrange interpolating polynomials satisfy an *isolation property*: $\phi_i(\tau_i) = 1$ and $\phi_i(\tau_j) = 0$ for all $j \neq i$. Hence, the basis coefficients $\boldsymbol{x}_i$ correspond exactly to trajectory values at the temporal grid nodes. Moreover, the trajectory at all other times is known *automatically* thanks to (24). This is known as implicit simulation or *collocation*. In effect, solving for the N trajectory values at the temporal grid nodes is enough to recover the complete (approximate) continuous-time trajectory. Because (24) approximates the state trajectory over the entire $[0, t_f]$ interval, this technique is known as a *global* collocation.

Collocation conditions are used in order to make the polynomial obtained from (24) behave according to the system dynamics (7):

$$\dot{\boldsymbol{x}}(t_j) = 2t_f^{-1} \sum_{i=1}^{N} \boldsymbol{x}_i \phi_i'\big(\tau(t_j)\big) = \tilde{A}\boldsymbol{x}(t_j) + \tilde{B}\boldsymbol{u}(t_j),\ \forall j \in C, \tag{25}$$

where the prime operator denotes differentiation with respect to τ (i.e., $\mathrm{d}\phi_i/\mathrm{d}\tau$) and $C \subseteq \{1, \dots, N\}$ is the set of collocation points (Malyuta et al., 2019). Note that we have already seen a disguised form of (25) earlier for the RK4 method. In particular, the well-known (20b)–(20e) are essentially collocation conditions.

According to the Stone–Weierstrass theorem (Boyd, 1989), (24) approximates a smooth signal with arbitrary accuracy as N is increased. To avoid the so-called *Runge's divergence phenomenon*, time is discretized according to one of several special non-uniform distributions, known as *orthogonal collocations* (de Boor & Swartz, 1973). In this scheme, the grid nodes τ_k are chosen to be the roots of a polynomial that is a member of a family of orthogonal polynomials. For example, Chebyshev–Gauss–Lobatto (CGL) orthogonal collocation places the scaled temporal grid nodes at the roots of $(1-\tau)^2 c_{N-1}'(\tau) = 0$, where $c_N(\tau) = \cos(N \arccos(\tau))$ is a Chebyshev polynomial of degree N. This particular collocation admits an explicit solution:

$$\tau_k = -\cos\left(\frac{k-1}{N-1}\pi\right),\ k = 1, \dots, N. \tag{26}$$

A *pseudospectral method* is a discretization scheme that approximates the state trajectory using (24), and selects a particular orthogonal collocation for the collocation points (Kelly, 2017; Rao, 2010; Ross & Karpenko, 2012). In fact, the choice of collocation points is so crucial that flavors of pseudospectral methods are named after them (e.g., the CGL pseudospectral method). Given this choice, if the dynamics and control are smooth, the approximation (24) will converge spectrally (i.e., at an exponential rate in N) to the exact state trajectory (Rao, 2010).

Associated with any C is a differentiation matrix $D \in \mathbb{R}^{|C| \times N}$ such that $D_{ji} = \phi_i'(\tau_j)$. Some collocations (e.g., CGL) admit an explicit differentiation matrix, while for others the matrix can be efficiently computed to within machine rounding error via barycentric Lagrange interpolation (Berrut & Trefethen, 2004). Having D available allows us to write the collocation conditions (25) in stacked form:

$$(2t_f^{-1} D \otimes I_{n_x})\boldsymbol{X} = (I_{|C|} \otimes \tilde{A})\boldsymbol{X} + (I_{|C|} \otimes \tilde{B})\boldsymbol{U}, \tag{27}$$

where the stacked state and input have the same dimensions as in RK4. We may thus write the discrete dynamics in the form of (19b) by defining:

$$F = 2t_f^{-1} D \otimes I_{n_x} - I_{|C|} \otimes \tilde{A}, \tag{28a}$$

$$G = I_{|C|} \otimes \tilde{B}. \tag{28b}$$

The sparsity pattern for the mass–spring–damper example, using $N = 4$ and $t_f = 10$ s, is shown in Fig. 2c. This time, due to F the dynamics constraint is not sparse. This has historically been a source of computational difficulty and a performance bottleneck for pseudospectral discretization-based optimal control (Malyuta et al., 2019; Sagliano, 2019).

Unlike for ZOH and RK4, where an IVP can be solved by recursively applying an update equation, pseudospectral methods require solving (27) *simultaneously*, which yields the state values at the temporal grid nodes all at once. In general, the solution is once again obtained via the pseudoinverse (18). However, some pseudospectral methods such as Legendre–Gauss (LG) and Legendre–Gauss–Radau (LGR) produce a square and invertible $F_{2:}$ (furthermore, $F_{2:}^{-1}F_1 = \mathbf{1} \otimes I_{n_x}$). This can be used to write (19b) in an "integral form" that has certain computation advantages (Françolin, Benson, Hager, & Rao, 2014):

$$\boldsymbol{X}_{2:} = -(\mathbf{1} \otimes I_{n_x})\boldsymbol{x}_1 + F_{2:}^{-1}G\boldsymbol{U}. \tag{29}$$

Returning to our example of the mass–spring–damper system, Fig. 3 shows a simulation using CGL collocation. Like RK4, pseudospectral discretization is an inexact method, and only approaches exactness as N grows large. In this case, the method struggles in particular due to the discontinuous nature of the input signal, which steps from one to zero at $t_{step} = 1$ s. The control trajectory is not smooth due to this discontinuity, hence the aforementioned spectral convergence guarantee does not apply. Indeed, it takes disproportionately more grid nodes to deal with this discontinuity, than if we were to subdivide the simulation into two segments $t \in [0, t_{step})$ and $t \in [t_{step}, t_f]$, where the pre-discontinuity input applies over the first interval and the post-discontinuity input applies over the second interval (Darby, Hager, & Rao, 2010).

This idea is at the core of so-called *adaptive*, or *local*, collocation methods (Darby et al., 2010; Koeppen, Ross, Wilcox, & Proulx, 2019; Sagliano, 2019; Zhao & Shang, 2018). These methods use schemes such as *hp*-adaptation (h and p stand for segment width and polynomial degree, respectively) in order to search for points like t_{step} and to subdivide the $[0, t_f]$ interval into multiple segments according to an error criterion. We defer to the above papers for the description of these adaptation schemes. For our purposes, suppose that a partition of $[0, t_f]$ into S segments is available. The ℓth segment has a basis of N_ℓ polynomials, a set of collocation points C_ℓ, and is of duration $t_{s,\ell}$ such that $\sum_{\ell=1}^{S} t_{s,\ell} = t_f$. Each segment has its own version of (28):

$$F_\ell = 2t_{s,\ell}^{-1} D^\ell \otimes I_{n_x} - I_{|C_\ell|} \otimes \tilde{A}, \tag{30a}$$

$$G_\ell = I_{|C_\ell|} \otimes \tilde{B}, \tag{30b}$$

where the newly defined F_ℓ is not to be confused with the earlier F_1 and $F_{2:}$ matrices. The new matrices in (30) are now combined to write a monolithic dynamics constraint (19b). Doing so is straightforward for the input, which can be discontinuous across segment interfaces:

$$G = \mathrm{blkdiag}\{G_1, \dots, G_S\}. \tag{31}$$

The state trajectory, however, must remain continuous across segment interfaces. To this end, the same coefficient $\boldsymbol{x}_i$ is used in (24) for both the final node of segment ℓ, and the start node of segment $\ell + 1$. Understanding this, we can write:

$$F = \mathrm{blkdiag}\{F_1, \dots, F_S\}E, \tag{32}$$

where E combines the columns of F in a similar way to (23). The net result is that the final n_x columns of F_ℓ sit above the first n_x columns of $F_{\ell+1}$.

An example of the sparsity pattern for an adaptive collocation scheme is shown in Fig. 2d using $N_1 = N_2 = 2$, $t_{s,1} = t_{step} = 1$ s, and $t_f = 10$ s. One can observe that a second benefit of adaptive collocation is that it results in a more sparse representation of the dynamics. This helps to improve optimization algorithm performance (Darby et al., 2010; Sagliano, 2019).

Solving an IVP with adaptive collocation works in the same way as for global collocation. An example is shown in Fig. 3, where two segments are used with the split occurring exactly at t_{step}. In this manner, two instances of (24) are used to approximate the state trajectory, which is smooth in the interior of both temporal segments. As such, the approximation is extremely accurate and, for practical purposes, may be considered exact in this case.

A final, and sometimes very important, aspect of pseudospectral discretization is that certain collocation schemes yield direct correspondence to the maximum principle costates of the original optimal control problem (Problem (4)). This is known as the *covector mapping theorem* (Gong, Ross, Kang, & Fahroo, 2007). One example is the integral form (29) for LG and LGR collocation (Françolin et al., 2014). Roughly speaking, the Lagrange multipliers of the corresponding parameter optimization Problem (19) can be mapped to the costates of Problem (4). Note that this requires approximating the running cost integral in (4a) using quadrature weights $\{w_k\}_{k=1}^N$ defined by the collocation scheme:

$$\int_0^{t_f} L(\boldsymbol{x}(\tau), \boldsymbol{u}(\tau), \tau)\mathrm{d}\tau \approx \sum_{k=1}^{N} w_k L(\boldsymbol{x}_k, \boldsymbol{u}_k, t_k). \tag{33}$$

This unique aspect of pseudospectral methods is why some of the optimal control problem solvers in Table 1 at the end of this article, such as DIDO, GPOPS-II, and SPARTAN, are listed as both direct and indirect solvers. In fact, they all solve a discretized version of Problem (4). Nevertheless, they are able to recover the maximum principle costate trajectories from the optimal solution (Patterson & Rao, 2014; Ross & Karpenko, 2012; Sagliano, 2019).

2.3. Convex optimization

We now come to the question of how to actually solve a finite-dimensional optimization problem such as Problem (19). As mentioned in the introduction, this can be done relatively reliably using well established tools if the problem is *convex*. Convexity has pervaded optimization algorithm design due to the following property. If a function is convex, global statements can be made from local function evaluations. The ramifications of this property cannot be understated, ranging from the guarantee of finding a global optimum (Rockafellar, 1970) to precise statements on the maximum iteration count (Peng, Roos, & Terlaky, 2002). For the purposes of this review, it is sufficient to keep in mind that a set $\mathcal{C} \subseteq \mathbb{R}^n$ is convex if it contains the line segment connecting any two of its points:

$$x, y \in \mathcal{C} \;\Leftrightarrow\; [x, y]_\theta \in \mathcal{C}\ \forall\theta \in [0, 1], \tag{34}$$

where $[x, y]_\theta \triangleq \theta x + (1-\theta)y$. Similarly, a function $f : \mathbb{R}^n \to \mathbb{R}$ is convex if its domain is convex and it lies below the line segment connecting any two of its points:

$$x, y \in \mathrm{dom}(f) \;\Leftrightarrow\; f([x, y]_\theta) \leq [f(x), f(y)]_\theta\ \forall\theta \in [0, 1]. \tag{35}$$

Countless resources cover the theory of convex optimization, among which are the notable books by Boyd & Vandenberghe (2004), Nocedal & Wright (1999), Rockafellar (1970). After applying a discretization technique akin to those in Section 2.2, a trajectory design convex optimization problem takes the following form (which is just another way of writing Problem (19)):

$$\boldsymbol{J}^*(t_f) = \min_{\boldsymbol{U}} J(\boldsymbol{X}, \boldsymbol{U}, t_f),\ \text{s.t.} \tag{36a}$$

$$\boldsymbol{x}_{k+1} = A_k\boldsymbol{x}_k + B_k\boldsymbol{u}_k + \boldsymbol{d}_k,\ \forall k = 1, \ldots, N-1, \tag{36b}$$

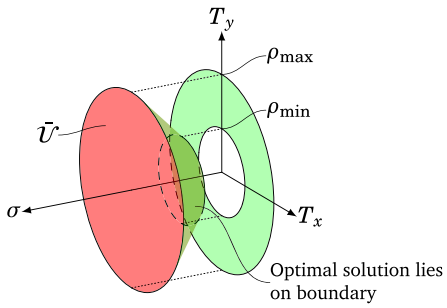


Fig. 4. Illustration of the convex relaxation technique used throughout much of lossless convexification literature for powered descent guidance. Using the maximum principle, lossless convexification proves that the optimal solution $(\boldsymbol{T}^*(t); \sigma^*(t))$ lies on the green boundary of the set $\bar{\mathcal{U}}$. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

$$\boldsymbol{g}(\boldsymbol{x}_k, \boldsymbol{u}_k, t_k) \leq 0,\ \forall k = 1, \ldots, N-1, \tag{36c}$$

$$\boldsymbol{b}(\boldsymbol{x}_1, \boldsymbol{x}_N) = 0. \tag{36d}$$

In Problem (36), $J : \mathbb{R}^{Nn_x} \times \mathbb{R}^{(N-1)n_u} \times \mathbb{R} \to \mathbb{R}$ is a convex cost function, $\boldsymbol{g} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R} \to \mathbb{R}^{n_c}$ defines a convex set of constraints, and $\boldsymbol{b} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_x} \to \mathbb{R}^{n_b}$ is an affine function defining the trajectory boundary conditions. If t_f is a decision variable, a sequence of Problem (36) instances can be solved using a line search that computes $\min_{t_f} \boldsymbol{J}^*(t_f)$ (Blackmore, Açikmeşe, & Scharf, 2010). The sequence $\{A_k, B_k, \boldsymbol{d}_k\}_{k=1}^{N-1}$ of matrices of commensurate dimensions represents the linear time-varying discretized dynamics (4b). In numerous aerospace applications, including rocket landing and spacecraft rendezvous, the dynamics are at least approximately of this form (Açıkmeşe & Ploen, 2007; de Ruiter, Damaren, & Forbes, 2013).

The path constraints (36c) are where nonconvexity appears most often for a space vehicle trajectory optimization problem. Sometimes the nonconvexity can be removed by a clever reformulation of the problem, a process called *convexification*. If the reformulation is exact, in other words the solution set is neither reduced nor expanded, the convexification is *lossless*. One example of lossless convexification that has pervaded rocket landing literature is a thrust lower-bound constraint. Let $\boldsymbol{T}(t) \in \mathbb{R}^3$ be a thrust vector, then combustion stability and engine performance dictate the following constraint:

$$\rho_{\min} \leq \|\boldsymbol{T}(t)\|_2 \leq \rho_{\max},\ \forall t \in [0, t_f]. \tag{37}$$

The lower-bound is nonconvex, but it was shown that it admits the following lossless convexification (Açıkmeşe & Ploen, 2007):

$$\rho_{\min} \leq \sigma(t) \leq \rho_{\max},\ \ \|\boldsymbol{T}(t)\|_2 \leq \sigma(t),\ \forall t \in [0, t_f]. \tag{38}$$

The convexification "lifts" the input space into an extra dimension, as illustrated in Fig. 4. Clearly the lifted feasible set $\bar{\mathcal{U}}$ is convex, and its projection onto the original coordinates contains the feasible set defined by (37). The backbone of lossless convexification is a proof via the maximum principle that the optimal solution lies on the boundary of $\bar{\mathcal{U}}$, as highlighted in Fig. 4. Thus, it can be shown that the solution using (38) is optimal for the original problem which uses (37).

Another example of lossless convexification comes from the constrained reorientation problem. Let $\boldsymbol{q}(t) \in \mathbb{R}^4$ with $\|\boldsymbol{q}(t)\|_2 = 1$ be a unit quaternion vector describing the attitude of a spacecraft. During the reorientation maneuver, it is critical that sensitive instruments on the spacecraft are not exposed to bright celestial objects. This dictates the following path constraint:

$$\boldsymbol{q}(t)^\mathsf{T} M \boldsymbol{q}(t) \leq 0,\ \forall t \in [0, t_f], \tag{39}$$

where $M \in \mathbb{R}^{4\times4}$ is a symmetric matrix that is not positive semidefinite, making the constraint nonconvex. However, when considered together

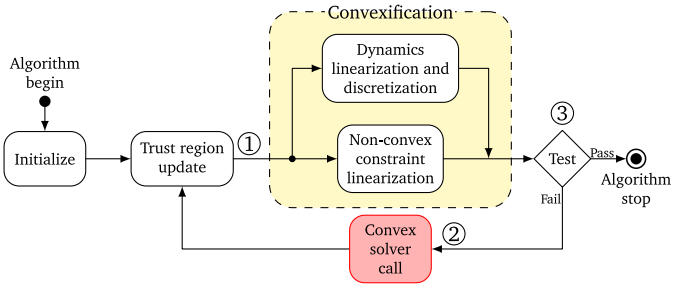


Fig. 5. Block diagram illustration of a typical SCP algorithm. The forward path can be seen as a "predictor" step, while the reverse path that calls the convex optimization solver can be seen as a "corrector" step. Although the test criterion can be guaranteed to trigger for certain SCP flavors, the solution may not be feasible for the original problem.

with the implicit constraint $\|\boldsymbol{q}(t)\|_2 = 1$, (39) can be losslessly replaced with the following convex constraint (Kim & Mesbahi, 2004):

$$\boldsymbol{q}(t)^{\mathsf{T}}(M + \mu I)\boldsymbol{q}(t) \leq \mu,\ \forall t \in [0, t_f], \tag{40}$$

where μ is chosen such that the matrix $M + \mu I$ is positive semidefinite. Instead of the maximum principle, the proof of this lossless convexification hinges on the geometry of the unit quaternion.

2.4. Sequential convex programming

Sequential convex programming (SCP) is an umbrella name for a family of nonconvex local optimization methods. It is one of many available tools alongside nonlinear programming, dynamic programming, and genetic algorithms (Floudas & Pardalos, 2009). If lossless convexification is a surgical knife to remove acute nonconvexity, SCP is a catch-all sledgehammer for nonconvex trajectory design. Clearly many aerospace problems are nonconvex, and SCP has proven to be a competitive solution method for many of them (Bonalli, Hérissé, & Trélat, 2017; Liu & Lu, 2014; Szmuk et al., 2018). This section provides intuition about how SCP algorithms work as well as their advantages and limitations. The interested reader can find further information in (Malyuta, Reynolds, Szmuk, Lew, Bonalli, Pavone, & Açıkmeşe, 2021) which provides a comprehensive tutorial on SCP.

At the core, every SCP algorithm is based on the following idea: iteratively solve a convex approximation of Problem (4), and update the approximation as new solutions are obtained. Fig. 5 provides an illustration and highlights how SCP can be thought of as a *predictor–corrector* algorithm. In the forward predictor path, the current solution is evaluated for its quality. If the quality check fails, the reverse corrector path improves the quality by solving a *subproblem* that is a better convex approximation. Examples of SCP algorithms include cases where the subproblem is linear (Palacios-Gomez, Lasdon, & Engquist, 1982), second-order conic (Mao, Szmuk, Xu and Açıkmeşe, 2018), and semidefinite (Fares, Noll, & Apkarian, 2002).

Consider Problem (36) for a simple concrete example of the SCP philosophy. Assume that $\boldsymbol{g}$ is the only nonconvex element and that t_f is fixed. At the location ① in Fig. 5, the SCP method provides an iterate in the form of a current trajectory guess $\{\bar{\boldsymbol{x}}_k, \bar{\boldsymbol{u}}_k\}_{k=1}^{N-1}$. In its most basic form, SCP linearizes and relaxes the $\boldsymbol{g}$ function:

$$\bar{\boldsymbol{g}} + \frac{\partial \bar{\boldsymbol{g}}}{\partial \boldsymbol{x}}\Delta\boldsymbol{x}_k + \frac{\partial \bar{\boldsymbol{g}}}{\partial \boldsymbol{u}}\Delta\boldsymbol{u}_k \leq \boldsymbol{\alpha}_k,\ \forall k = 1, \dots, N-1, \tag{41}$$

where $\boldsymbol{\alpha}_k \in \mathbb{R}^{n_c}$ is a *virtual buffer zone* and we define $\bar{\boldsymbol{g}} \triangleq \boldsymbol{g}(\bar{\boldsymbol{x}}_k, \bar{\boldsymbol{u}}_k, t_k)$, $\Delta\boldsymbol{x}_k \triangleq \boldsymbol{x}_k - \bar{\boldsymbol{x}}_k$, and $\Delta\boldsymbol{u}_k \triangleq \boldsymbol{u}_k - \bar{\boldsymbol{u}}_k$. The subproblem solved at location ② in Fig. 5 is then given by:

$$\min_{\substack{\boldsymbol{u}_1,\dots,\boldsymbol{u}_{N-1}\\ \boldsymbol{\alpha}_1,\dots,\boldsymbol{\alpha}_{N-1}\\ \eta_1,\dots,\eta_{N-1}}} J(\boldsymbol{X}, \boldsymbol{U}, t_f) + w_{\text{vc}}\mathbf{1}^{\mathsf{T}}\sum_{k=1}^{N-1}[\boldsymbol{\alpha}_k]^+ + w_{\text{tr}}\sum_{k=1}^{N-1}\eta_k \text{ s.t.} \tag{42a}$$

$$\boldsymbol{x}_{k+1} = A_k\boldsymbol{x}_k + B_k\boldsymbol{u}_k + \boldsymbol{d}_k,\ \forall k = 1, \dots, N-1, \tag{42b}$$

$$\bar{\boldsymbol{g}} + \frac{\partial \bar{\boldsymbol{g}}}{\partial \boldsymbol{x}}\Delta\boldsymbol{x}_k + \frac{\partial \bar{\boldsymbol{g}}}{\partial \boldsymbol{u}}\Delta\boldsymbol{u}_k \leq \boldsymbol{\alpha}_k,\ \forall k = 1, \dots, N-1, \tag{42c}$$

$$\left\|\Delta\boldsymbol{u}_k\right\| \leq \eta_k,\ \forall k = 1, \dots, N-1, \tag{42d}$$

$$\boldsymbol{b}(\boldsymbol{x}_1, \boldsymbol{x}_N) = 0. \tag{42e}$$

Problem (42) introduces several new elements. The variables η_k regulate the size of *trust regions* around the previous solution, and the weights $w_{\text{tr}},\ w_{\text{vc}} \in \mathbb{R}$ are set to large positive values that encourage convergence and zero constraint violation. The best choice of p-norm $\|\cdot\|$ in (42d) depends on the problem structure. The stopping criterion used in ③ of Fig. 5 may be, for example:

$$\max_{k\in\{1,\dots,N-1\}}\eta_k \leq \epsilon \text{ and } \max_{k\in\{1,\dots,N-1\}}\left\|[\boldsymbol{\alpha}_k]^+\right\|_\infty \leq \epsilon, \tag{43}$$

where ϵ is a user-chosen convergence tolerance constant that can be interpreted as a small "numerical error".

SCP denotes a *family* of solution methods and, as such, countless variations of Problem (42) exist. Early versions of SCP for trajectory generation focused on motion kinematics alone (Schulman et al., 2014) or included dynamics but with few convergence guarantees (Augugliaro, Schoellig, & D'Andrea, 2012). Today, a family of methods is emerging with stronger convergence guarantees, including SCvx (Mao, Szmuk et al., 2018), GuSTO (Bonalli, Cauligi, Bylard, & Pavone, 2019; Bonalli, Lew, & Pavone, 2021), and penalized trust region (PTR) (Reynolds, Malyuta, Mesbahi, Açıkmeşe and Carson III, 2020). Problem (42) exemplifies the PTR method, where the trust region sizes η_k are themselves optimization variables that are kept small using a penalty in the cost (42a). PTR is often the fastest method, but its theoretical convergence properties are relatively unexplored. In comparison, SCvx and GuSTO provide a guarantee that the algorithm converges to a locally optimal solution, albeit with potentially non-zero η_k and $\boldsymbol{\alpha}_k$. When these variables are zero, however, the solution is locally optimal for the original optimal control problem.

The main algorithmic differences across SCP methods lie in how the convex approximations are formulated, what methods are used to update the intermediate solutions and to measure progress towards optimality, and how all of this lends itself to theoretical analysis. For example, SCvx uses a discrete-time convergence proof while GuSTO uses the continuous-time maximum principle. The main difference with the PTR method is that both SCvx and GuSTO update η_k outside of the optimization problem. Interestingly, the PTR method has been observed to yield much faster convergence in practice, and a theoretical explanation recently appeared (Reynolds & Mesbahi, 2020a).

2.4.1. Related algorithms

In the general context of optimization, SCP belongs to the class of so-called trust region methods (Conn, Gould, & Toint, 2000; Nocedal & Wright, 1999). However, SCP is not to be confused with another popular trust region method, sequential quadratic programming (SQP). First of all, SCP solves its subproblems to full optimality. While this increases the number of iterations in the reverse path of Fig. 5, it vastly reduces the number of forward passes. Owing to the growing maturity of IPM solvers and the advent of solver customization (Domahidi et al., 2013; Dueri et al., 2014), iterations in the reverse path are relatively "cheap", making the trade-off a good one. Second, SCP requires only first-order problem information, since nonconvexities are handled by a simple linearization such as in (41). On the other hand, SQP is a second-order method that requires the factorization of a Hessian. This raises concerns about matrix positive semidefiniteness and may require computationally expensive techniques such as the BFGS update (Gill & Wong, 2011).

Differential dynamic programming (DDP) is another family of algorithms that, like SCP, is built around the idea of linearization (Jacobson, 1968; Mayne, 1966). More precisely, DDP solves a discrete-time optimal control problem with an additive cost function like the one in

(19a). Although DDP falls outside the scope of this survey paper, we will mention that it has major successful applications in space vehicle trajectory optimization and provides an interesting variation of the linearize-and-solve framework of Fig. 5. DDP is particularly popular for low-thrust orbit transfer trajectory optimization. For example, the NASA Mystic software used DDP for low-thrust trajectory optimization of the Dawn Discovery mission to the Vesta and Ceres protoplanets of the asteroid belt (Whiffen, 2006; Whiffen & Sims, 2001). Other appearances of DDP include multi-revolution and multi-target orbit transfer (Lantoine & Russell, 2012a, 2012b), Earth to Moon transfer with an exclusion zone (Pellegrini & Russell, 2020a, 2020b), and low-thrust flyby trajectory planning to near-Earth objects (Colombo, Vasile, & Radice, 2009).

A disadvantage of the original DDP algorithm is that it is an unconstrained optimization method. This means that while SCP naturally handles state and input constraints like (36c), implementing such constraints is still an active research area for DDP. Most attempts to incorporate constraints make use of penalty, barrier, augmented Lagrangian, and active set methods (Tassa, Mansard, & Todorov, 2014; Xie, Liu, & Hauser, 2017). Most recently, extensions of DDP were proposed to handle general nonconvex state and input constraints using a primal–dual interior point method (Aoyama, Boutselis, Patel, & Theodorou, 2020; Pavlov, Shames, & Manzie, 2020).

Another disadvantage of DDP is that it is a second-order method. Like SQP, this makes DDP more computationally expensive than SCP which only requires first-order information. Nevertheless, there is numerical evidence that DDP is faster than SQP (Lantoine & Russell, 2012b). Furthermore, related algorithms have been developed that only use first-order information, such as the iterative linear quadratic regulator (iLQR). The ALTRO software is a popular modern trajectory optimization toolbox based on the iLQR and augmented Lagrangian methods (Howell, Jackson, & Manchester, 2019).

2.5. Mixed-integer programming

Mixed-integer programming (MIP) solves problems where some decision variables are binary. Consider a concrete example in the context of Problem (4). Without loss of generality, suppose that the control input is partitioned into continuous and binary variables:

$$\boldsymbol{u}(t) = \begin{bmatrix} \boldsymbol{\nu}(t) \\ \boldsymbol{\zeta}(t) \end{bmatrix} \in \mathbb{R}^{n_u},\ \boldsymbol{\nu}(t) \in \mathbb{R}^{n_u - n_\zeta},\ \boldsymbol{\zeta}(t) \in \mathbb{I}^{n_\zeta}. \tag{44}$$

Binary variables naturally encode discrete events such as the opening of valves and relays, the pulsing of space thrusters, and mission phase transitions (Bemporad & Morari, 1999; Sun, Dai, & Lu, 2019). Furthermore, binary variables can help to approximate nonlinear gravity, aerodynamic drag, and other salient features of a space vehicle trajectory optimization problem (Blackmore, Açıkmeşe, & Carson III, 2012; Marcucci & Tedrake, 2019).

To formally discuss how MIP might be relevant for a spacecraft trajectory optimization problem like Problem (4), consider the space vehicle to be an *autonomously switched hybrid system* (Saranathan & Grant, 2018). In particular, suppose that the vehicle dynamics (4b) and its constraints (4c) are continuous except for the following extra "if-then" condition:

$$\boldsymbol{q}(\boldsymbol{z}) < 0 \ \Rightarrow\ \boldsymbol{c}(\boldsymbol{z}) = 0, \tag{45}$$

where $\boldsymbol{z} \in \mathbb{R}^{n_z}$ is some mixture of inputs, states, and time. In this formulation, the *constraint function* $\boldsymbol{c} : \mathbb{R}^{n_z} \to \mathbb{R}^{n_c}$ is activated if the *trigger function* $\boldsymbol{q} : \mathbb{R}^{n_z} \to \mathbb{R}^{n_\zeta}$ maps to the negative orthant $\mathbb{R}^{n_\zeta}_{<0}$. The conditional statement (45) can be formulated as the following set of mixed-integer constraints:

$$-\zeta_i M \le q_i(\boldsymbol{z}) \le (1-\zeta_i)M, \quad i = 1, \dots, n_\zeta, \tag{46a}$$

$$-\big(1-\sigma(\boldsymbol{\zeta})\big)M\mathbf{1} \le \boldsymbol{c}(\boldsymbol{z}) \le \big(1-\sigma(\boldsymbol{\zeta})\big)M\mathbf{1}, \tag{46b}$$

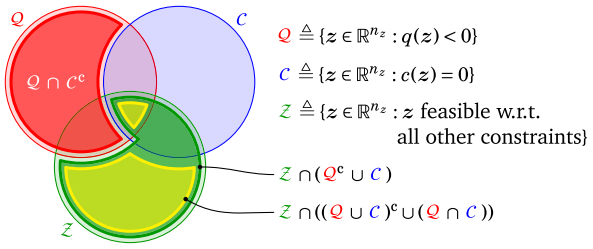


Fig. 6. A Venn diagram of the sets of solution variables $\boldsymbol{z} \in \mathbb{R}^{n_z}$ that satisfy the trigger condition $\mathcal{Q}$, constraint condition $\mathcal{C}$, and all other constraints $\mathcal{Z}$. STCs ensure that feasible solutions satisfy $\boldsymbol{z} \notin \mathcal{Q} \cap \mathcal{C}^c$ (illustrated by the red set). The feasible set with an STC is shaded green, and the feasible set with the bidirectional constraint (47) is shaded in yellow. The sets with bold outlines are padded to help visual separation only. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

$$\sigma(\boldsymbol{\zeta}) = \prod_{i=1}^{n_\zeta} \zeta_i, \tag{46c}$$

where $M \in \mathbb{R}$ is a sufficiently large positive number. The function $\sigma : \mathbb{I}^{n_\zeta} \to \mathbb{I}$ is called the *activation function*, and it imposes the if-then logic of (45) through (46a)–(46b). When the binary variable ζ_i equals one, the ith trigger is activated. Thus, when $\sigma(\boldsymbol{\zeta}) = 1$, the left-hand side of (45) holds. In fact, $\boldsymbol{q}(\boldsymbol{z}) = 0$ is also possible, but this case is irrelevant since an optimal solution will not activate the constraint function unnecessarily.

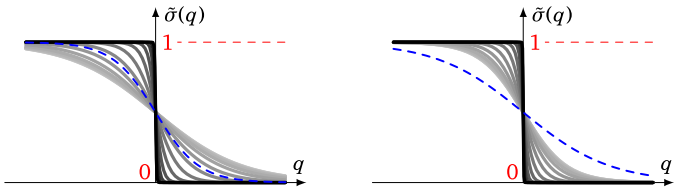
Mixed-integer programming can be used to solve Problem (4) in the presence of the constraints (46). Traditional MIP solvers are based on the branch-and-bound method (Cook, Cunningham, Pulleyblank, & Schrijver, 1998; Nemhauser & Wolsey, 1999). At their core is a divide-and-conquer logic that often, though not always, speeds up the solution process by eliminating large numbers of possible $\boldsymbol{\zeta}$ combinations. Modern MIP solvers also improve runtime through methods like pre-solving (which reduces n_ζ), cutting planes (which introduce new constraints to tighten the feasible space), heuristics, parallelism, branching variable selection, symmetry detection, and so on (Achterberg, 2007; Achterberg & Wunderling, 2013). In the worst case, however, MIP runtime remains exponential in n_ζ. This is a large hindrance to onboard implementation, since space vehicle hardware is often not able to support the large MIP computational demand (Malyuta & Açıkmeşe, 2020a; Malyuta & Açıkmeşe, 2020b; Malyuta, Reynolds, Szmuk, Açıkmeşe, & Mesbahi, 2020).

As usual in optimization, one can trade the global optimality of MIP for solution speed by accepting local optimality or by approximating the precise statement (45) with a more efficient formulation. In the following subsections, we will introduce two popular approaches that have recently emerged in both direct and indirect solution methods for solving MIP problems without introducing integer variables.

2.5.1. State-triggered constraints

State-triggered constraints (STCs) take the direct solution approach, and are under active study using the SCP framework from Section 2.4 (Malyuta et al., 2020; Szmuk et al., 2018). Roughly speaking, STCs embed the discrete if-then logic from (45) into a continuous direct formulation with minimal penalty to the solution time (Reynolds et al., 2019b). In its most basic form, an STC models (45) for the scalar case $n_\zeta = 1$ and $n_c = 1$. While there are several useful theoretical connections between STCs and the linear complementarity problem (LCP) (Cottle, Pang, & Stone, 2009), STCs encode a much larger feasible space than LCP constraints (Szmuk et al., 2018). Namely, STCs only encode forward implications, and they are *not* bidirectional statements of the following form:

$$q(\boldsymbol{z}) < 0 \ \Leftrightarrow\ c(\boldsymbol{z}) = 0. \tag{47}$$



(a) The RASHS sigmoid (53a). (b) The CSC sigmoid (53b).

Fig. 7. Comparison of the RASHS and CSC sigmoids, as defined in (53). A sweep is shown from homotopy parameter $\kappa = 1$ (lighter colors) to $\kappa = 100$ (darker colors). The blue dashed curve shows the alternative sigmoid for $\kappa = 1$ (i.e., CSC for (a) and RASHS for (b)). As κ increases, the sigmoid quickly converges to an accurate approximation of the binary activation function in (46c). While the nature of both sigmoids is similar, for a given κ the CSC sigmoid is more localized around the y-axis, and hence is a closer approximation of a step signal. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

Fig. 6 illustrates the distinction between (45) and (47). The green set denotes the feasible set with the STC, while the yellow set denotes the feasible set of the more restrictive constraint (47). Clearly, the feasible space is larger when using the STC, and this can translate into a more optimal solution.

Continuing our discussion for the scalar case, it can be shown that (45) is equivalent to either one of the following two continuous constraints:

$$q(\boldsymbol{z}) + \sigma \geq 0,\ \sigma \geq 0,\ \sigma \cdot c(\boldsymbol{z}) = 0,\ \text{or} \tag{48a}$$

$$-\min(0, q(\boldsymbol{z})) \cdot c(\boldsymbol{z}) = 0, \tag{48b}$$

where $\sigma \in \mathbb{R}$ is a slack variable that plays the role of the activation function from (46c). Although both constraints in (48) are nonconvex, they are readily ingested by the SCP linearization process described in Section 2.4.

A notable feature of STCs is that they readily extend to the multi-variable case of (45), and have been shown to handle both AND and OR combinations of triggers and constraints (Szmuk, Malyuta, Reynolds, Mceowen and Açıkmeşe, 2019; Szmuk, Reynolds et al., 2019):

$$\bigwedge_{i=1}^{n_\zeta} q_i(\boldsymbol{z}) < 0 \ \Rightarrow\ \bigwedge_{i=1}^{n_c} c_i(\boldsymbol{z}) = 0, \tag{49a}$$

$$\bigvee_{i=1}^{n_\zeta} q_i(\boldsymbol{z}) < 0 \ \Rightarrow\ \bigwedge_{i=1}^{n_c} c_i(\boldsymbol{z}) = 0, \tag{49b}$$

$$\bigwedge_{i=1}^{n_\zeta} q_i(\boldsymbol{z}) < 0 \ \Rightarrow\ \bigvee_{i=1}^{n_c} c_i(\boldsymbol{z}) = 0, \tag{49c}$$

$$\bigvee_{i=1}^{n_\zeta} q_i(\boldsymbol{z}) < 0 \ \Rightarrow\ \bigvee_{i=1}^{n_c} c_i(\boldsymbol{z}) = 0. \tag{49d}$$

In the general context of optimization, STCs do for trajectory optimization what the $\mathcal{S}$-procedure from linear matrix inequalities (LMIs) does for stability analysis and controller synthesis (Boyd et al., 1994), and what sum-of-squares (SOS) programming does to impose polynomial non-negativity over basic semialgebraic sets (Majumdar & Tedrake, 2017). That is, STCs embed an otherwise difficult logic constraint into a tractable continuous formulation. In particular, note that the scalar version of (45) can be written as:

$$c(\boldsymbol{z}) = 0 \quad \forall \boldsymbol{z} \text{ s.t. } q(\boldsymbol{z}) < 0. \tag{50}$$

On the other hand, the $\mathcal{S}$-procedure and SOS programming consider the following constraints, respectively:

$$f_0(\boldsymbol{z}) \geq 0 \quad \forall \boldsymbol{z} \text{ s.t. } f_i(\boldsymbol{z}) \geq 0,\ i = 1, \dots, p, \tag{51a}$$

$$p(\boldsymbol{z}) \geq 0 \quad \forall \boldsymbol{z} \text{ s.t. } p_{\text{eq}}(\boldsymbol{z}) = 0,\ p_{\text{ineq}}(\boldsymbol{z}) \geq 0, \tag{51b}$$

where f_i, $i = 0, \dots, p$, are quadratic functions, while p, p_{eq}, and p_{ineq} are polynomials. Comparing (50) with (51) makes the connection to STCs clear.

2.5.2. Homotopy methods

Homotopy methods, also known as *numerical continuation* schemes, were previously introduced in Section 2.1.1 in the context of solving standard optimal control problems. It turns out that homotopy can also be used to encode (45) in a continuous framework, and has been successfully embedded into recent indirect trajectory optimization algorithms. In this section, we briefly introduce the relaxed autonomously switched hybrid system (RASHS) and composite smooth control (CSC) methods (Saranathan & Grant, 2018; Arya, Taheri, & Junkins, 2020; Taheri, Junkins, Kolmanovsky, & Girard, 2020a).

To begin, let σ denote the activation function from (46c). Using the third equation from (48a), we note that (45) is exactly equivalent to the following constraint:

$$\sigma(\boldsymbol{\zeta})\boldsymbol{c}(\boldsymbol{z}) = 0. \tag{52}$$

At the core of the RASHS and CSC methods is an approximation of the binary function σ by a continuous sigmoid function $\tilde{\sigma} : \mathbb{R}^{n_\zeta} \to [0, 1]$:

$$\text{RASHS: } \tilde{\sigma}(\boldsymbol{q}) = \prod_{i=1}^{n_\zeta} \left(1 + e^{\kappa q_i}\right)^{-1}, \tag{53a}$$

$$\text{CSC: } \tilde{\sigma}(\boldsymbol{q}) = \prod_{i=1}^{n_\zeta} \frac{1}{2}\left(1 - \tanh(\kappa q_i)\right), \tag{53b}$$

where the latter equation uses the theory of hyperbolic tangent smoothing (Taheri & Junkins, 2018). The homotopy parameter $\kappa \in [0, \infty)$ regulates the accuracy of the approximation, with increasing accuracy as κ grows, such that $\lim_{\kappa \to \infty} \tilde{\sigma}(\boldsymbol{q}) = \sigma(\boldsymbol{\zeta})$. Fig. 7 illustrates how both sigmoid functions evolve as κ increases. The core idea of RASHS and CSC is to begin with a small κ where the optimal control problem is continuous and "easy" to solve, and to judiciously increase κ to such a large value that the solution becomes indistinguishable from its MIP counterpart.

It is worth noting the specific instances of (45) considered by RASHS and CSC. The former method was developed to compute time- or state-triggered multiphase trajectories, where vehicle dynamics change across phases (e.g., stage separation during rocket ascent) (Saranathan & Grant, 2018). Such a system is also known as a *differential automaton* (Tavernini, 1987). In this case, we can have m constraints of the form (45), where the kth constraint is:

$$\boldsymbol{q}^k(t) < 0 \ \Rightarrow\ \dot{\boldsymbol{x}}(t) - \boldsymbol{f}^k(\boldsymbol{x}(t), \boldsymbol{u}(t), t) = 0, \tag{54}$$

and $\boldsymbol{q}^k : \mathbb{R} \to \mathbb{R}^{n_\zeta^k}$ indicates the time interval where the kth dynamics apply. Assuming that the time intervals do not overlap, we can sum the smoothed versions of (54) to obtain a single continuous system dynamics constraint:

$$\dot{\boldsymbol{x}}(t) = \left[\sum_{k=1}^{m} \tilde{\sigma}\left(\boldsymbol{q}^k(t)\right)\right]^{-1} \sum_{k=1}^{m} \tilde{\sigma}\left(\boldsymbol{q}^k(t)\right) \boldsymbol{f}^k[t]. \tag{55}$$

Note that the new dynamics (55) are a convex combination of the individual dynamics over the m time intervals. As κ is increased, the approximation becomes more accurate, and the correct $\boldsymbol{f}^k$ functions begin to dominate their respective intervals. Moreover, using (55) instead of (54) has the algorithmic advantage of replacing a multi-point BVP with a TPBVP.

The CSC method, on the other hand, considers systems with fixed dynamics but multiple control inputs or constraints (Taheri et al., 2020a). In both cases, the overall control input can be expressed as a function of m "building block" inputs $\boldsymbol{u}^k$, such that:

$$\boldsymbol{q}^k(\boldsymbol{x}(t), \boldsymbol{u}(t), t) < 0 \ \Rightarrow\ \boldsymbol{u}(t) = \boldsymbol{u}^k(t), \tag{56}$$

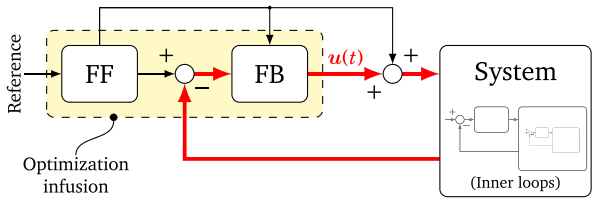


Fig. 8. A typical control architecture consists of nested layers of feedforward (FF) and feedback (FB) elements. The execution frequency increases going from the outermost to the innermost layers. In particular, elements in the FB path (highlighted in red) have stricter execution time requirements than FF elements. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

where $\boldsymbol{q}^k : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R} \to \mathbb{R}^{n_\zeta^k}$ are mutually exclusive indicators of when the kth building block input applies. Like for RASHS, the following equation provides a smooth approximation of the control, from which CSC derives its name:

$$\boldsymbol{u}(t) = \left[\sum_{k=1}^{m} \tilde{\sigma}\left(\boldsymbol{q}^k[t]\right)\right]^{-1} \sum_{k=1}^{m} \tilde{\sigma}\left(\boldsymbol{q}^k[t]\right)\boldsymbol{u}^k(t). \tag{57}$$

Clearly, RASHS, CSC, and STCs are all approaching the same problem of efficiently handling (45) from subtly different angles. It is worth noting that for the moment, both RASHS and CSC can only handle the AND combination (49a) of trigger and constraint functions. Most recently, (Malyuta & Açıkmeşe, 2021) showed that a similar homotopy framework can handle OR combinations of trigger functions. This opens up an interesting research avenue to develop a unifying homotopy method that handles all the logical combinations in (49).

2.6. Model predictive control

The preceding sections focused on solving one instance of Problem (4). We now place ourselves in the context of a control system whose architecture is illustrated in Fig. 8. Two important algorithm categories that are part of a control system are so-called *feedforward* and *feedback* (Lurie & Enright, 2000), and optimization-based methods can potentially be applied to both. In the feedback path, the current state estimate of the system is used to continually update the control signal, which means that Problem (4) must be re-solved many times. This is the core idea of model predictive control (MPC).

In its most basic form, an MPC formulation of Problem (36) can be expressed as follows:

$$\boldsymbol{u}_1^* = \underset{\boldsymbol{u}_1,\dots,\boldsymbol{u}_{N-1}}{\operatorname{argmin}} \ \boldsymbol{x}_N^\mathsf{T} Q_f \boldsymbol{x}_N + \sum_{k=1}^{N-1} \boldsymbol{x}_k^\mathsf{T} Q \boldsymbol{x}_k + \boldsymbol{u}_k^\mathsf{T} R \boldsymbol{u}_k \ \text{s.t.} \tag{58a}$$

$$\boldsymbol{x}_{k+1} = A_k \boldsymbol{x}_k + B_k \boldsymbol{u}_k + \boldsymbol{d}_k, \ \forall k = 1, \dots, N-1, \tag{58b}$$

$$\boldsymbol{g}(\boldsymbol{x}_k, \boldsymbol{u}_k, t_k) \leq 0, \ \forall k = 1, \dots, N-1, \tag{58c}$$

$$\boldsymbol{x}_1 = \hat{\boldsymbol{x}}, \ \boldsymbol{b}(\boldsymbol{x}_N) = 0. \tag{58d}$$

Fig. 9 illustrates how Problem (58) can be used to control a dynamical system. Note that Problem (58) is a parametric optimization problem because it depends on the current state estimate $\hat{\boldsymbol{x}} \in \mathbb{R}^{n_x}$. The first optimal control input $\boldsymbol{u}_1^*$ for Problem (58) becomes $\boldsymbol{u}(t)$ in Fig. 8. The weight matrices $Q \succeq 0$ and $R \succ 0$ in the running cost and the terminal weight matrix $Q_f \succeq 0$ are chosen to get a desired response. Together with the terminal constraint (58d), these choices must ensure stability and recursive feasibility in closed-loop operation (i.e., the problem must be feasible the next time that it is solved).

The main advantage of MPC is that it is arguably the most natural methodology for handling system constraints in a feedback controller (Mayne et al., 2000). However, because MPC operates in a feedback loop, stability and performance are both critical and strongly dependent on uncertainty robustness and execution frequency (Lurie & Enright, 2000; Skogestad & Postlethwaite, 2005). Troves of information have been compiled on the subject, which remains an active research area. Numerous surveys on MPC cover general and future methods (Mayne, 2014), robustness (Bemporad & Morari, 2007; García, Prett, & Morari, 1989; Mayne, 2015), computational requirements (Alessio & Bemporad, 2009), and industrial applications (Di Cairano & Kolmanovsky, 2018; Eren et al., 2017; Mao, Dueri et al., 2018; Qin & Badgwell, 2003). For space vehicle applications in particular, where onboard computation is limited, we single out so-called explicit MPC (Bemporad, Morari, Dua, & Pistikopoulos, 2002; Borrelli, Bemporad, & Morari, 2017; Rawlings, Mayne, & Diehl, 2017). The concept is to pre-compute a lookup table for the solution of Problem (58). This turns out to be possible to do exactly when the MPC problem is a QP, and approximately in more general cases up to MICP (Malyuta & Açıkmeşe, 2020a). When onboard storage and problem dimensionality permit, explicit MPC yields a much faster and computationally cheaper algorithm in which onboard optimization is replaced by a static lookup table.

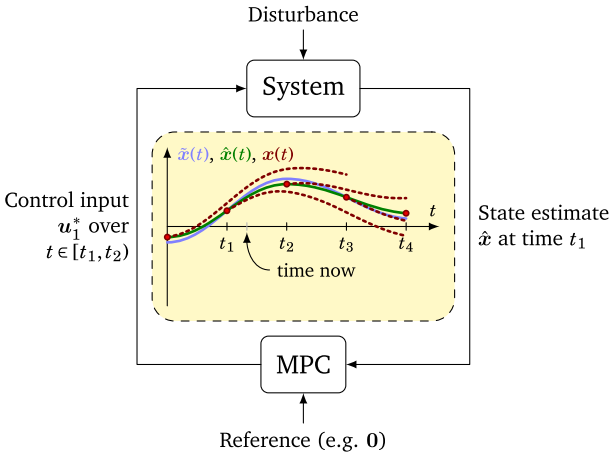


Fig. 9. Block diagram illustration of an MPC controller. At each time step t_k, MPC computes the optimal control input $\boldsymbol{u}_1^*$ by using a mathematical model of the system and solving Problem (58), which is a receding horizon optimal control problem. Note the three states drawn in the diagram: the actual state $\tilde{\boldsymbol{x}}$, the estimated state $\hat{\boldsymbol{x}}$, and the internally propagated MPC state $\boldsymbol{x}$. Each state may be slightly different due to estimation error, model uncertainty, and disturbances.

3. Applications

This section describes the application of optimization methods from the previous section to state-of-the-art space vehicle control problems. The following subsections cover the following key areas of spaceflight. Section 3.1 discusses rocket powered descent for planetary landing. Section 3.2 covers spacecraft rendezvous and Section 3.3 covers a closely related problem of small body landing. Section 3.4 talks about attitude reorientation. Endo-atmospheric ascent and entry are surveyed in Section 3.5. Last but not least, orbit transfer is discussed in Section 3.6.

3.1. Powered descent guidance for rocket landing

Powered descent guidance (PDG) is the terminal phase of EDL spanning the last few kilometers of altitude. The goal is for a lander to achieve a soft and precise touchdown on a planet's surface by using its rocket engine(s). PDG technology is fundamental for reducing cost and enabling access to hazardous yet scientifically rich sites (Braun & Manning, 2006; Carson III et al., 2019; Europa Study Team, 2012; Jones, 2018; NASA Science, 2019; Robertson, 2017; Robinson, 2018; Starek, Açıkmeşe, Nesnas, & Pavone, 2015; Steinfeldt, Grant, Matz, Braun, & Barton, 2010). The modern consensus is that iteration-based algorithms within the CGC paradigm, rather than closed-form

solutions, are required for future landers (Carson III et al., 2019; Lu, 2017). The survey of applications in this section demonstrates that optimization offers a compelling iteration-based solution method due to the availability of real-time algorithms that can enforce relevant PDG constraints.

To place state-of-the-art PDG into context, let us briefly mention some key heritage methods. Closed-form algorithms are known as *explicit guidance*, which is characterized by directly considering the targeting condition each time the guidance command is generated (Lu, 2020). Early algorithms sought closed-form solutions to versions of the following OCP:

$$\min_{t_f,\boldsymbol{a}} \int_0^{t_f} \boldsymbol{a}(t)^\mathsf{T}\boldsymbol{a}(t)\mathrm{d}t \text{ s.t.} \tag{59a}$$

$$\ddot{\boldsymbol{r}}(t) = \boldsymbol{g} + \boldsymbol{a}(t), \tag{59b}$$

$$\boldsymbol{r}(0) = \boldsymbol{r}_0,\ \dot{\boldsymbol{r}}(0) = \dot{\boldsymbol{r}}_0,\ \boldsymbol{r}(0) = \boldsymbol{r}_f,\ \dot{\boldsymbol{r}}(0) = \dot{\boldsymbol{r}}_f. \tag{59c}$$

Here, $\boldsymbol{r}(t) \in \mathbb{R}^3$ denotes position, $\boldsymbol{a}(t) \in \mathbb{R}^3$ is the acceleration control input, $\boldsymbol{g} \in \mathbb{R}^3$ is the gravitational acceleration vector and t_f is the flight duration. Position and velocity boundary values are fixed. The optimal solution to Problem (59) is known as the E-Guidance (EG) law (Cherry, 1964; D'Souza, 1997):

$$\boldsymbol{a}(t) = 6t_{\text{go}}^{-2}\boldsymbol{ZEM}(t) - 2t_{\text{go}}^{-1}\boldsymbol{ZEV}(t), \tag{60}$$

where $t_{\text{go}} \triangleq t_f - t$ is the time-to-go and:

$$\boldsymbol{ZEM}(t) \triangleq \boldsymbol{r}_f - \left(\boldsymbol{r}(t) + t_{\text{go}}\dot{\boldsymbol{r}}(t) + 0.5t_{\text{go}}^2\boldsymbol{g}\right), \tag{61a}$$

$$\boldsymbol{ZEV}(t) \triangleq \dot{\boldsymbol{r}}_f - \left(\dot{\boldsymbol{r}}(t) + t_{\text{go}}\boldsymbol{g}\right), \tag{61b}$$

are the zero-effort-miss and zero-effort-velocity terms (Furfaro, Selnick, Cupples, & Cribb, 2011; Song et al., 2020). Nominally, (60) results in an affine acceleration profile. If instead one allows the acceleration profile to be quadratic, an additional DoF appears, which is fixed by setting the terminal acceleration $\boldsymbol{a}(t_f) = \boldsymbol{a}_f$. This results in the Apollo guidance (APG) law, which flew on the historic Lunar missions (Klumpp, 1974):

$$\boldsymbol{a}(t) = 12t_{\text{go}}^{-2}\boldsymbol{ZEM}(t) - 6t_{\text{go}}^{-1}\boldsymbol{ZEV}(t) + \boldsymbol{a}_f. \tag{62}$$

The concept of an acceleration profile behind EG and APG has since been extended and generalized, resulting in a *polynomial guidance* family of algorithms. Zhang, Guo, Ma, & Zeng (2017) augment the cost (59a) with a surface collision-avoidance term. Guo, Hawkins, & Wie (2013) formulate a QP to solve for an intermediate waypoint that augments collision-avoidance capabilities and enforces actuator saturation for thrust- and power-limited engines. Lu (2019) develops a general theory for polynomial guidance laws that contains EG and APG as special cases. For one of the best modern explanations of polynomial guidance methods, the reader should consult (Lu, 2020). Unfortunately, closed-form polynomial guidance is unable to handle many operational constraints (Lu, 2018) and is not fuel optimal since the cost (59a) rather penalizes control power.

To overcome these limitations, research has long sought to characterize and eventually solve the more general fuel-optimal PDG problem. The first milestone towards fuel-optimal PDG was a closed-form single-DoF vertical descent solution (Meditch, 1964), illustrated in Fig. 12a. Evidence suggests that Apollo landings came close to this optimum (Klumpp, 1974; Mindell, 2008). The maximum principle (Pontryagin et al., 1986) played a key role back then and continues to do so in the present day.

Seeking to generalize the single-DoF result, Lawden formulated the necessary conditions of optimality for 3-DoF PDG (Lawden, 1963; Marec, 1979). However, solving the necessary conditions requires shooting methods, which are typically too computationally expensive and sensitive to the initial guess to allow efficient onboard implementation (Betts, 1998). More recently, (Topcu, Casoliva, & Mease, 2005, 2007) extended the results from (Lawden, 1963) to the case of angular velocity control, and compared the solution quality of fuel-optimal 3-DoF PDG to the necessary conditions of optimality. However, the aim of the work was not real-time onboard implementation, so nonlinear programming (SQP) was used via the GESOP solver.

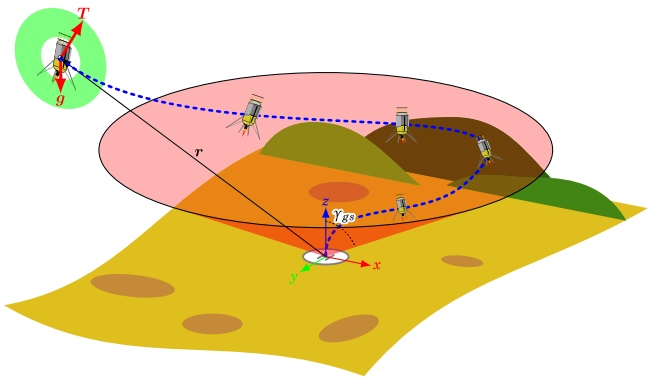


Fig. 10. Illustration of the basic powered descent guidance solved by Problem (63) via lossless convexification. The goal is to safely bring the rocket lander to standstill on the landing pad while satisfying the thrust magnitude constraints and maintaining a minimum glideslope.

After decades of research into problem *characterization*, a watershed moment for problem *solution* came in the mid 2000s with the papers (Açıkmeşe & Ploen, 2005, 2007). The authors solved the following 3-DoF PDG problem, illustrated in Fig. 10, via the process of lossless convexification described in Section 2.3:

$$\min_{t_f,\boldsymbol{T}} \int_0^{t_f} \|\boldsymbol{T}(t)\|_2\,\mathrm{d}t \text{ s.t.} \tag{63a}$$

$$\ddot{\boldsymbol{r}}(t) = \boldsymbol{g} + \boldsymbol{T}(t)m(t)^{-1}, \tag{63b}$$

$$\dot{m}(t) = -\alpha\|\boldsymbol{T}(t)\|_2, \tag{63c}$$

$$\rho_{\min} \le \|\boldsymbol{T}(t)\|_2 \le \rho_{\max}, \tag{63d}$$

$$\boldsymbol{r}(t)^\mathsf{T}\hat{\boldsymbol{e}}_z \ge \|\boldsymbol{r}(t)\|_2\cos(\gamma_{gs}), \tag{63e}$$

$$m(0) = m_0,\ \boldsymbol{r}(0) = \boldsymbol{r}_0,\ \dot{\boldsymbol{r}}(0) = \dot{\boldsymbol{r}}_0,\ \boldsymbol{r}(t_f) = 0,\ \dot{\boldsymbol{r}}(t_f) = 0. \tag{63f}$$

Unlike the classical Problem (59), Problem (63) readily handles several important operational constraints, including thrust bounds (63d) and glide slope (63e). Through numerical simulations for a prototype Mars lander, Açıkmeşe & Ploen (2007) confirmed that the optimal thrust has a max–min–max profile as shown in Fig. 11. This profile was proven to be optimal for 3-DoF PDG in Lawden (1963) and Topcu et al. (2007).

Over the course of the next decade, the method was expanded to handle fairly general nonconvex input sets (Açıkmeşe & Blackmore, 2011), minimum-error landing and thrust pointing constraints (Açıkmeşe, Carson III and Blackmore, 2013; Blackmore et al., 2010; Carson III, Açıkmeşe and Blackmore, 2011), classes of affine and quadratic state constraints (Harris & Açıkmeşe, 2013a, 2013b, 2013c, 2014), classes of nonlinear (mixed-integer) dynamics (Blackmore et al., 2012), certain binary input constraints (Harris, 2021; Malyuta & Açıkmeşe, 2020b), fixed time-of-flight problems (Kunhippurayil, Harris, & Jansson, 2021), and conservative conic obstacles (Bai, Guo, & Zheng, 2019).

The maturity of a method can be gauged by the availability of a precise statement of its limits, similar to the role played by the Bode integral in frequency-domain control (Lurie & Enright, 2000; Skogestad & Postlethwaite, 2005). Such a characterization appeared for lossless convexification in the form of constrained reachable or controllable sets (Dueri, 2018; Dueri, Raković, & Açıkmeşe, 2016; Eren, Dueri and Açıkmeşe, 2015) or "access" conditions (Song et al., 2020). These sets, obtained numerically and with arbitrarily high precision, define the boundary conditions for which versions of Problem (63) are feasible.

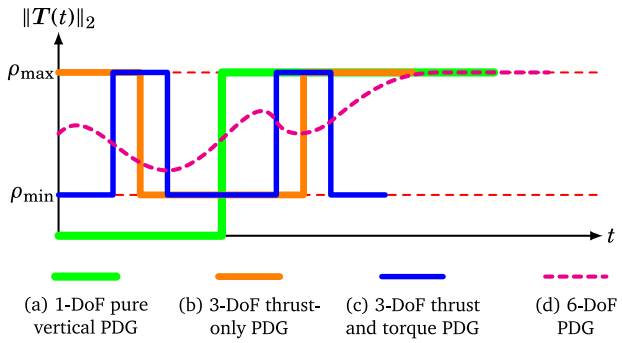


Fig. 11. Optimal thrust profiles for several powered descent guidance formulations. (a) Corresponds to the classical single-DoF result by Meditch (1964); (b) corresponds to 3-DoF translation-only landing from Açıkmeşe & Ploen (2007) and Lawden (1963); (c) corresponds to planar landing with rotation from Reynolds & Mesbahi (2020b). The thrust profile for general 6-DoF PDG with translation and rotation is an open problem. In particular, there is no theory to guarantee that it should be bang–bang, thus (d) shows a profile with no clear structure. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

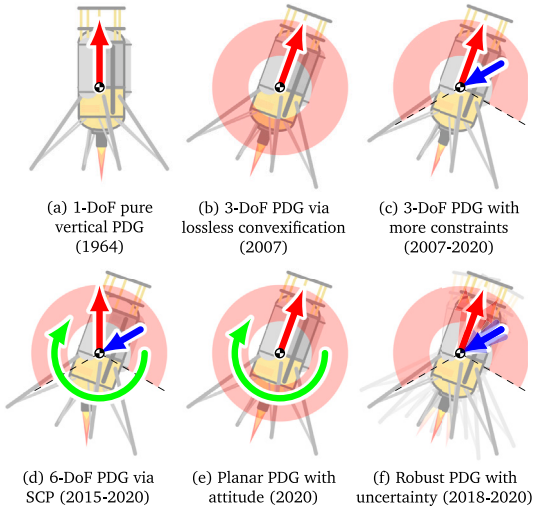


Fig. 12. Progression of PDG problem complexity. The red, green, and blue arrows denote thrust, torque, and aerodynamic force respectively. The red region denotes the feasible thrust set. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

The practicality of lossless convexification-based PDG methods was demonstrated through a series of flight tests conducted by the NASA Jet Propulsion Laboratory, Masten Space Systems, and university partners. In a 3-year 7-flight test campaign, the Masten Xombie sounding rocket demonstrated that robust onboard real-time optimization is feasible on spaceflight processors (Açıkmeşe et al., 2013; JPL and Masten Space Systems, 2012a, 2012b; Scharf et al., 2017, 2014). A number of publications accompanied this flight test campaign, including a comparison of lossless convexification to polynomial guidance (Ploen, Açıkmeşe, & Wolf, 2006), onboard computation time reduction via time-of-flight interpolation (Scharf, Ploen, & Açıkmeşe, 2015), and complete off-line lookup table generation (Açıkmeşe et al., 2008). The resulting algorithm, G-FOLD (Açıkmeşe, Blackmore and Scharf, 2013; Açıkmeşe, Casoliva, Carson III, & Blackmore, 2012), solves a tailored version of Problem (63) using a custom C-language SOCP solver called Bsocp (Dueri et al., 2014). G-FOLD is able to compute rocket landing trajectories in 100 ms on a 1.4 GHz Intel Pentium M processor. Further evidence of real-time performance was presented by Dueri et al. (2017), where Bsocp ran on a radiation-hardened BAE RAD750 PowerPC.

Despite the significant flight envelope expansion afforded by lossless convexification (Carson III, Açıkmeşe, Blackmore and Wolf, 2011; Ploen et al., 2006; Wolf, Casoliva, Manrique, Açıkmeşe, & Ploen, 2012), an inherent limitation of 3-DoF PDG is that the computed trajectory cannot incorporate attitude constraints other than those on the thrust vector, which serves as an attitude proxy. An extensive simulation campaign is required to validate the 3-DoF trajectory to be executable by a fundamentally 6-DoF lander system (Carson III et al., 2019; Kamath, Assadian, & Robinson, 2020). Thus, recent PDG research has sought 6-DoF formulations that are able to incorporate attitude dynamics and constraints.

The SCP family of methods, discussed in Section 2.4, has emerged as an effective approach to transition from a fully convex 3-DoF problem to a 6-DoF problem with some nonconvexity. Some of the popular SCP algorithms include SCvx (Mao, Szmuk et al., 2018), penalized trust region (Reynolds, Malyuta, Mesbahi, Açıkmeşe and Carson III, 2020), and GuSTO (Bonalli et al., 2019, 2021). Some other algorithms based around similar ideas have also emerged, such as ALTRO which is based on iterative LQR (Howell et al., 2019).

A vast number of flavors of SCP exist, however, since it is a nonlinear optimization technique that works best when tailored to exploit problem structure. In certain cases, lossless convexification is embedded to remove some nonlinearity. Liu (2019) convexifies an angle-of-attack (AoA) constraint relating to an aerodynamic control capability, Simplício, Marcos, & Bennani (2019) solve a version of Problem (63) in a first step and pass the solution to a second step involving SCP, while Li, Pang, Wei, Cui, & Liu (2020), Szmuk, Açıkmeşe, & Berning (2016) and Wang, Cui, & Wei (2019a) use the classical convexification result for the thrust magnitude constraint (38).

Since the mid 2010s, SCP technology enabled the expression of quadratic aerodynamic drag and thrust slew-rate constraints (Szmuk et al., 2016), attitude dynamics (Szmuk, Eren, & Açıkmeşe, 2017), variable time-of-flight (Szmuk & Açıkmeşe, 2018), and an ellipsoidal drag model that allows aerodynamic lift generation along with variable ignition time (Szmuk et al., 2018). Several papers on SCP "best practices" have also appeared, including thrust input modeling (Szmuk et al., 2017), the effect of discretization on performance (Malyuta et al., 2019), and using dual quaternions to alleviate nonconvexity in the constraints by off-loading it into the dynamics (Reynolds et al., 2019a). Practical details on real-time implementation are also available (Reynolds, Malyuta, Mesbahi, Açıkmeşe and Carson III, 2020), where the SCP solution is compared to the globally optimal trajectory for a planar landing problem (Reynolds & Mesbahi, 2020b). Most recently, a comprehensive tutorial paper with open-source code was published, and describes the algorithmic and practical aspects of SCP methods and of lossless convexification (Malyuta et al., 2021).

Fig. 12 summarizes the dominant directions of PDG development since 2005. Starting from the classical vertical-descent result by Meditch (1964), Fig. 12a, the early breakthrough for practical onboard PDG solution was achieved in 2007 by Açıkmeşe & Ploen (2007), Fig. 12b. Since then, 3-DoF PDG methods have been extended and flight tested, Fig. 12c. In particular, more complicated effects such as aerodynamic drag force were added by these extensions, which are listed in the preceding paragraph. Perhaps the biggest modern shift in PDG technology development has been to consider attitude dynamics, which is motivated by the inability to impose non-trivial attitude constraints in a 3-DoF formulation (Carson III et al., 2019). This has led to a family of so-called 6-DoF PDG algorithms, Fig. 12d, that often rely on SCP methods. To compare how close SCP comes to the global optimum, recent work found optimal solutions for "planar" PDG (Reynolds & Mesbahi, 2020b), Fig. 12e. This work restricts the landing trajectory to a 2D plane, but does include attitude dynamics. Therefore it represents both a generalization of Fig. 12b and a restriction of Fig. 12d, and provides new insight into the 6-DoF PDG optimal solution structure. Today, PDG research evolves along the following broad directions: guaranteeing real-time performance, convergence, and solution quality,

handling binary constraints, and incorporating uncertainty as shown in Fig. 12f.

One exciting development for SCP in recent years has been the advent of state-triggered constraints, introduced in Section 2.5.1. This allows real-time capable embedding of if-then logic into the guidance problem. To demonstrate the capability, Szmuk et al. (2018) imposed a velocity-triggered AoA constraint, Reynolds et al. (2019a) imposed a distance-triggered line-of-sight constraint, Szmuk, Reynolds et al. (2019) imposed a collision-avoidance constraint, and Reynolds et al. (2019b) imposed a slant-range-triggered line-of-sight constraint. In particular, the latter two works develop a theory of *compound* STCs that apply Boolean logic to combine multiple trigger and constraint functions, as shown in (49). The impact of STCs on the ability to compute solutions in real-time is discussed in Szmuk et al. (2018) and Reynolds et al. (2019b).

Simultaneously with the development of SCP for PDG, the pseudospectral discretization community has produced a rich body of work investigating the solution quality benefits of that method. Building on foundational early work (Fahroo & Ross, 2002; Garg et al., 2010; Kelly, 2017; Rao, 2010), it was demonstrated for a variant of Problem (63) that pseudospectral methods yield greater solution accuracy with fewer temporal nodes (Sagliano, 2018b). However, as discussed in Section 2.2.4, pseudospectral methods traditionally yield slower solution times because they generate non-sparse matrices for the discretized equations of motion (Malyuta et al., 2019). By using an *hp*-adaptive scheme inspired by the finite element method (Darby et al., 2010), it was shown that this can be somewhat circumvented (Sagliano, 2018a, 2019). Furthermore, it was shown that pseudospectral discretization within an SCP framework yields solutions up to 20 times faster than using sequential quadratic programming (Wang & Cui, 2018).

As deterministic PDG algorithms mature, research is becoming increasingly interested in making the trajectory planning problem robust to various sources of uncertainty. One approach is to design a feedback controller to correct for deviations from the nominal trajectory, such that the overall control input is given by:

$$\boldsymbol{u}(t) = \bar{\boldsymbol{u}}(t) + K(t)(\boldsymbol{x}(t) - \bar{\boldsymbol{x}}(t)), \tag{64}$$

where $\bar{\boldsymbol{x}}(t)$ and $\bar{\boldsymbol{u}}(t)$ are the nominal state and control respectively, and $K(t) \in \mathbb{R}^{n_u \times n_x}$ is a feedback gain matrix. In Ganet-Schoeller & Brunel (2019) and Scharf et al. (2017), the feedback controller is designed separately from the nominal trajectory. However, incorporating feedback law synthesis into the nominal trajectory generation problem can achieve more optimal solutions (Garcia-Sanz, 2019). This "simultaneous" feedback–feedforward design was done via multi-disciplinary optimization in Jiang, Li, & Tao (2018), desensitized optimal control in Shen, Seywald, & Powell (2010) and Seywald & Seywald (2019), chance-constrained optimization in Ono, Pavone, Kuwata, & Balaram (2015), and covariance steering in Ridderhof & Tsiotras (2018, 2019). Other work in this domain includes open-loop robust trajectory design via Chebyshev interval inclusion (Cheng, Wen and Jin, 2019), and *a posteriori* statistical analysis through linear covariance propagation (Woffinden, Robinson, Williams, & Putnam, 2019) and Monte Carlo simulation (Scharf et al., 2017).

PDG methods based on lossless convexification and SCP are in most cases *implicit guidance* methods. In this setup, the targeting condition (e.g., soft touchdown on the landing pad) is met by tracking a reference trajectory that yields the correct terminal state. Functionally, PDG methods are most often situated in the FF block of Fig. 8, and they generate a complete trajectory upfront that is tracked by a feedback controller. From a systems engineering perspective, this has a clear advantage of allowing heritage control methods to perform the intricate and critical control of the actual vehicle. However, it was mentioned at the start of this section and in Section 2.6 that continually re-solving for the PDG trajectory can offer additional robustness. In contrast to traditional polynomial guidance, some modern approaches aim to leverage this robustness and also satisfy system constraints via model predictive control.

Cui, Gao, & Cui (2012) show how to leverage MPC for landing with an uncertain state and variable gravitational field, while Wang et al. (2019a) show how to ensure recursive feasibility and a bounded guidance error by executing a nominal and relaxed optimization problem in parallel. In both methods, the full trajectory optimization problem is solved from the current state to the final landing location, thus the MPC horizon "shrinks" throughout the PDG maneuver. A more traditional approach is taken by Lee & Mesbahi (2017), where the prediction horizon extends for a finite duration beyond the current state. The authors also show that difficult constraints on sensor line-of-sight and spacecraft attitude are convex using a dual quaternion representation. Numerical performance of MPC for PDG on an embedded ARM platform was documented in Pascucci, Bennani, & Bemporad (2015).

3.2. Rendezvous and proximity operations

Let us now switch contexts from the final stages of planetary landing to the realm of orbital spaceflight. A key task for a spacecraft in orbit is to perform rendezvous and proximity operations (RPO). The goal is to bring an actively controlled chaser vehicle and a passively controlled target vehicle to a prescribed relative configuration, in order to achieve mission objectives such as inspection or docking. A detailed overview of RPO history and technology development can be found in Fehse (2003), Goodman (2006), Luo, Zhang, & Tang (2014) and Woffinden & Geller (2007). This section focuses on the challenges and developments in RPO using convex optimization-based solution methods.

Throughout this section we consider the following RPO trajectory optimization problem, illustrated in Fig. 13:

$$\min_{t_f, \boldsymbol{T}} \int_0^{t_f} \|\boldsymbol{T}(t)\|_2 \, \mathrm{d}t \text{ s.t.} \tag{65a}$$

$$\ddot{\boldsymbol{r}}(t) = -\mu \|\boldsymbol{r}(t)\|_2^{-3} \boldsymbol{r}(t) + \boldsymbol{T}(t) m(t)^{-1}, \tag{65b}$$

$$\dot{m}(t) = -\alpha \|\boldsymbol{T}(t)\|_2, \tag{65c}$$

$$\|\boldsymbol{T}(t)\|_2 \leq \rho, \tag{65d}$$

$$\boldsymbol{r}(t) \notin \mathcal{B}(t), \tag{65e}$$

$$\|\boldsymbol{r}(t) - \hat{\boldsymbol{r}}(t)\|_2 \cos\gamma \leq (\boldsymbol{r}(t) - \hat{\boldsymbol{r}}(t))^\mathsf{T} \boldsymbol{n}(t), \tag{65f}$$

$$m(0) = m_0, \ \boldsymbol{r}(0) = \boldsymbol{r}_0, \ \dot{\boldsymbol{r}}(0) = \dot{\boldsymbol{r}}_0, \tag{65g}$$

$$\boldsymbol{r}(t_f) = \hat{\boldsymbol{r}}(t_f), \ \dot{\boldsymbol{r}}(t_f) = \dot{\hat{\boldsymbol{r}}}(t_f), \tag{65h}$$

where $\boldsymbol{r}(t), \ \hat{\boldsymbol{r}}(t) \in \mathbb{R}^3$ denote the positions of the chaser and target spacecraft in the inertial frame. The basic objective in (65a) is to minimize fuel consumption (Park, Kolmanovsky, & Sun, 2013). Other choices include sparsification of the control sequence (Hartley, Gallieri, & Maciejowski, 2013), trading off flight duration with fuel consumption (Hu, Xie, & Liu, 2018), encouraging smoothness of the control sequence (Li & Zhu, 2018a), and reducing the sensitivity to sensing and control uncertainties (Jin, Geller, & Luo, 2020). We note that Problem (65) only characterizes the last phase of RPO. The reader is referred to Hartley, Trodden, Richards, & Maciejowski (2012) and Sun et al. (2019) for examples of multi-phase RPO trajectory optimization.

Since the quantity of interest in RPO is the relative motion between the chaser and the target, it is commonplace to express the dynamics (65b) in a different reference frame. Examples include the local-vertical local-horizontal (LVLH) frame centered at the target, or a line-of-sight polar reference frame (Li & Zhu, 2017). Based on this choice, different models of relative dynamics have been studied, and are surveyed in Sullivan, Grimberg, & D'Amico (2017). For near-circular orbits, linear time-invariant Hill–Clohessy–Wiltshire (HCW) equations are the most popular model (Clohessy & Wiltshire, 1960). For elliptical orbits, the linear time-varying Yamanaka–Ankerson (YA) state transition matrix is the usual choice (Yamanaka & Ankersen, 2002). Perhaps a cleaner approach is to avoid relative dynamics by working in the inertial frame,

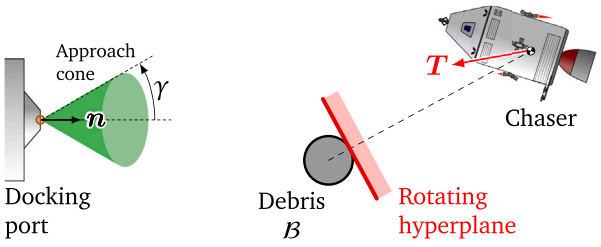


Fig. 13. Illustration of a basic rendezvous scenario. Roughly speaking, the goal is for a chaser spacecraft to use the thrust $\boldsymbol{T}$ from its reaction control system to dock with a target while avoiding debris and respecting constraints such as staying within the approach cone.

as done in (65b). Lu & Liu (2013) and Liu & Lu (2014) showed that fast and reliable trajectory optimization is still possible in this case, by applying the same lossless convexification as in Problem (63) to the constraints (65c) and (65d) and successively linearizing the dynamics (65b). Benedikter, Zavoli, & Colasurdo (2019b) further proposed a filtering technique for updating the linearization reference point to improve the algorithm robustness. The advantage of this approach is its compatibility with general Keplerian orbits and perturbations like J_2 harmonic and aerodynamic drag.

One key challenge in RPO is to avoid collision with external debris or part of the target vehicle itself, which is described by constraint (65e). One approach to enforcing (65e) is to pre-compute a so-called virtual net of trajectories that allows to avoid obstacles in real-time via a simple graph search (Frey, Petersen, Leve, Kolmanovsky, & Girard, 2017; Weiss, Petersen, Baldwin, Erwin and Kolmanovsky, 2015). The pre-computation procedure, however, may be prohibitively computationally demanding. In comparison, solving Problem (65) directly can avoid virtual net construction altogether if an efficient solution method is available. To this end, the keep-out zone $\mathcal{B}(t)$ is usually chosen to be a polytope, an ellipsoid, or the union of a mix of both if multiple keep-out zones are considered (Hu et al., 2018). As shall be seen below, polytope approximation methods yield better optimality, while ellipsoidal methods yield better computational efficiency. The distinction goes back to Sections 2.4 and 2.5, because polytope methods often rely on MIP programming while ellipsoidal methods tend to use SCP.

For the case where $\mathcal{B}(t)$ is a polytope, Schouwenaars, Richards, Feron, & How (2001) and Richards, Schouwenaars, How, & Feron (2002) first proposed to write (65e) as a set of mixed-integer constraints defined by the polytope facets. The resulting trajectory optimization can be solved using MIP methods discussed in Section 2.5. Richards & How (2003a, 2003b, 2006) apply this approach in the context of MPC with a variable horizon trajectory.

For the case where $\mathcal{B}(t)$ is an ellipsoid, (65e) is typically enforced by checking for collision using a conservative time-varying halfspace inclusion constraint:

$$\boldsymbol{r}(t) \in \mathcal{H}(t) \;\Rightarrow\; \boldsymbol{r}(t) \notin \mathcal{B}(t), \tag{66}$$

where $\mathcal{H}(t)$ is a halfspace. Three methods belonging to this family have been used. The first is a rotating hyperplane method, proposed by Di Cairano, Park, & Kolmanovsky (2012) and Park, Di Cairano, & Kolmanovsky (2011). Here, (65e) is replaced by a pre-determined sequence of halfspaces that are tangent to the ellipsoid and rotate around it at a fixed rate. This approach was first applied to a 2D mission, and later extended to 3D (Weiss, Baldwin, Erwin and Kolmanovsky, 2015; Weiss, Kolmanovsky, Baldwin, & Erwin, 2012). A variation was introduced in Park et al. (2016) and further studied in Zagaris et al. (2018), where the rotating sequence is replaced by just two halfspaces tangent to the obstacle and passing through the chaser and target positions. This method requires pre-specifying which of the two halfspaces the chaser belongs to at each time instant.

Fixing the halfspace sequence enables the first two approaches to retain convexity. However, a third and most natural approach is to impose (66) directly by linearizing the ellipsoidal obstacle. This approach is taken in Liu & Lu (2014), and has also been applied to multiple moving obstacles (Jewison, Erwin, & Saenz-Otero, 2015; Wang, Wang, & Zhang, 2018). Because convexity is not maintained, SCP solution methods are used as discussed in Section 2.4. Zagaris et al. (2018) provide a detailed comparison of the three methods.

Another challenge in RPO is the thrust constraint (65d). This constraint allows the thrust magnitude to take any value in the continuous interval $[0, \rho]$. In reality, however, control is often realized by a reaction control system (RCS) that produces short-duration pulses of constant thrust. Therefore, in many applications it makes more sense to consider an *impulse* constraint of the form:

$$\Delta v(t) \in \{0\} \cup [\Delta v_{\min}, \Delta v_{\max}], \tag{67}$$

where $\Delta v(t) \in \mathbb{R}$ approximates the instantaneous change in the chaser's velocity following a firing from the RCS jets. Realistic RCS thrusters have a minimum impulse-bit (MIB) performance metric that governs the magnitude of the smallest possible velocity change $\Delta v_{\min} > 0$. Because (67) is a nonconvex disjoint constraint of the form (2), it has been historically challenging to handle. Indeed, Larsson, Berge, Bodin, & Jönsson (2006) suggest that MIP is necessary in general, but in certain cases the LP relaxation $\Delta v(t) \in [0, \Delta v_{\max}]$ of (67) suffices. This happens, for example, when the velocity measurement noise exceeds the MIB value.

More recently, it was shown that the impulsive rendezvous problem can be solved via polynomial optimization (Arzelier, Kara-Zaitri, Louembet, & Delibasi, 2011; Arzelier, Louembet, Rondepierre, & Kara-Zaitri, 2013). Using results on non-negative polynomials, Deaconu, Louembet, & Théron (2015) showed that impulsive rendezvous with linear path constraints can be solved as an SDP. This formulation was further embedded in a glideslope guidance framework for RPO (Ariba, Arzelier, & Urbina-Iglesias, 2018) and in an MPC approach (Gilz, Joldes, Louembet, & Camps, 2019). Distinct from polynomial optimization, Malyuta & Açıkmeşe (2020b) proved that in some special cases the constraint (67) can be losslessly convexified using techniques similar to those in Section 2.3. For problems where lossless convexification is not possible, (Malyuta & Açıkmeşe, 2021) showed that SCP with a numerical continuation scheme is an effective solution method. Yet another approach was presented in Wan, Dai, & Lu (2019), where an alternating minimization algorithm was proposed for the case $\Delta v_{\min} = \Delta v_{\max}$, in other words when the control is bang–bang.

The impulsive rendezvous model (67) considers an instantaneous firing duration. The model's accuracy can be improved by explicitly considering the finite firing duration, leading to a representation of the actual pulse-width modulated (PWM) thrust signal. PWM rendezvous was first studied in Vazquez, Gavilan, & Camacho (2011, 2014), where an optimization similar to Problem (65) was first solved, then the optimal continuous-valued thrust signal was discretized using a PWM filter and iteratively improved using linearized dynamics. This approach was later embedded in MPC (Vazquez, Gavilan, & Camacho, 2015, 2017). A subtly different approach is presented in Li, Zhu, & Meguid (2016) and Li & Zhu (2018b), called pulse-width pulse-frequency modulation (PWPF). Instead of iteratively refining the thrust signal, PWPF passes the continuous-valued thrust signal to a Schmitt trigger that converts it into a PWM signal. It was shown that this can save fuel and that stability is maintained. However, a potential implementation disadvantage is that the duration of each period in the resulting PWM signal varies continuously, which conflicts with typical hardware where this period is an integer multiple of a fixed duration. An SCP approach was recently used to account for this via state-triggered constraints from Section 2.5.1 (Malyuta et al., 2020).

Although RPO literature tends to focus on the relative chaser-target position using a 3-DoF model, relative attitude control also plays an important role, especially if the target is tumbling (Dong, Luo, Dang,

& Wei, 2020; Li, Yuan, Zhang, & Gao, 2017). Thanks to advances in the speed and reliability of optimization solvers as mentioned in Section 2.1, there has been an increasing interest to optimize 6-DoF RPO trajectories with explicit consideration of position–attitude coupling through constraints such as plume impingement and sensor pointing (Ventura, Ciarcià, Romano, & Walter, 2017; Zhou, Zhang, & Li, 2019). The resulting 6-DoF RPO trajectory optimization, however, is much more challenging to solve due to the presence of attitude kinematics and dynamics. Nevertheless, a special case with field of view and glideslope constraints was presented in Lee & Mesbahi (2014), where 6-DoF RPO was solved as a convex quadratically constrained QP by using a dual quaternion representation of the dynamics, effectively establishing a convexification.

For more general RPO problems, nonlinear programming software has been used frequently. For example, Ventura et al. (2017) used SNOPT (Gill, Murray, & Saunders, 2005) after parameterizing the desired trajectory using polynomials. A B-spline parametrization was used in Sanchez, Gavilan, Vazquez, & Louembet (2020), and the resulting nonlinear optimization was solved by the IPOPT software (Wächter & Biegler, 2005). MATLAB-based packages were also used in Malladi, Di Cairano, & Weiss (2019), Volpe & Circi (2019). Recently, SCP techniques discussed in Section 2.4 were applied to 6-DoF RPO trajectory optimization. Zhou et al. (2019) considered both collision avoidance and sensor pointing constraints. Malyuta et al. (2020) further considered integer constraints on the PWM pulse width in order to respect the RCS MIB value, and constraints on plume impingement, by using state-triggered constraints. The algorithm was improved in (Malyuta & Açıkmeşe, 2021) by making the solution method faster and more robust. The approach uses homotopy ideas from Section 2.5.2 to blend the PTR sequential convex programming method with numerical continuation into a single iterative solution process.

The operation of two spacecraft in close proximity naturally makes RPO a safety-critical phase of any mission. Thus, trajectory optimization that is robust to modeling errors, disturbances, and measurement noise has been an active research topic. MPC has been a popular approach in this context, as it allows efficiently re-solving Problem (65) with online updated parameters using hardware with limited resources (Goodyear et al., 2015; Hartley & Maciejowski, 2014; Park et al., 2013). Hartley (2015) provides a tutorial and a detailed discussion. Among the many different approaches that have been developed to explicitly address robustness, we may count feedback corrections (Baldwin, Erwin, & Kolmanovsky, 2013), the extended command governor (Petersen, Jaunzemis, Baldwin, Holzinger, & Kolmanovsky, 2014), worst-case analysis (Louembet, Arzelier, & Deaconu, 2015; Xu, Chen, Huang, Bai, & Chen, 2018), stochastic trajectory optimization (Jewison & Miller, 2018), chance constrained MPC (Gavilan, Vazquez, & Camacho, 2012; Zhu, Sun, Wang, Wang, & Shao, 2018), sampling-based MPC (Mammarella et al., 2020), tube-based MPC (Dong et al., 2020; Mammarella, Capello, Park, Guglieri, & Romano, 2018), and reactive collision avoidance (Scharf, Açıkmeşe, Ploen, & Hadaegh, 2006). In addition to various uncertainties, anomalous system behavior such as guidance system shutdowns, thruster failures, and loss of sensing, also poses unique challenges in RPO. In order to ensure safety in the presence of these anomalies, Luo, Tang and Lei (2007), Luo, Lei and Tang (2007), and Luo, Tang, & Parks (2008) used a safety performance index to discourage collision with the target, and Breger & How (2008) considered both passive and active collision avoidance constraints in online trajectory optimization. Zhang, Zhao, Zhang and Li (2015) considered passive safety constraints together with field of view and impulse constraints. Aside from optimization-based methods, artificial potential functions (Dong, Hu, & Akella, 2017; Li, Zhang, Yuan, & Wang, 2018; Liu & Li, 2019) and sampling-based methods (Starek, Schmerling, Maher, Barbee, & Pavone, 2017) have also been applied to achieve safety in RPO.

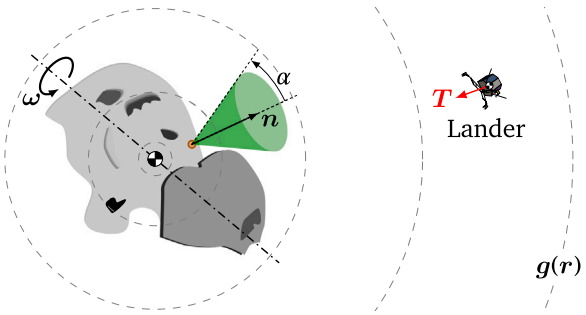


Fig. 14. Illustration of a basic small body landing scenario. The basic concept is to use the thrust $\boldsymbol{T}$ to bring the lander spacecraft to a soft touchdown in the presence of rotational and gravitational nonlinearities, and operational constraints on glideslope, plume impingement, and collision avoidance.

3.3. Small body landing

A maneuver similar to RPO is that of small body landing, where the target spacecraft is replaced by a small celestial object such as an asteroid or a comet. Trajectory optimization for small body landing has gathered increasing levels of attention, spurred by recent high-profile asteroid exploration missions including Hayabusa (Kawaguchi, Fujiwara, & Uesugi, 2008), Hayabusa2 (Crane, 2019), and OSIRIS-REx (Berry et al., 2013; Lauretta et al., 2017). Unlike planetary rocket landing from Section 3.1, small body landing dynamics are highly nonlinear due to the irregular shape, density, and rotation of the small body (Scheeres, Ostro, Hudson, DeJong, & Suzuki, 1998; Werner & Scheeres, 1997). Landing must furthermore ensure a small touchdown velocity, possible plume impingement requirements, and collision avoidance. These aspects pose unique challenges for trajectory optimization. This section reviews recent developments in convex optimization-based small body landing algorithms. Alternative trajectory optimization methods have also been studied for this problem and which we do not cover, such as indirect methods (Chen, Cai, Wang, Zhang, & Liang, 2019; Yang & Baoyin, 2015).

The prototypical small body landing OCP is illustrated in Fig. 14 and can be summarized as follows:

$$\min_{t_f,\boldsymbol{T}} \int_0^{t_f} \|\boldsymbol{T}(t)\|_2 \,\mathrm{d}t \text{ s.t.} \tag{68a}$$

$$\ddot{\boldsymbol{r}}(t) = -2\boldsymbol{\omega}\times\dot{\boldsymbol{r}}(t) - \boldsymbol{\omega}\times(\boldsymbol{\omega}\times\boldsymbol{r}(t)) + \boldsymbol{T}(t)m(t)^{-1} + \boldsymbol{g}(\boldsymbol{r}(t)), \tag{68b}$$

$$\dot{m}(t) = -\alpha\,\|\boldsymbol{T}(t)\|_2\,, \tag{68c}$$

$$\rho_{\min} \le \|\boldsymbol{T}(t)\|_2 \le \rho_{\max}, \tag{68d}$$

$$\|\boldsymbol{r}(t) - \boldsymbol{r}_f\|_2 \cos\alpha \le (\boldsymbol{r}(t) - \boldsymbol{r}_f)^{\mathsf{T}}\boldsymbol{n}, \tag{68e}$$

$$m(0) = m_0,\ \boldsymbol{r}(0) = \boldsymbol{r}_0,\ \dot{\boldsymbol{r}}(0) = \dot{\boldsymbol{r}}_0,\ \boldsymbol{r}(t_f) = \boldsymbol{r}_f,\ \dot{\boldsymbol{r}}(t_f) = 0. \tag{68f}$$

Note the similarity between Problems (63), (65), and (68). Compared to Problem (63), small body landing is expressed in the rotating frame of the target. Thus, the main difference is in the dynamics (68b) that contain a general nonlinear gravity term $\boldsymbol{g}(\boldsymbol{r}(t))$ and inertial forces from the non-negligible angular velocity $\boldsymbol{\omega}$ of the small body. The glideslope constraint (68e) is also shared with the approach cone in RPO (65f).

Early work by Carson III & Açıkmeşe (2006) and Carson III, Açıkmeşe, Murray, & MacMynowski (2008) ignored the mass dynamics (68c), while (68b) was linearized to solve for acceleration rather than a thrust profile. The resulting tube MPC algorithm includes a predetermined feedback controller optimized using SDP and tracking a feedforward trajectory from an SOCP in a robust and recursively feasible manner. Some time later, Pinson & Lu (2015) solved for a fixed-duration trajectory by applying lossless convexification to (68d) and successive linearization to (68b), resulting in an SCP solution method

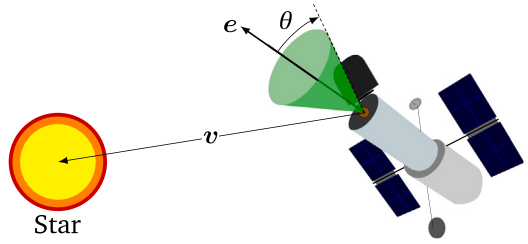


Fig. 15. Illustration of a basic constrained reorientation scenario. The core challenge is to execute an attitude change maneuver while ensuring that the star vector $\boldsymbol{v}$ remains outside of the sensor keep-out cone.

consisting of a sequence of SOCP subproblems. Pinson & Lu (2018) further combine this solution procedure with Brent's line search method to solve for the minimum-fuel flight duration, which is similar to the use of golden-section search in the PDG context (Blackmore et al., 2010). Cui, Liu, Yu, Zhu, & Shao (2017) combined convexification with classic Runge–Kutta discretization to improve the solution accuracy. Yang, Bai, & Baoyin (2017) showed how to solve the minimum-time landing problem as a sequence of convex optimization problems. As a byproduct, they showed that for time-optimal and short-duration minimum-landing-error versions of Problem (68), the thrust stays at its maximum value, in which case the lower bound in (68d) can be removed and (68c) simplified.

Constraint (68e) is the most basic type of collision avoidance constraint. The heuristic reasoning behind (68e) is that if the lander stays above a minimum glideslope, then it will avoid nearby geologic hazards. An alternative two-phase trajectory optimization was introduced in Dunham, Petersen, & Kolmanovsky (2016) and Liao-McPherson, Dunham, & Kolmanovsky (2016) by splitting the landing maneuver into a circumnavigation and a landing phase. During circumnavigation, the spacecraft is far away from the landing site and (68e) is replaced by collision avoidance constraint with the small body. In the same manner as Section 3.2, the small body is wrapped in an ellipsoid and a rotating hyperplane constraint is used (Dunham et al., 2016; Liao-McPherson et al., 2016; Sanchez, Gavilan, & Vazquez, 2018). Reynolds & Mesbahi (2017) introduced an optimal separating hyperplane constraint that also generates auxiliary setpoints for MPC tracking that converge to the landing site. Once in close proximity to the landing site, the spacecraft enters the landing phase where constraint (68e) is enforced to facilitate pinpoint landing.

Most small body landing work is 3-DoF in the sense that it considers point mass translational dynamics. However, recently Zhang, Huang, & Cui (2020) studied a two-phase variable landing duration 6-DoF problem. The motivation was to impose a field of view constraint for a landing camera. The resulting nonconvex optimization trajectory problem was solved using SCP as covered in Section 2.4.

Parameters of the small body, such as $\boldsymbol{\omega}$ and $\boldsymbol{g}$, are often subject to inevitable uncertainty, requiring judicious trajectory design. As a result, many aforementioned works use MPC to cope with the uncertainty in small body landing (Reynolds & Mesbahi, 2017; Sanchez et al., 2018). Application examples include tube MPC (Carson III & Açıkmeşe, 2006; Carson III et al., 2008) and input observers to compensate for gravity modeling errors (Dunham et al., 2016; Liao-McPherson et al., 2016). Hu, Zhu, & Cui (2016) also proposed to jointly minimize fuel and trajectory dispersion described by closed-loop linear covariance. For a detailed discussion on achieving robustness in small body landing, we refer interested readers to the recent survey (Simplício, Marcos, Joffre, Zamaro, & Silva, 2018).

3.4. Constrained reorientation

Scientific observation satellites commonly need to execute large angle reorientation maneuvers while ensuring that their sensitive instruments, such as cryogenically cooled infrared telescopes, are not exposed to direct sunlight or heat. Famous examples include the Cassini spacecraft, the Hubble Space Telescope, and the upcoming James Webb Space Telescope (Downes & Rose, 2001; Long, 2004; Singh et al., 1997). This section discusses the challenges of constrained reorientation as a trajectory optimization problem, and focuses on how convex optimization methods have been leveraged to address these challenges.

A basic constrained reorientation OCP is illustrated in Fig. 15 and can be formulated as follows:

$$\min_{t_f,\boldsymbol{u}} \int_0^{t_f} \|\boldsymbol{u}(t)\|_2 \,\mathrm{d}t \text{ s.t.} \tag{69a}$$

$$\dot{\boldsymbol{q}}(t) = \frac{1}{2}\boldsymbol{q}(t) \otimes \boldsymbol{\omega}(t), \tag{69b}$$

$$\boldsymbol{J}\dot{\boldsymbol{\omega}}(t) = \boldsymbol{u}(t) - \boldsymbol{\omega}(t) \times (\boldsymbol{J}\boldsymbol{\omega}(t)), \tag{69c}$$

$$\boldsymbol{q}(t)^\mathsf{T} \boldsymbol{M}_i \boldsymbol{q}(t) \leq 0,\ i = 1, \dots, n, \tag{69d}$$

$$\|\boldsymbol{\omega}(t)\|_\infty \leq \omega_{\max},\ \ \|\boldsymbol{u}(t)\|_\infty \leq u_{\max}, \tag{69e}$$

$$\boldsymbol{q}(0) = \boldsymbol{q}_0,\ \boldsymbol{\omega}(0) = \boldsymbol{\omega}_0,\ \boldsymbol{q}(t_f) = \boldsymbol{q}_f,\ \boldsymbol{\omega}(t_f) = \boldsymbol{\omega}_f. \tag{69f}$$

The set of constraints (69d) encodes conical keep-out zones for n stars, similarly to the illustration in Fig. 15 for one star. The parameters $\boldsymbol{M}_i \in \mathbb{R}^{4\times4}$ are symmetric matrices that are not positive semidefinite, as introduced in Section 2.3. The main challenge of solving Problem (69) stems from the fact that (69d) and the attitude dynamics (69b)–(69c) are nonconvex. Kim & Mesbahi (2004) were the first to prove that (69d) can be losslessly replaced by convex quadratic constraints, provided $\|\boldsymbol{q}(t)\|_2 = 1$. Based on this observation, Kim & Mesbahi (2004) proposed to greedily optimize one discretization point at a time instead of the entire trajectory jointly. The method was further extended to the case of integral and dynamic pointing constraints in Kim, Mesbahi, Singh, & Hadaegh (2010).

Although the method of Kim & Mesbahi (2004) is computationally efficient, it is inherently conservative and may fail to find a feasible solution to Problem (69) by greedily optimizing one discretization point at a time. As a result, several attempts have been made to improve its performance. For example, Tam & Lightsey (2016) propose to replace constraint (69d) with penalty terms in the objective function in order to ensure that a feasible trajectory can be found. Binary logical variables were also introduced in (69d) to account for redundant sensors. Hutao, Xiaojun, Rui, & Pingyuan (2011) showed how the convexification of constraints (69d) should be adjusted when optimizing an entire trajectory, rather than a single time step as originally done in Kim & Mesbahi (2004). Put into an MPC framework, the resulting trajectory optimization yields less conservative performance. Alternatively, Eren, Açıkmeşe and Scharf (2015) proposed to first optimize a quaternion sequence without kinematic and dynamic constraints, and then to compute the corresponding angular velocity and torque using the resulting quaternions. A hyperplane approximation of the unit sphere is used during quaternion optimization to ensure dynamic feasibility, and is imposed via MIP. Recently, McDonald, Grizzel, & Wang (2020) proposed an SCP method with a line search step that helps convergence, which provides a potential real-time solution to Problem (69).

Aside from the quaternion representation in Problem (69), which is the most popular choice, a direction cosine matrix representation of attitude was also used by Walsh & Forbes (2018) to solve an equivalent problem. The resulting trajectory optimization can be approximated as an SDP using successive linearization and relaxing (69d).

Due to its challenging nature, Problem (69) has inspired many optimization solutions other than those based on convex optimization. Pseudospectral methods and NLP optimization software have all been used to solve Problem (69) directly (Lee & Mesbahi, 2013; Xiaojun, Hutao, Pingyuan, & Rui, 2010). An indirect shooting method was used in Lee et al. (2017), Phogat, Chatterjee, & Banavar (2018b), and a differential evolution method was used in Wu, Xu, Zhu, & Cui (2017) and Wu & Han (2019). Compared with convex optimization based methods, these methods typically require more computational resources to achieve real-time implementation.

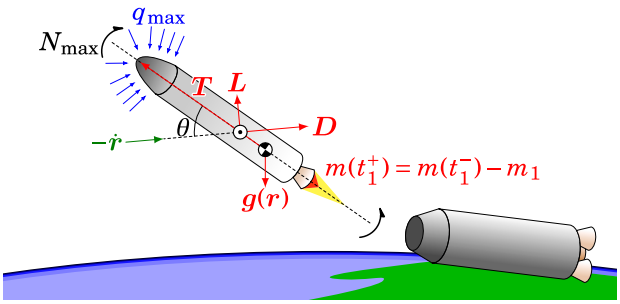


Fig. 16. Illustration of a basic ascent scenario. The goal is to find an optimal angle-of-attack θ trajectory to transfer the launch vehicle's payload from the planet's surface to orbit, while minimizing fuel and satisfying structural integrity constraints.

3.5. Endo-atmospheric flight

Launching from or returning to a planet with an atmosphere are integral parts of many space missions. These problems concern launch vehicles, missiles, and entry vehicles such as capsules, reusable launchers, and hypersonic gliders. Significant portions of launch and entry occur at high velocities and in the presence of an atmosphere, making aerodynamics play a large role. Aerodynamics and thermal heating are indeed the core differentiating factors between endo-atmospheric flight and PDG from Section 3.1. For the latter problem, small velocities and thinness of the atmosphere make aerodynamic effects negligible in many cases (Eren, Dueri et al., 2015). This section summarizes recent contributions to endo-atmospheric trajectory planning using convex optimization-based methods. In particular, Section 3.5.1 discusses ascent and Section 3.5.2 discusses entry.

3.5.1. Ascent flight

The optimal ascent problem seeks to transfer a launch vehicle's payload from a planet's surface to orbit while minimizing a quantity such as fuel. Naturally, optimal control theory from Section 2.1 has found frequent applications in ascent guidance, and we refer the reader to Hanson, Shrader, & Cruzen (1994) for a survey. Heritage algorithms date back to the iterative guidance mode (IGM) of Saturn rockets (Adkins, 1970; Chandler & Smith, 1967; Haeussermann, 1970; Horn, Chandler, & Buckelew, 1969) and the powered explicit guidance (PEG) of the Space Shuttle (McHenry, Long, Cockrell, Thibodeau III, & Brand, 1979). A simple yet relevant optimal control problem describing an orbital launch scenario is known as the *Goddard rocket problem* (Betts, 2010; Bryson Jr. & Ho, 1975). A version with variable gravity and no atmospheric drag is stated as follows:

$$\min_{t_f, \boldsymbol{T}} -m(t_f) \text{ s.t.} \tag{70a}$$

$$\ddot{\boldsymbol{r}}(t) = -\mu \left\|\boldsymbol{r}(t)\right\|_2^{-3} \boldsymbol{r}(t) + m(t)^{-1}\boldsymbol{T}(t), \tag{70b}$$

$$\dot{m}(t) = -\alpha \left\|\boldsymbol{T}(t)\right\|_2, \tag{70c}$$

$$m(0) = m_0,\ \boldsymbol{r}(0) = \boldsymbol{r}_0,\ \dot{\boldsymbol{r}}(0) = \dot{\boldsymbol{r}}_0,\ \boldsymbol{\psi}(\boldsymbol{r}(t_f), \dot{\boldsymbol{r}}(t_f)) = 0. \tag{70d}$$

Problem (70) models a three-dimensional point mass moving in a gravity field under the influence of thrust. As such, it also applies to orbit transfer problems which we discuss later in Section 3.6. The vector $\boldsymbol{r}(t) \in \mathbb{R}^3$ is the position vector, $\boldsymbol{T}(t) \in \mathbb{R}^3$ is the thrust vector, and $m(t) \in \mathbb{R}$ is the vehicle mass. The vector function $\boldsymbol{\psi} : \mathbb{R}^6 \to \mathbb{R}^k$ imposes $k \leq 6$ terminal conditions. In ascent and orbit transfer applications, $\boldsymbol{\psi}$ usually acts to constrain k orbital elements while leaving the other $6-k$ orbital elements free.

A key issue when solving Problem (70) using an indirect method is to resolve the transversality conditions of the resulting TPBVP (Berkovitz, 1974; Pontryagin et al., 1986):

$$\boldsymbol{p}(t_f) = \left[\nabla_{\boldsymbol{x}} \boldsymbol{\psi}\left(\boldsymbol{x}(t_f)\right)\right]^\mathsf{T} \boldsymbol{\nu}_p, \tag{71}$$

where $\boldsymbol{x}(t) \triangleq (\boldsymbol{r}(t); \dot{\boldsymbol{r}}(t))$, $\boldsymbol{p}(\cdot) \in \mathbb{R}^6$ are the costates relating to the position and velocity, and $\boldsymbol{\nu}_p \in \mathbb{R}^k$ is a Lagrange multiplier vector. Unfortunately, $\boldsymbol{\nu}_p$ has no physical or exploitable numerical interpretation, and the magnitudes of its elements can vary wildly (Pan, Chen, Lu, & Gao, 2013). This causes a lot of difficulty for the solution process in terms of numerics, robustness, and initial guess selection. Traditionally, the problem is overcome by converting (71) into a set of $6-k$ so-called reduced transversality conditions, which are equivalent (Lu, Sun, & Tsai, 2003):

$$\left[\nabla_{\boldsymbol{x}} \boldsymbol{\psi}\left(\boldsymbol{x}(t_f)\right)\right] \boldsymbol{y}_i = 0, \quad i = 1, \ldots, 6-k, \tag{72a}$$

$$\boldsymbol{y}_i^\mathsf{T} \boldsymbol{p}(t_f) = 0, \quad i = 1, \ldots, 6-k. \tag{72b}$$

The linearly independent vectors $\boldsymbol{y}_i \in \mathbb{R}^6$ are known as the reduced transversality vectors, and are a function of $\boldsymbol{x}(t_f)$. If they are known analytically, then (72b) can replace (71), which eliminates $\boldsymbol{\nu}_p$ from the problem and simplifies the solution process considerably. However, solving for $\boldsymbol{y}_i$ symbolically is a difficult task, and the resulting expressions can be complicated (Brown & Johnson, 1967). An alternative approach was introduced in Pan et al. (2013), where the authors provide an easy to use "menu" of the $6-k$ constraints in (72b) that are needed. This is achieved by considering Problem (70) specifically and exploiting the structure offered by the classical orbital elements. It is only assumed that the terminal constraint function $\boldsymbol{\psi}$ fixes exactly k of the 6 orbital elements, and leaves the other orbital elements free.

The Goddard rocket problem in Problem (70) assumes no atmosphere. When there is an atmosphere, a popular classical method is the *gravity turn* maneuver, which maintains a low angle-of-attack so as to minimize lateral aerodynamic loads. However, the general ascent problem with an atmosphere is complicated due to strong coupling of aerodynamic and thrust forces (Pan & Lu, 2010). Thus, ascent is typically performed via open-loop implicit guidance, in the sense that feedback control is used to track a pre-computed ascent trajectory stored onboard as a lookup table. However, this approach cannot robustly handle off-nominal conditions, aborts, and contingencies, which motivates research into closed-loop ascent techniques (Brown & Johnson, 1967; Lu, 2017).

A notable strategy in this context is to include aerodynamics in an onboard ascent solution via a homotopy method starting from an optimal vacuum ascent initial guess (Calise & Brandt, 2004; Calise, Melamed, & Lee, 1998; Gath & Calise, 2001). Another approach was developed in Lu & Pan (2010), Lu et al. (2003), Lu, Zhang, & Sun (2005) and Pan & Lu (2010) using indirect trajectory optimization. Here, a finite-difference scheme is proposed to solve for the necessary conditions of optimality for ascent with an atmosphere. In particular, fixed-point formulations were considered (Lu et al., 2005), primer vector theory was invoked to determine trajectory optimality (Lu & Pan, 2010), and a generalization to arbitrary numbers of burn and coast arcs was developed (Pan & Lu, 2010). Finally, indirect methods relying on control smoothing via trigonometrization have been developed to address problems with bang–bang input and singular arcs (Mall, Grant, & Taheri, 2020). The Epsilon-Trig method (Mall & Grant, 2017), which is an example of such an approach, was applied to the Goddard maximum altitude ascent problem to obtain its bang–singular–bang optimal solution. See Section 2.1.1 for a brief description of these approaches.

Modern improvements in convex optimization have made direct optimization methods attractive for ascent guidance. To this end, consider the following illustrative ascent problem for a two-stage launch vehicle, as shown in Fig. 16:

$$\min_{t_f, \theta} -m(t_f) \text{ s.t.} \tag{73a}$$

$$\ddot{\boldsymbol{r}}(t) = -\mu \left\|\boldsymbol{r}(t)\right\|_2^{-3} \boldsymbol{r}(t) + m(t)^{-1}(\boldsymbol{T}[t] + \boldsymbol{L}[t] + \boldsymbol{D}[t]), \tag{73b}$$

$$\dot{m}(t) = -\alpha \left\|\boldsymbol{T}[t]\right\|_2, \tag{73c}$$

$$\theta_1 \leq \theta(t) \leq \theta_2, \tag{73d}$$

$$\rho[t] \left\|\dot{\boldsymbol{r}}(t)\right\|_2^2 \leq q_{\max}, \tag{73e}$$

$$\rho[t]\,\|\dot{\boldsymbol{r}}(t)\|_2^2\,|\theta(t)| \le N_{\max}, \tag{73f}$$

$$m(0) = m_0,\ \boldsymbol{r}(0) = \boldsymbol{r}_0,\ \dot{\boldsymbol{r}}(0) = \dot{\boldsymbol{r}}_0,\ \boldsymbol{\psi}(\boldsymbol{r}(t_f), \dot{\boldsymbol{r}}(t_f)) = 0, \tag{73g}$$

$$m(t_1^+) = m(t_1^-) - m_1. \tag{73h}$$

Problem (73) is planar and formulated in an Earth-centered inertial (ECI) frame. Control is performed using the angle-of-attack θ, which determines the direction of an otherwise pre-determined thrust profile (Li et al., 2020; Liu & Lu, 2014; Zhang, Yang, & Xiong, 2019). The major aerodynamic forces are those of lift $\boldsymbol{L}$ and drag $\boldsymbol{D}$, each of which may be complex expressions of state and control. Note that in (73b) we used the shorthand $\boldsymbol{T}[t]$, $\boldsymbol{L}[t]$, and $\boldsymbol{D}[t]$ from Section 1 to abstract away the possible state and control arguments. The atmospheric density is denoted by ρ, which varies during ascent as a nonlinear function of the position $\boldsymbol{r}$. An example is given later in (75c). Important constraints on the dynamic pressure (73e) and bending moment (73f) are used to ensure the vehicle's structural integrity (Lu & Pan, 2010). The target orbit is prescribed by the vector function $\boldsymbol{\psi}$ in (73g), which is the same as in (70d) and specifies some or all of the target orbital elements. Benedikter, Zavoli, Colasurdo, Pizzurro, & Cavallini (2020) chose boundary conditions based on the radius and inclination of a circular target orbit. A final nuance is that if the rocket is assumed to be a two-stage vehicle, a stage separation event must be scheduled at a pre-determined time t_1 via (73h). At the separation instant, the mass variable experiences a discontinuous decrease that amounts to the dry weight of the first stage (Benedikter, Zavoli, & Colasurdo, 2019a; Benedikter et al., 2020). A related constraint for stage separation requires $\theta(t_1) = 0$ in order to reduce lateral load (Zhengxiang, Tao, Songyan, & Ming, 2018). Furthermore, the splashdown location of a burnt-out separated stage can also be constrained (Benedikter et al., 2020).

Due to the presence of strong nonlinearities, convex optimization-based solution algorithms for Problem (73) typically use SCP from Section 2.4. However, several manipulations have been helpful to make the problem less nonlinear. Conversion of the system dynamics (73b) to control-affine form, at times by choosing an independent variable other than time, followed by the use of lossless convexification within an SCP framework has been a common approach. Zhang et al. (2019) obtained a control-affine form by assuming the AoA to be small and defining $u_1 = \theta$, $u_2 = \theta^2$ as the new control variables. This choice makes drag a linear function of the control, while the constraint $u_1^2 = u_2$ is relaxed to $u_1^2 \le u_2$ via lossless convexification. Similarly, Benedikter et al. (2019a, 2020) chose thrust direction as input and losslessly convexified the unit norm constraint on the thrust direction to a convex inequality. Cheng, Li, & Zhang (2017) considered a 3D problem with AoA and bank angle as control inputs, and applied lossless convexification to a constraint of the form $u_1^2 + u_2^2 + u_3^2 = 1$. Furthermore, their choice of altitude as the independent variable simplified the convexification of constraints involving density, since it renders the density a state-independent quantity. In particular, during collocation over a known grid within an altitude interval, the density value is known at each node. This fails to be the case when collocation is performed over time. The choice of altitude as independent variable was also explored in Liu, Shen, & Lu (2016b).

The two-agent launch problem is an interesting and relevant modern-day extension of Problem (73) (Ross & D'Souza, 2004). In this case, the launch vehicle first stage is not just an idle dropped mass, but is a controlled vehicle that must be brought back to Earth. This is the case for the SpaceX Falcon 9 rocket, whose first stage is recovered by propulsive landing after a series of post-separation maneuvers (Blackmore, 2016). It was shown in Ross & D'Souza (2005) how hybrid optimal control can be used to solve the problem via mixed-integer programming. More generally, hybrid optimal control has also found applications in low-thrust orbit transfer using solar sails (Stevens & Ross, 2005; Stevens, Ross, & Matousek, 2004).

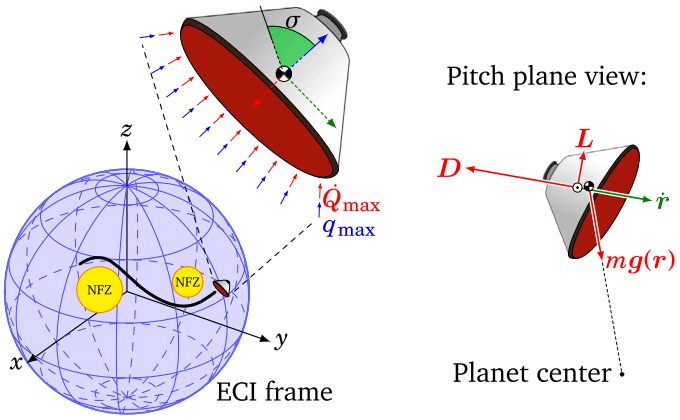


Fig. 17. Illustration of a basic atmospheric entry scenario. The goal is to find an optimal bank angle σ trajectory to dissipate energy while meeting structural integrity and targeting requirements.

3.5.2. Atmospheric entry

Atmospheric entry, also known as reentry, is fundamentally a process of controlled energy dissipation while meeting targeting and structural integrity constraints (Lu, 2014). Computer-controlled entry guidance dates back to the Gemini and Apollo projects, and D'Souza & Sarigul-Klijn (2014) provide a comprehensive survey of existing methods. Good documentation is available for Mars Science Laboratory's entry guidance, which is based on Apollo heritage (Mendeck & Craig, 2011; Steltzner, San Martín, Rivellini, Chen, & Kipp, 2014; Way et al., 2007).

A large body of work is available on predictor–corrector methods for entry guidance (Johnson, Braden, Sostaric, Cerimele, & Lu, 2018; Johnson, Lu, & Stachowiak, 2017; Johnson et al., 2020; Lu, 2014; Xue & Lu, 2010) and for aerocapture (Lu, Cerimele, Tigges, & Matz, 2015). These methods are based on root-finding algorithms, or variations thereof, and some versions are grounded in solving the necessary conditions of optimality (Lu, 2018). We refer the reader to Lu (2008) for further details. In addition to reentry trajectory generation, mission analysis tools for generating landing footprints have also been developed (Eren, Dueri et al., 2015; Lu & Xue, 2010).

Guidance schemes based on univariate root-finding, which are near-optimal for reentry (Lu, 2014) and optimal for aerocapture (Lu et al., 2015), have also been developed. reentry applications use the quasi-equilibrium glide condition (QEGC), while aerocapture leverages the bang–bang nature of the control solution obtained via the maximum principle. By using the knowledge of where the input switches, univariate root-finding can approximate the optimal solution in each phase to high accuracy. Such an approach, though based on an indirect method, avoids directly solving the TPBVP. Recall that lossless convexification, discussed in Section 2.3, is another approach where clever reformulation of the optimal control problem and application of the maximum principle yields an efficient solution strategy. This ties back to the last paragraph of Section 2.1.2, which states that the fusion of indirect and direct solution methods often yields more efficient solution algorithms than using any one method in isolation. Because root-finding algorithms do not involve an explicit call to an optimizer, we do not survey them here. Instead, this section focuses on contributions by convex optimization-based methods to the problem of entry trajectory computation.

Another methodology that simplifies the typical strategy in indirect methods is the RASHS approach (Saranathan & Grant, 2018). As discussed in Section 2.5.2, RASHS converts a multi-phase optimal control problem into a single-phase problem by using sigmoid functions of state-based conditions to instigate smooth transitions between phases. As a consequence, the multi-point BVP from Pontryagin's maximum

principle is reduced to a TPBVP. The complete entry, descent, and landing (EDL) problem is one example that can be solved effectively via this technique.

Consider a basic entry guidance problem illustrated in Fig. 17, which is formulated as follows:

$$\min_{u} \max_{t\in[0,t_f]} \dot{Q}[t] \text{ s.t.} \tag{74a}$$

$$\ddot{\boldsymbol{r}}(t) = -\mu \left\|\boldsymbol{r}(t)\right\|_2^{-3} \boldsymbol{r}(t) + m^{-1}\left(\boldsymbol{L}[t] + \boldsymbol{D}[t]\right), \tag{74b}$$

$$|u(t)| \leq 1, \tag{74c}$$

$$\rho[t] \left\|\dot{\boldsymbol{r}}(t)\right\|_2^2 \leq q_{\max}, \tag{74d}$$

$$\left\|\boldsymbol{L}[t] + \boldsymbol{D}[t]\right\|_2 \leq n_{\max}, \tag{74e}$$

$$\boldsymbol{r}(0) = \boldsymbol{r}_0,\ \dot{\boldsymbol{r}}(0) = \dot{\boldsymbol{r}}_0. \tag{74f}$$

Problem (74) is planar and formulated in an ECI frame like Problem (73). Aerodynamic forces are governed by the lift, drag, and atmospheric density, which are expressed as follows:

$$\boldsymbol{L}[t] = R_{\pi/2} c_L \rho[t] \left\|\dot{\boldsymbol{r}}(t)\right\|_2 \dot{\boldsymbol{r}}(t) u(t), \tag{75a}$$

$$\boldsymbol{D}[t] = R_{\pi} c_D \rho[t] \left\|\dot{\boldsymbol{r}}(t)\right\|_2 \dot{\boldsymbol{r}}(t), \tag{75b}$$

$$\rho[t] = \rho_0 \exp\left(-(\left\|\boldsymbol{r}(t)\right\|_2 - r_0)/h_0\right). \tag{75c}$$

The lift and drag coefficients are given by c_L and c_D while ρ_0, h_0, and r_0 denote the reference density, reference altitude, and planet radius. R_θ corresponds to a counter-clockwise rotation by θ radians. The control input $u(t) = \cos(\sigma(t))$ is the cosine of the bank angle, and serves to modulate the projection of the lift vector onto the plane of descent, known also as the *pitch plane*. Entry is an extremely stressful maneuver for much of the spacecraft's hardware, therefore structural integrity constraints are placed on dynamic pressure (74d) and aerodynamic load (74e). The objective is to minimize the peak heating rate, given by the Detra–Kemp–Riddell stagnation point heating correlation (Detra, Kemp, & Riddell, 1957; Garrett & Pitts, 1970), which is appropriate for an insulative reusable thermal protection system (TPS) such as on the Space Shuttle and SpaceX Starship:

$$\dot{Q}[t] \propto \sqrt{\rho[t]} \left\|\dot{\boldsymbol{r}}(t)\right\|_2^{3.15}. \tag{76}$$

Problem (74) is a simple example that gives a taste for the reentry problem. We now survey variants of this problem that have been explored in the literature. First of all, many other objectives have been proposed in place of (74a). These include minimum heat load (Han, Qiao, Chen, & Li, 2019; Wang & Grant, 2017), minimum peak normal load (Wang, 2019a, 2019b), minimum time-of-flight (Han et al., 2019; Wang, Cui, & Wei, 2019b), minimum terminal velocity (Wang & Grant, 2017), maximum terminal velocity (Wang & Grant, 2019), minimum phugoid oscillation (Liu & Shen, 2015), and minimum cross-range error (Fahroo, Doman, & Ngo, 2003a, 2003b). In the problem of aerocapture, where a spacecraft uses the planet's atmosphere for insertion into a parking orbit, minimum velocity error (Zhang, Açıkmeşe, Swei and Prabhu, 2015) and minimum impulse, time-of-flight, or heat load (Han et al., 2019) were studied. Minimizing the total heat load, which is equivalent to the average heating rate, is particularly relevant for ablative TPS that work by carrying heat away from the spacecraft surface through mass loss. This has been the method of choice for Apollo, SpaceX Crew Dragon, and almost all interplanetary entry systems, because it can sustain very high transient peak heating rates (Hicks, 2009).

Problem (74) is expressed in the pitch plane and without regard for planetary rotation. To account for rotation and aspects like cross-range tracking, other formulations have been explored. This includes a pitch plane formulation with rotation (Chawla, Sarmah, & Padhi, 2010), a 3D formulation without rotation (Zhao & Song, 2017), and a 3D formulation with rotation (Han et al., 2019; Liu & Shen, 2015; Liu, Shen, & Lu, 2015, 2016a; Wang, 2019a; Wang et al., 2019a, 2019b; Wang & Grant, 2017, 2018a).

The two main path constraints present in Problem (74) are on the dynamic pressure (74d) and aerodynamic load (74e). The heating rate is also indirectly constrained since (74a) must achieve a lower value than the maximum heating rate $\dot{Q}_{\max}$, otherwise the computed trajectory melts the spacecraft. Since these three constraints are critical for structural integrity, they permeate much of reentry optimization literature (Han et al., 2019; Liu & Shen, 2015; Liu et al., 2015, 2016a; Sagliano & Mooij, 2018; Wang, 2019a; Wang et al., 2019a, 2019b; Wang & Grant, 2017, 2018a, 2019; Zhao & Song, 2017). Some researchers have also included no-fly zone (NFZ) constraints, as illustrated in Fig. 17 (Liu et al., 2016a; Zhao & Song, 2017). A bank angle reversal constraint has also been considered in Han et al. (2019), Liu et al. (2015, 2016a), Zhao & Song (2017) and Liu & Shen (2015). This is a nonconvex constraint of the form:

$$0 < \sigma_{\min} \leq |\sigma(t)| \leq \sigma_{\max}. \tag{77}$$

A common approach to handle (77) is to define $u_1(t) \triangleq \cos(\sigma(t))$ and $u_2(t) \triangleq \sin(\sigma(t))$, and to impose:

$$\cos(\sigma_{\max}) \leq u_1(t) \leq \cos(\sigma_{\min}),\ u_1(t)^2 + u_2(t)^2 = 1, \tag{78}$$

where the nonconvex equality constraint is subsequently losslessly convexified to $u_1(t)^2 + u_2(t)^2 \leq 1$ (Liu & Shen, 2015; Liu et al., 2015, 2016a).

The bank angle with a prescribed AoA profile is a popular control input choice for reentry, dating back to Apollo (Rea & Putnam, 2007). Some works have considered bank angle rate as the input, which improves control smoothness (Wang, 2019a; Wang et al., 2019a, 2019b; Wang & Grant, 2017, 2018a, 2019). However, banking is not the only possible control mechanism for reentry, and several other choices have been explored. Chawla et al. (2010) use the AoA as input and omit bank and heading. Fahroo et al. (2003a) use AoA, bank angle, and altitude, assuming the aforementioned QEGC with a small flight-path angle between the velocity vector and the local horizontal. Zhao & Song (2017) use bank angle and a normalized lift coefficient as inputs.

High frequency oscillation in the control signal, known as jitter, is a common issue in entry trajectory optimization. Several works explicitly address this issue (Liu et al., 2015, 2016a; Szmuk et al., 2017; Wang & Grant, 2017). Jitter is believed to be caused by the nonlinear coupling of state and control constraints (Wang & Grant, 2017), and it appears to be reduced by a control-affine reformulation of the dynamics (Liu et al., 2016a). Other strategies to remove jitter have been to apply the reparametrization (78) or to filter the control signal. The latter approach includes the aforementioned use of bank angle rate as the control, or using a first-order low-pass filter (Liu et al., 2015).

Aside from fixing jitter, efforts have been devoted to simplifying the SCP-based solution methods, and to improving their convergence properties. Reformulating the dynamics using energy as the independent variable, in a similar way to how altitude was used for optimal ascent, is one tactic that achieves the former (Liu & Shen, 2015; Liu et al., 2015, 2016a; Lu, 2014). Such a parametrization eliminates the differential equation for airspeed, and instead yields an algebraic approximation for airspeed in terms of energy. Fahroo et al. (2003a, 2003b) applied a related elimination process by considering energy as a state variable. Apart from this, it is worth noting the heuristics proposed for improving the convergence of the SCP-based approaches. Liu et al. (2015), Wang (2019b) and Wang & Grant (2019) used backtracking line search at each SCP iteration to reduce constraint violation. It was found that with the line search, the number of iterations required for convergence reduced by half (Liu et al., 2015). Zhang, Açıkmeşe et al. (2015) constrained the SCP iterates to form a Cauchy sequence. Wang et al. (2019b) proposed a dynamic trust region update scheme that is tailored for hypersonic reentry. In particular, the trust region update accounts for the linearization error due to each state instead of the typical approach of considering the average linearization error.

Aside from using SCP to optimize the entire entry trajectory, another popular approach for entry guidance is via MPC from Section 2.6. Some

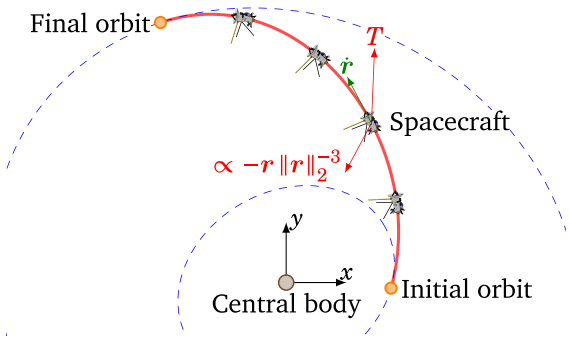


Fig. 18. Illustration of a basic orbit transfer scenario. The goal is to use thrust $\boldsymbol{T}$ to transfer the spacecraft state from an initial orbit to a target orbit under the influence of gravity, while minimizing a quantity such as fuel or time.

important recent MPC-based developments are the dynamic control allocation scheme (Luo, Serrani, Yurkovich, Oppenheimer and Doman, 2007) and the application of model predictive static programming (MPSP). The approach by Luo, Serrani et al. (2007) centers around posing an SQP problem as a linear complimentarity problem, while the principle behind MPSP is to combine MPC and approximate dynamic programming though a parametric optimization formulation (Chawla et al., 2010; Halbe, Mathavaraj, & Padhi, 2010; Halbe, Raja, & Padhi, 2014). van Soest, Chu, & Mulder (2006) and Recasens, Chu, & Mulder (2005) corroborate the effectiveness of MPC-based approaches by comparing the performance of constrained MPC with that of PID control applied to feedback-linearized reentry flight.

Last but not least, we conclude by discussing pseudospectral discretization from Section 2.2.4 as a popular methodology in a variety of reentry problem formulations. The method is appealing for its ability to yield accurate solutions with a relatively sparse temporal collocation grid, and recent results on the estimation of costates with spectral accuracy provide a strong theoretical connection to optimal control theory and the maximum principle (Françolin et al., 2014; Gong, Ross, & Fahroo, 2010). Fahroo et al. (2003a, 2003b), Rea (2003) applied direct Legendre collocation to generate an entry vehicle footprint by solving a nonconvex NLP. Sagliano & Mooij (2018) used Legendre–Lobatto collocation and lossless convexification to generate an optimal profile for the heritage drag-energy guidance scheme. In addition to these approaches, which rely solely on the direct method, a combination of direct and indirect methods was discussed in Josselyn & Ross (2003) for verifying optimality of reentry trajectories using the DIDO solver (see Table 1). In Tian & Zong (2011), a feedback guidance law through an indirect Legendre pseudospectral method was developed to track a reference generated using a direct pseudospectral method. Finally, akin to explicit MPC, Sagliano, Mooij, & Theil (2017) developed a pre-computed interpolation-based multivariate pseudospectral technique that is coupled with a subspace selection algorithm to generate nearly optimal trajectories in real-time for entry scenarios in the presence of wide dispersions at the entry interface.

3.6. Orbit transfer and injection

A usual task in a space mission is to attain a certain orbit, or to change orbits. The goal of the so-called orbit transfer and injection (OTI) problem is to transport a low-thrust space vehicle from an initial to a target orbit while minimizing a quantity such as time or fuel. Unsurprisingly, the problem is as old as spaceflight itself, with the earliest bibliographic entry dating to the late 1950s (Faulders, 1958). Traditionally, the problem has been solved using optimal control theory from Section 2.1, and for this we can cite the books (Bryson Jr. & Ho, 1975; Conway, 2014; Kirk, 1970; Lawden, 1963; Longuski et al., 2014). Numerous solution methods have been studied, including methods based on primer vector theory (Petropoulos & Russell, 2008; Restrepo & Russell, 2017; Russell, 2007), direct methods based on solving an NLP (Arrieta-Camacho & Biegler, 2005; Betts, 2000; Graham & Rao, 2015, 2016; Ross, Gong, & Sekhavat, 2007; Starek & Kolmanovsky, 2012), and indirect methods (Alfano & Thorne, 1994; Cerf, 2016; Fernandes, 1995; Gil-Fernandez & Gomez-Tierno, 2010; Gong, Fahroo, & Ross, 2008; Haberkorn, Martinon, & Gergaud, 2004; Kechichian, 1995; Pan, Lu, & Chen, 2012; Pontani & Conway, 2013; Taheri, Kolmanovsky, & Atkins, 2016; Taheri, Li, & Kolmanovsky, 2017; Zimmer, Ocampo, & Bishop, 2010). Some recent advances for indirect methods include homotopy methods (Cerf, Haberkorn, & Trélat, 2011; Di Lizia et al., 2013; Pan & Pan, 2020; Pan et al., 2019; Rasotto et al., 2015), optimal switching surfaces (Taheri & Junkins, 2019), the RASHS and CSC approaches from Section 2.5.2 (Saranathan & Grant, 2018; Taheri et al., 2020a; Taheri, Junkins, Kolmanovsky, & Girard, 2020b), and simultaneous optimization (also known as co-optimization) of the trajectory and the spacecraft design parameters (Arya et al., 2020).

In a similar way to the previous sections, improvements in convex optimization technology have prompted an increased interest in applying the direct family of methods to OTI. For example, Betts & Erb (2003) solved a minimum-fuel Earth to Moon transfer using a solar electric propulsion system, which is a complex problem with a transfer duration of over 200 days. The problem is highly nonconvex, and the optimization algorithm is based on SQP. This section discusses some of the recent developments for solving OTI using convex optimization-based methods, and their extensions to optimal exo-atmospheric launch vehicle ascent.

A basic optimal OTI problem is illustrated in Fig. 18 and can be formulated as follows:

$$\min_{t_f,\boldsymbol{T}} \int_0^{t_f} \|\boldsymbol{T}(t)\|_2 \,\mathrm{d}t \text{ s.t.} \tag{79a}$$

$$\ddot{\boldsymbol{r}}(t) = -\mu \|\boldsymbol{r}(t)\|_2^{-3} \boldsymbol{r}(t) + m(t)^{-1}\boldsymbol{T}(t), \tag{79b}$$

$$\dot{m}(t) = -\alpha \|\boldsymbol{T}(t)\|_2, \tag{79c}$$

$$\|\boldsymbol{T}(t)\|_2 \le \rho, \tag{79d}$$

$$m(0) = m_0,\ \boldsymbol{r}(0) = \boldsymbol{r}_0,\ \dot{\boldsymbol{r}}(0) = \dot{\boldsymbol{r}}_0, \tag{79e}$$

$$\boldsymbol{\psi}(\boldsymbol{r}(t_f), \dot{\boldsymbol{r}}(t_f)) = 0. \tag{79f}$$

Just like in Problem (70), the vector function $\boldsymbol{\psi}$ in (79f) describes the final orbit insertion constraints, usually in the form of orbital elements. Note that (79b)–(79d) are identical to Problem (65). Naturally, we may hope that previously developed lossless convexification and SCP techniques from Lu & Liu (2013) apply for OTI. The main novelty is the nonlinear insertion constraint (79f). Liu & Lu (2014) showed that (79f) can be linearized with a second-order correction term, and Problem (79) can be solved via SCP as a sequence of SOCPs. The method is efficient and reliable, even for extremely sensitive cases like McCue's orbit transfer problem (McCue, 1967). Using similar convexification techniques, a 3D minimum-fuel OTI problem was considered in Wang & Grant (2018b). Similarly, a 2D minimum-time OTI problem was studied in Wang & Grant (2018c), where the dynamics were parametrized by transfer angle (i.e., orbit true anomaly) instead of time as the independent variable. Both works consider circular orbits, where (79f) can be linearized using spherical or polar coordinates. Tang, Jiang, & Li (2018) solved a minimum-fuel orbit transfer problem by combining SCP with lossless convexification and pseudospectral discretization. Song & Gong (2019) studied a minimum-time interplanetary solar sail mission, where the thrust is replaced by solar radiation force, and optimized the trajectory via SCP as a sequence of SOCP problems.

The above paragraph mentions works that deal mainly with orbit transfer. A companion problem is that of orbit injection, where the vehicle is taken from a non-orbiting state to a target orbit. This occurs, for instance, in the last stage of rocket ascent. Liu & Lu (2014) showed that Problem (79) can also model the optimal exo-atmospheric ascent

flight of a medium-lift launch vehicle. In this case, the initial condition (79e) typically denotes burnout of the launch vehicle's previous stage. In Liu & Lu (2014), constraint (79f) denotes the radius and velocity at the perigee of the target circular orbit. Li, Guan, Wei, & Hu (2019) considered a similar optimal ascent problem where the thrust magnitude is constant, and constraint (79f) describes the orbital elements of a general elliptical orbit. Using pseudospectral discretization and SCP, this optimal ascent problem is solved as a sequence of SOCPs. Li et al. (2020) further considered optimal ascent flight in the case of a power system fault. In this case, depending on the severity of the fault, (79f) describes progressively relaxed insertion constraints.

Mission planning often sits one layer above the OTI problem. For example, a mission plan may consist of a series of planetary flyby and gravity assist maneuvers. A mission, then, can be viewed as a sequence of OTI solutions that minimizes an overall objective such as fuel usage or travel time. A modern approach to mission planning is through hybrid optimal control, and some methods were already mentioned at the end of Section 3.5.1 (Ross & D'Souza, 2005; Stevens & Ross, 2005; Stevens et al., 2004). Evolutionary optimization using genetic algorithms offers an alternative solution for mission planning (Conway, 2014). This approach was used to plan several complex missions: a Galileo-type mission from Earth to Jupiter, a Cassini-type mission from Earth to Saturn, and an OSIRIS-REx type mission from Earth to the asteroid Bennu (Englander, Conway, & Williams, 2012). The Saturn mission is almost identical to that used by the actual NASA/ESA Cassini mission, but is obtained fully automatically at a fraction of the time and cost. The algorithm, known as the evolutionary mission trajectory generator (EMTG), has been made available by NASA Goddard as an open-source software package (Englander, 2020).

4. Outlook

This paper surveyed promising convex optimization-based techniques for next generation space vehicle control systems. We touched on planetary rocket landing, small body landing, spacecraft rendezvous, attitude reorientation, orbit transfer, and endo-atmospheric flight including ascent and reentry. The discussion topics were chosen with a particular sensitivity towards computational efficiency and guaranteed functionality, which are questions of utmost importance for spaceflight control. We conclude by listing in Section 4.1 some of the most popular optimization software now available to the controls engineer, and outlining in Section 4.2 future research directions to which the reader may wish to contribute.

4.1. Optimization software

Success in any computational engineering discipline owes in large part to the availability of good software. Table 1 lists modern optimization software packages that facilitate the implementation of algorithms discussed in Section 3. This list is by no means complete, and should be understood to merely indicate some of the popular optimization software packages that are quite mature and already available today.

4.2. Future directions

We conclude this survey paper by listing some interesting and important future directions for optimization-based space vehicle control.

4.2.1. Guaranteed performance

When proposing a new control algorithm for a real system, it is sobering to remember that the vehicle's survival, along with that of its occupants, literally hangs in the balance (Stein, 2003). The modern controls engineer has immense responsibility both to mission success and to upholding the foundation of trust created by the high reliability of traditional control methods. If we cannot guarantee an equal or greater level of reliability, then new optimization-based control methods will quite certainly be relegated to a ground support role (Ploen et al., 2006).

Consequently, a direction of great importance for optimization-based space vehicle control is to rigorously certify that optimization-based algorithms converge to solutions that yield safe and robust operation in the real world. Active research is being done in the area, but general results are limited and many promising optimization-based methods lack proper guarantees. Today, researchers are looking at real-time performance (Malyuta et al., 2020; Reynolds, Malyuta, Mesbahi, Açıkmeşe and Carson III, 2020), optimality (Reynolds & Mesbahi, 2020b), and convergence rates (Bonalli et al., 2019; Mao, Szmuk et al., 2018). Perhaps the most important yet difficult guarantee is that the algorithm terminates in finite time, which is imperative for control. In the convex setting, algorithms with guaranteed convergence are available and have been flight-tested (Dueri et al., 2017; Scharf et al., 2017), so one direction to explore is how to convexify more general types of nonlinearity (Lee & Mesbahi, 2014; Liu & Lu, 2014; Malyuta & Açıkmeşe, 2020b). For more difficult nonlinearities that are not convexifiable, an emerging subject of *funnel libraries* is being investigated (Açıkmeşe et al., 2008; Majumdar & Tedrake, 2017; Reynolds, 2020; Reynolds, Malyuta, Mesbahi and Açıkmeşe, 2020). The idea, akin to explicit MPC, is to pre-compute a lookup table of trajectories and invariant controllers in order to replace onboard optimization with a search algorithm followed by, in some cases, numerical integration. This can result in a substantially simpler onboard implementation at the expense of a higher storage memory footprint.

4.2.2. Machine learning

Impressive advances in machine learning, and particularly in reinforcement learning (RL), could not side-step space vehicle control without due consideration (Tsiotras & Mesbahi, 2017). The main advantage of RL is that it is able to optimize over a stochastic data stream rather than assuming a particular description of a dynamic model (Arulkumaran, Deisenroth, Brundage, & Bharath, 2017; Buşoniu, de Bruin, Tolić, Kober, & Palunko, 2018). As an optimization tool for nonlinear stochastic systems, it is not surprising that the RL method is attractive for aerospace control.

Although RL for space vehicle control is less than a decade old, a certain amount of literature is now available that addresses almost all of the applications presented in Section 3. The reader is referred to Izzo, Märtens, & Pan (2019) for a dedicated survey. In powered descent guidance, Cheng, Wang, & Jiang (2019) use deep RL (DRL) for lunar landing, Furfaro, Scorsoglio, Linares, & Massari (2020) improve ZEM/ZEV guidance via DRL, and Gaudet & Linares (2019) use recursive RL for Mars landing. For spacecraft rendezvous, Scorsoglio, Furfaro, Linares, & Massari (2019) use actor-critic RL (ACRL) in near-rectilinear orbits, Gaudet, Linares, & Furfaro (2018) consider cluttered environments, and Doerr, Linares, & Furfaro (2020), Linares & Raquepas (2018) use inverse RL to learn the target's behavior. In reentry guidance, Shi & Wang (2020) aim for real-time computation by training a deep neural network (DNN) to learn the functional relationship between state–action pairs obtained from a high-fidelity optimizer. Alternatively, Cheng, Jiang, Wang and Li (2020) use a DNN to provide a numerical predictor–corrector guidance algorithm with a range prediction based on the current vehicle state. This method improves runtime performance by replacing traditional propagation-based trajectory prediction with a neural network. A different line of work is presented in Jin, Chen, Sheng, & Liu (2016), where the attitude of a reentry vehicle with model uncertainty and external disturbances is controlled by a robust adaptive fuzzy PID-type sliding mode controller designed using a radial basis function neural network. For small body landing, Gaudet, Linares, & Furfaro (2019, 2020a, 2020b) use RL meta-learning for greater adaptability, and Cheng, Wang, Song and Jiang (2020) train several DNNs to approximate a nonlinear gravity field as well as the optimal solution obtained using an indirect method. Another interesting

Table 1
Summary of popular optimization software packages. The columns *Direct* and *Indirect* specify which solution method the software uses, as discussed in Section 2.1. The column *Real-time* denotes if the software is destined for real-time onboard use. *Open-source* identifies free-to-download packages with viewable source code. *Pseudospectral* identifies software that is compatible with pseudospectral discretization. *Class* describes the most general class of problems that the software can solve. *Back-end* lists which low-level optimizers are used, and *Language* lists the implementation and front-end interface languages of the package. Certain classifications that do not apply to the "generic parsers" software category are indicated by an empty cell background.

Name	Reference	Direct	Indirect	Real-time	Open-source	Pseudo-spectral	Class	Back-end	Language
Generic parsers									
JuMP	(Dunning et al., 2017)				●		MINLP	▼	Julia
CVX	(Grant and Boyd, 2008)				●		MICP	‖	§
YALMIP	(Löfberg, 2004)				●		MINLP	♮	MATLAB
AMPL	(Fourer et al., 1990)						MINLP	∇	+
Optimal control problem parsers									
SCP Toolbox	(Malyuta et al., 2021)	●			●		NLP	‖	Julia
CasADi	(Andersson et al., 2019)	●	●	●	●	●	NLP	†	▲
SPARTAN	(Sagliano, 2019)	●	●			●	NLP	∗	MATLAB
MISER	(Jennings et al., 2015)	●	●				NLP	‡	MATLAB
GPOPS-II	(Patterson and Rao, 2014)	●	●			●	NLP	∗	MATLAB
DIDO	(Ross and Karpenko, 2012)	●	●			●	NLP	∘	MATLAB
HamPath	(Caillau et al., 2012)		●		●		NLP	++	⊕
ACADO	(Houska et al., 2010)	●		●	●		NLP	¶	∗∗
PSOPT	(Becerra, 2010)	●			●	●	NLP	∗	C++
EZopt	(Shen et al., 2010)	●					NLP	SNOPT	■
GESOP	(Topcu et al., 2007)	●					NLP	□	Δ
Generic solvers									
COSMO	(Garstka et al., 2019)	●			●		SDP	⋆	Julia
FORCES	(Zanelli et al., 2017)	●		●			NLP	IPM	C
CVXGEN	(Mattingley and Boyd, 2011)	●		●	●		QP	IPM	C
Spaceflight-specific solvers									
G-FOLD	(Scharf et al., 2017)	●		●			SOCP	Bsocp	C

▼ ECOS, Gurobi, MOSEK, CPLEX, SDPT3, SeDuMi, IPOPT, SNOPT, COSMO, and more.
‖ ECOS, Gurobi, MOSEK, CPLEX, SDPT3, SeDuMi, and more. Varies by implementation language.
§ MATLAB (CVX), Python (CVXPY), Julia (Convex.jl), and R (CVXR).
♮ All from ‖ as well as IPOPT, KNITRO, SNOPT, and more.
∇ CBC, CPLEX, FortMP, Gurobi, MINOS, IPOPT, SNOPT, KNITRO, and more.
+ An algebraic modelling language influenced by C and AWK.
† IPOPT, qpOASES, OOQP, CPLEX, Gurobi, and others.
▲ Written in C++ with Python, C++, MATLAB/Octave interfaces available.
∗ IPOPT, SNOPT.
‡ Sequential quadratic programing (SQP) solver NLPQL, FFSQP, or NLPQLP.
∘ Fast, spectral Hamiltonian programming (Ross and Gong, 2008).
++ ODE numerical integration and a shooting method.
⊕ Written in Fortran 90 with Python and MATLAB/Octave interfaces available.
¶ Internal SQP-type methods as well as interface to external solver qpOASES.
∗∗ Written in C++ with a MATLAB interface available.
■ Written in Fortran 90 with a MATLAB interface available.
□ SLLSQP, SNOPT, and others.
Δ Fortran 77, Ada 95, C, and MATLAB.
⋆ Conic operator splitting method (based on ADMM).

approach was proposed in Cheng, Wang, Jiang and Li (2020), where DNNs are used to supply good costate initial guesses, while an accurate trajectory is obtained by a downstream shooting method and a homotopy process. In orbit insertion and transfer applications, Cheng, Wang, Jiang and Zhou (2019) develop a multiscale DNN architecture to approximate the optimal solution for a solar sail mission, Holt, Armellin, Baresi, Hashida, Turconi, Scorsoglio, & Furfaro (2021) use ACRL for low-thrust trajectory optimization under changing dynamics, LaFarge, Miller, Howell, & Linares (2020) use RL for libration point transfer in lunar applications, and Miller, Englander, & Linares (2019), Miller & Linares (2019) use proximal policy optimization.

A promising modern direction for spacecraft trajectory RL is to learn a small number of "behind the scenes" parameters (called *solution hyperparameters*) that govern the optimal solution, instead of directly learning the high-dimensional optimal state-input map. Most importantly, the relationship between these parameters and the control policy is much more predictable, and hence can be learned more easily and with less training data. This survey paper makes it clear that most if not all spaceflight trajectory generation problems can be formulated as a variant of the optimal control Problem (4). Hence, the solution hyperparameters are often the maximum principle costates, or combinations thereof, that completely define the optimal control policy. Among these, we find aforementioned concepts of a *primer vector* (Açıkmeşe & Ploen, 2007; Lawden, 1963; Lu & Pan, 2010), and switching functions for bang–bang control (Taheri et al., 2020a). This RL approach was shown to be effective for 3-DoF PDG in You, Wan, Dai, Lu and Rea (2020) and You, Wan, Dai and Rea (2020), where the authors learned 10 hyperparameters instead of the map from a 7D state to a 3D input. Most importantly, only $\approx 10^3$ training trajectories were required. In comparison, the state-input map learning approach of Sánchez-Sánchez & Izzo (2018) also achieved good results, but required $\approx 10^7$ training samples. A slightly different approach was taken for 3-DoF small body landing in Cheng, Wang et al. (2020), where homotopy and coordinate transforms were used to learn a 5D costate vector instead of the map from a 7D state to a 3D input. The DNN's output was then used to provide accurate initial guesses and to improve the convergence of a downstream shooting method. To summarize, the fact that learning hyperparameters works better than learning the optimal state-input mapping is just an observation of the fact that application domain knowledge can go a long way towards improving learning performance (Tabuada & Fraile, 2020). In the case of spacecraft trajectory optimization, this knowledge often comes from applying Pontryagin's maximum principle.

As discussed in Section 4.2.1, performance guarantees for an RL-based controller will have to be provided before serious onboard consideration. This may be harder to achieve for RL, since controllers are typically based on neural networks whose out-of-sample performance is still very difficult to characterize. Nevertheless, RL and other machine learning approaches are appealing for adaptive control systems. Future research will likely see the aerospace control community search for the right opportunities where RL can be embedded to improve traditional control systems.

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.

Acknowledgments

This work is funded in part by the National Science Foundation (Grants CMMI-1613235 and ECCS-1931744) and the Office of Naval Research (Grant N00014-20-1-2288). The authors would like to extend their gratitude to Yuanqi Mao for his helpful inputs on sequential convex programming algorithms, and to Taylor P. Reynolds for his invaluable review of an early draft and for his knowledge of space vehicle control at large.

References

Açıkmeşe, B., Aung, M., Casoliva, J., Mohan, S., Johnson, A., Scharf, D., et al. (2013). Flight testing of trajectories computed by G-FOLD: Fuel optimal large divert guidance algorithm for planetary landing. In *23rd AAS/AIAA space flight mechanics meeting, Kauai, HI, 2013*. AAS/AIAA.

Açıkmeşe, B., & Blackmore, L. (2011). Lossless convexification of a class of optimal control problems with non-convex control constraints. *Automatica*, *47*(2), 341–347. http://dx.doi.org/10.1016/j.automatica.2010.10.037.

Açıkmeşe, B., Blackmore, J. C. L., & Scharf, D. P. (2013). *Method and apparatus for powered descent guidance*. United States Patent Classification (USPC) System, US Patent 8, 489, 260 B2.

Açıkmeşe, B., Carson III, J. M., & Blackmore, L. (2013). Lossless convexification of nonconvex control bound and pointing constraints of the soft landing optimal control problem. *IEEE Transactions on Control Systems Technology*, *21*(6), 2104–2113. http://dx.doi.org/10.1109/tcst.2012.2237346.

Açıkmeşe, B., Casoliva, J., Carson III, J. M., & Blackmore, L. (2012). G-FOLD: A real-time implementable fuel optimal large divert guidance algorithm for planetary pinpoint landing. In *Concepts and approaches for mars exploration, Vol. 1679* (p. 4193).

Açıkmeşe, B., & Ploen, S. (2005). A powered descent guidance algorithm for Mars pinpoint landing. In *AIAA guidance, navigation, and control conference and exhibit*. American Institute of Aeronautics and Astronautics, http://dx.doi.org/10.2514/6.2005-6288.

Açıkmeşe, B., & Ploen, S. R. (2007). Convex programming approach to powered descent guidance for Mars landing. *Journal of Guidance, Control, and Dynamics*, *30*(5), 1353–1366. http://dx.doi.org/10.2514/1.27553.

Açıkmeşe, B., Scharf, D., Blackmore, L., & Wolf, A. (2008). Enhancements on the convex programming based powered descent guidance algorithm for Mars landing. In *AIAA/AAS astrodynamics specialist conference and exhibit*. American Institute of Aeronautics and Astronautics, http://dx.doi.org/10.2514/6.2008-6426.

Achterberg, T. (2007). *Constrained integer programming* (Ph.D. thesis), Berlin: Technische Universität Berlin.

Achterberg, T., & Wunderling, R. (2013). Mixed integer programming: Analyzing 12 years of progress. In *Facets of combinatorial optimization* (pp. 449–481). Springer Berlin Heidelberg, http://dx.doi.org/10.1007/978-3-642-38189-8_18.

Adkins, C. N. (1970). Optimization of multistage rockets including drag. *Journal of Spacecraft and Rockets*, *7*(6), 751–755. http://dx.doi.org/10.2514/3.30032.

Agamawi, Y. M., & Rao, A. V. (2020). Comparison of derivative estimation methods in optimal control using direct collocation. *AIAA Journal*, *58*(1), 341–354. http://dx.doi.org/10.2514/1.j058514.

Alessio, A., & Bemporad, A. (2009). A survey on explicit model predictive control. In *Nonlinear model predictive control* (pp. 345–369). Springer Berlin Heidelberg, http://dx.doi.org/10.1007/978-3-642-01094-1_29.

Alfano, S., & Thorne, J. D. (1994). Circle-to-circle constant-thrust orbit raising. *Journal of the Astronautical Sciences*, *42*(1), 35–45.

Andersson, J. A. E., Gillis, J., Horn, G., Rawlings, J. B., & Diehl, M. (2019). CasADi – A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, *11*(1), 1–36. http://dx.doi.org/10.1007/s12532-018-0139-4.

Antsaklis, P. J., & Michel, A. N. (2006). *Linear systems*. Birkhauser Basel, http://dx.doi.org/10.1007/0-8176-4435-0.

Aoyama, Y., Boutselis, G., Patel, A., & Theodorou, E. A. (2020). Constrained differential dynamic programming revisited. arXiv:2005.00985.

Ariba, Y., Arzelier, D., & Urbina-Iglesias, L. S. (2018). Minimum-fuel fixed-time impulsive elliptic glide-slope guidance algorithms using semidefinite programming. *Journal of Guidance, Control, and Dynamics*, *41*(9), 1873–1887. http://dx.doi.org/10.2514/1.g003395.

Armellin, R. (2021). Collision avoidance maneuver optimization with a multiple-impulse convex formulation. *Acta Astronautica*, *186*, 347–362. http://dx.doi.org/10.1016/j.actaastro.2021.05.046.

Arrieta-Camacho, J. J., & Biegler, L. T. (2005). Real time optimal guidance of low-thrust spacecraft: An application of nonlinear model predictive control. *Annals of the New York Academy of Sciences*, *1065*(1), 174–188. http://dx.doi.org/10.1196/annals.1370.001.

Arulkumaran, K., Deisenroth, M. P., Brundage, M., & Bharath, A. A. (2017). Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine*, *34*(6), 26–38. http://dx.doi.org/10.1109/msp.2017.2743240.

Arya, V., Taheri, E., & Junkins, J. L. (2020). A composite framework for co-optimization of spacecraft trajectory and propulsion system. *Acta Astronautica*, *178*, 773–782. http://dx.doi.org/10.1016/j.actaastro.2020.10.007.

Arzelier, D., Kara-Zaitri, M., Louembet, C., & Delibasi, A. (2011). Using polynomial optimization to solve the fuel-optimal linear impulsive rendezvous problem. *Journal of Guidance, Control, and Dynamics*, *34*(5), 1567–1576. http://dx.doi.org/10.2514/1.52227.

Arzelier, D., Louembet, C., Rondepierre, A., & Kara-Zaitri, M. (2013). A new mixed iterative algorithm to solve the fuel-optimal linear impulsive rendezvous problem. *Journal of Optimization Theory and Applications*, *159*(1), 210–230. http://dx.doi.org/10.1007/s10957-013-0282-z.

Augugliaro, F., Schoellig, A. P., & D'Andrea, R. (2012). Generation of collision-free trajectories for a quadrocopter fleet: A sequential convex programming approach. In *2012 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, http://dx.doi.org/10.1109/iros.2012.6385823.

Bai, C., Guo, J., & Zheng, H. (2019). Optimal guidance for planetary landing in hazardous terrains. *IEEE Transactions on Aerospace and Electronic Systems*, 1. http://dx.doi.org/10.1109/taes.2019.2955785.

Baldwin, M., Erwin, R. S., & Kolmanovsky, I. (2013). Robust controller for constrained relative motion maneuvering with disturbance rejection. In *AIAA guidance, navigation, and control (GNC) conference*. American Institute of Aeronautics and Astronautics, http://dx.doi.org/10.2514/6.2013-4721.

Becerra, V. M. (2010). Solving complex optimal control problems at no cost with PSOPT. In *2010 IEEE International Symposium on Computer-Aided Control System Design*. IEEE, http://dx.doi.org/10.1109/cacsd.2010.5612676.

Bemporad, A., & Morari, M. (1999). Control of systems integrating logic, dynamics, and constraints. *Automatica*, *35*(3), 407–427. http://dx.doi.org/10.1016/s0005-1098(98)00178-2.

Bemporad, A., & Morari, M. (2007). Robust model predictive control: A survey. In *Robustness in identification and control* (pp. 207–226). Springer London, http://dx.doi.org/10.1007/bfb0109870.

Bemporad, A., Morari, M., Dua, V., & Pistikopoulos, E. N. (2002). The explicit linear quadratic regulator for constrained systems. *Automatica*, *38*(1), 3–20. http://dx.doi.org/10.1016/s0005-1098(01)00174-1.

Benedikter, B., Zavoli, A., & Colasurdo, G. (2019a). A convex approach to rocket ascent trajectory optimization. In *Proceedings of the 8th European conference for aeronautics and space sciences. Madrid, Spain, 1-4 july 2019*. http://dx.doi.org/10.13009/EUCASS2019-430.

Benedikter, B., Zavoli, A., & Colasurdo, G. (2019b). A convex optimization approach for finite-thrust time-constrained cooperative rendezvous. arXiv:1909.09443.

Benedikter, B., Zavoli, A., Colasurdo, G., Pizzurro, S., & Cavallini, E. (2020). Convex optimization of launch vehicle ascent trajectory with heat-flux and splash-down constraints. arXiv:2008.13239.

Berkovitz, L. D. (1974). *Optimal control theory*. New York: Springer, http://dx.doi.org/10.1007/978-1-4757-6097-2.

Berrut, J.-P., & Trefethen, L. N. (2004). Barycentric Lagrange interpolation. *SIAM Review*, *46*(3), 501–517. http://dx.doi.org/10.1137/s0036144502417715.

Berry, K., Sutter, B., May, A., Williams, K., Barbee, B. W., Beckman, M., et al. (2013). OSIRIS-REx touch-and-go (TAG) mission design and analysis. *Advances in the Astronautical Sciences*, 667–678.

Betts, J. T. (1998). Survey of numerical methods for trajectory optimization. *Journal of Guidance, Control, and Dynamics*, *21*(2), 193–207. http://dx.doi.org/10.2514/2.4231.

Betts, J. T. (2000). Very low-thrust trajectory optimization using a direct SQP method. *Journal of Computational and Applied Mathematics*, *120*(1–2), 27–40. http://dx.doi.org/10.1016/s0377-0427(00)00301-0.

Betts, J. T. (2010). *Practical methods for optimal control and estimation using nonlinear programming*. Society for Industrial and Applied Mathematics, http://dx.doi.org/10.1137/1.9780898718577.

Betts, J. T., & Erb, S. O. (2003). Optimal low thrust trajectories to the Moon. *SIAM Journal on Applied Dynamical Systems*, *2*(2), 144–170. http://dx.doi.org/10.1137/s1111111102409080.

Blackmore, L. (2016). Autonomous precision landing of space rockets. *The Bridge*, *4*(46), 15–20.

Blackmore, L., Açıkmeşe, B., & Carson III, J. M. (2012). Lossless convexification of control constraints for a class of nonlinear optimal control problems. *Systems & Control Letters*, *61*(8), 863–870. http://dx.doi.org/10.1016/j.sysconle.2012.04.010.

Blackmore, L., Açikmeşe, B., & Scharf, D. P. (2010). Minimum-landing-error powered-descent guidance for Mars landing using convex optimization. *Journal of Guidance, Control, and Dynamics*, *33*(4), 1161–1171. http://dx.doi.org/10.2514/1.47202.

Bonalli, R., Cauligi, A., Bylard, A., & Pavone, M. (2019). GuSTO: Guaranteed sequential trajectory optimization via sequential convex programming. In *2019 international conference on robotics and automation (ICRA)*. IEEE, http://dx.doi.org/10.1109/icra.2019.8794205.

Bonalli, R., Hérissé, B., & Trélat, E. (2017). Analytical initialization of a continuation-based indirect method for optimal control of endo-atmospheric launch vehicle systems. *IFAC-PapersOnLine*, *50*(1), 482–487. http://dx.doi.org/10.1016/j.ifacol.2017.08.095.

Bonalli, R., Lew, T., & Pavone, M. (2021). Analysis of theoretical and numerical properties of sequential convex programming for continuous-time optimal control. *IEEE Transactions on Automatic Control (in review)*, https://arxiv.org/abs/2009.05038.

de Boor, C., & Swartz, B. (1973). Collocation at Gaussian points. *SIAM Journal on Numerical Analysis*, *10*(4), 582–606. http://dx.doi.org/10.1137/0710052.

Borrelli, F., Bemporad, A., & Morari, M. (2017). *Predictive control for linear and hybrid systems*. Cambridge University Press, http://dx.doi.org/10.1017/9781139061759.

Boyd, J. P. (1989). *Chebyshev and fourier spectral methods* (2nd ed.). Springer-Verlag.

Boyd, S., Ghaoui, L. E., Feron, E., & Balakrishnan, V. (1994). *Linear matrix inequalities in system and control theory*. Society for Industrial and Applied Mathematics, http://dx.doi.org/10.1137/1.9781611970777.

Boyd, S., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge, UK: Cambridge University Press.

Braun, R., & Manning, R. (2006). Mars exploration entry, descent and landing challenges. In *2006 IEEE aerospace conference*. IEEE, http://dx.doi.org/10.1109/aero.2006.1655790.

Breger, L., & How, J. P. (2008). Safe trajectories for autonomous rendezvous of spacecraft. *Journal of Guidance, Control, and Dynamics*, *31*(5), 1478–1489. http://dx.doi.org/10.2514/1.29590.

Brown, K., & Johnson, G. (1967). Real-time optimal guidance. *IEEE Transactions on Automatic Control*, *12*(5), 501–506. http://dx.doi.org/10.1109/tac.1967.1098718.

Bryson Jr., A. E., & Ho, Y.-C. (1975). *Applied optimal control*. Washington, D.C.: Hemisphere Publishing Corporation.

Buşoniu, L., de Bruin, T., Tolić, D., Kober, J., & Palunko, I. (2018). Reinforcement learning for control: Performance, stability, and deep approximators. *Annual Reviews in Control*, *46*, 8–28. http://dx.doi.org/10.1016/j.arcontrol.2018.09.005.

Butcher, J. C. (2016). *Numerical methods for ordinary differential equations*. John Wiley & Sons, Ltd, http://dx.doi.org/10.1002/9781119121534.

Caillau, J.-B., Cots, O., & Gergaud, J. (2012). Differential continuation for regular optimal control problems. *Optimization Methods & Software*, *27*(2), 177–196. http://dx.doi.org/10.1080/10556788.2011.593625.

Caillau, J.-B., Daoud, B., & Gergaud, J. (2012). Minimum fuel control of the planar circular restricted three-body problem. *Celestial Mechanics & Dynamical Astronomy*, *114*(1–2), 137–150. http://dx.doi.org/10.1007/s10569-012-9443-x.

Calise, A. J., & Brandt, N. (2004). Generation of launch vehicle abort trajectories using a hybrid optimization method. *Journal of Guidance, Control, and Dynamics*, *27*(6), 929. http://dx.doi.org/10.2514/1.7989.

Calise, A. J., Melamed, N., & Lee, S. (1998). Design and evaluation of a three-dimensional optimal ascent guidance algorithm. *Journal of Guidance, Control, and Dynamics*, *21*(6), 867–875. http://dx.doi.org/10.2514/2.4350.

Carson III, J., & Açıkmeşe, B. (2006). A model-predictive control technique with guaranteed resolvability and required thruster silent times for small-body proximity operations. In *AIAA guidance, navigation, and control conference and exhibit*. American Institute of Aeronautics and Astronautics, http://dx.doi.org/10.2514/6.2006-6780.

Carson III, J. M., Açıkmeşe, B., & Blackmore, L. (2011). Lossless convexification of powered-descent guidance with nonconvex thrust bound and pointing constraints. In *Proceedings of the 2011 American control conference*. IEEE, http://dx.doi.org/10.1109/acc.2011.5990959.

Carson III, J. M., Açıkmeşe, B., Blackmore, L., & Wolf, A. A. (2011). Capabilities of convex powered-descent guidance algorithms for pinpoint and precision landing. In *2011 aerospace conference*. IEEE, http://dx.doi.org/10.1109/aero.2011.5747244.

Carson III, J., Açıkmeşe, B., Murray, R., & MacMynowski, D. (2008). Robust model predictive control with a safety mode: Applied to small-body proximity operations. In *AIAA guidance, navigation and control conference and exhibit*. American Institute of Aeronautics and Astronautics, http://dx.doi.org/10.2514/6.2008-7243.

Carson III, J. M., Munk, M. M., Sostaric, R. R., Estes, J. N., Amzajerdian, F., Blair, J. B., et al. (2019). The SPLICE project: Continuing NASA development of GN&C technologies for safe and precise landing. In *AIAA SciTech 2019 forum*. American Institute of Aeronautics and Astronautics, http://dx.doi.org/10.2514/6.2019-0660.

Cerf, M. (2016). Low-thrust transfer between circular orbits using natural precession. *Journal of Guidance, Control, and Dynamics*, *39*(10), 2232–2239. http://dx.doi.org/10.2514/1.g001331.

Cerf, M., Haberkorn, T., & Trélat, E. (2011). Continuation from a flat to a round earth model in the coplanar orbit transfer problem. *Optimal Control Applications & Methods*, *33*(6), 654–675. http://dx.doi.org/10.1002/oca.1016.

Cesari, L. (1983). *Optimization – Theory and applications*. New York: Springer-Verlag.

Chandler, D. C., & Smith, I. E. (1967). Development of the iterative guidance mode with its application to various vehicles and missions. *Journal of Spacecraft and Rockets*, *4*(7), 898–903. http://dx.doi.org/10.2514/3.28985.

Chawla, C., Sarmah, P., & Padhi, R. (2010). Suboptimal reentry guidance of a reusable launch vehicle using pitch plane maneuver. *Aerospace Science and Technology*, *14*(6), 377–386. http://dx.doi.org/10.1016/j.ast.2010.04.001.

Chen, G., Cai, P., Wang, Y., Zhang, L., & Liang, J. (2019). Trajectory optimization for asteroid landing considering gravitational orbit-attitude coupling. *IEEE Access*, *7*, 126464–126478. http://dx.doi.org/10.1109/access.2019.2938821.

Cheng, L., Jiang, F., Wang, Z., & Li, J. (2020). Multi-constrained real-time entry guidance using deep neural networks. *IEEE Transactions on Aerospace and Electronic Systems*, http://dx.doi.org/10.1109/taes.2020.3015321.

Cheng, X., Li, H., & Zhang, R. (2017). Efficient ascent trajectory optimization using convex models based on the Newton-Kantorovich/Pseudospectral approach. *Aerospace Science and Technology*, *66*, 140–151. http://dx.doi.org/10.1016/j.ast.2017.02.023.

Cheng, L., Wang, Z., & Jiang, F. (2019). Real-time control for fuel-optimal moon landing based on an interactive deep reinforcement learning algorithm. *Astrodynamics*, *3*(4), 375–386. http://dx.doi.org/10.1007/s42064-018-0052-2.

Cheng, L., Wang, Z., Jiang, F., & Li, J. (2020). Fast generation of optimal asteroid landing trajectories using deep neural networks. *IEEE Transactions on Aerospace and Electronic Systems*, *56*(4), 2642–2655. http://dx.doi.org/10.1109/taes.2019.2952700.

Cheng, L., Wang, Z., Jiang, F., & Zhou, C. (2019). Real-time optimal control for spacecraft orbit transfer via multiscale deep neural networks. *IEEE Transactions on Aerospace and Electronic Systems*, *55*(5), 2436–2450. http://dx.doi.org/10.1109/taes.2018.2889571.

Cheng, L., Wang, Z., Song, Y., & Jiang, F. (2020). Real-time optimal control for irregular asteroid landings using deep neural networks. *Acta Astronautica*, *170*, 66–79. http://dx.doi.org/10.1016/j.actaastro.2019.11.039.

Cheng, L., Wen, H., & Jin, D. (2019). Uncertain parameters analysis of powered-descent guidance based on Chebyshev interval method. *Acta Astronautica*, *162*, 581–588. http://dx.doi.org/10.1016/j.actaastro.2019.05.040.

Cherry, G. (1964). A general, explicit, optimizing guidance law for rocket-propelled spaceflight. In *Astrodynamics guidance and control conference*. American Institute of Aeronautics and Astronautics, http://dx.doi.org/10.2514/6.1964-638.

Clohessy, W. H., & Wiltshire, R. S. (1960). Terminal guidance system for satellite rendezvous. *Journal of the Aerospace Sciences*, *27*(9), 653–658. http://dx.doi.org/10.2514/8.8704.

Colombo, C., Vasile, M., & Radice, G. (2009). Optimal low-thrust trajectories to asteroids through an algorithm based on differential dynamic programming. *Celestial Mechanics & Dynamical Astronomy*, *105*(1–3), 75–112. http://dx.doi.org/10.1007/s10569-009-9224-3.

Conn, A. R., Gould, N. I. M., & Toint, P. L. (2000). *Trust region methods*. Society for Industrial and Applied Mathematics, http://dx.doi.org/10.1137/1.9780898719857.

Conway, B. A. (2011). A survey of methods available for the numerical optimization of continuous dynamic systems. *Journal of Optimization Theory and Applications*, *152*(2), 271–306. http://dx.doi.org/10.1007/s10957-011-9918-z.

Conway, B. A. (2014). *Spacecraft trajectory optimization*. Cambridge, UK: Cambridge University Press.

Cook, W. J., Cunningham, W. H., Pulleyblank, W. R., & Schrijver, A. (1998). *Combinatorial optimization*. Hoboken, NJ: Wiley.

Cottle, R. W., Pang, J.-S., & Stone, R. E. (2009). *The linear complementarity problem*. Society for Industrial and Applied Mathematics, http://dx.doi.org/10.1137/1.9780898719000.

Crane, L. (2019). The final frontier. *New Scientist*, *241*(3218), 20–22. http://dx.doi.org/10.1016/s0262-4079(19)30327-6.

Cui, P.-Y., Gao, A., & Cui, H.-T. (2012). Receding horizon-based dual control strategy for pinpoint planetary landing. *Transactions of the Japan Society for Aeronautical and Space Sciences*, *55*(4), 222–228. http://dx.doi.org/10.2322/tjsass.55.222.

Cui, P., Liu, Y., Yu, Z., Zhu, S., & Shao, W. (2017). Intelligent landing strategy for the small bodies: from passive bounce to active trajectory control. *Acta Astronautica*, *137*, 232–242. http://dx.doi.org/10.1016/j.actaastro.2017.04.033.

Darby, C. L., Hager, W. W., & Rao, A. V. (2010). An *hp*-adaptive pseudospectral method for solving optimal control problems. *Optimal Control Applications & Methods*, *32*(4), 476–502. http://dx.doi.org/10.1002/oca.957.

de Ruiter, A. H. J., Damaren, C. J., & Forbes, J. R. (2013). *Spacecraft dynamics and control: An introduction*. Wiley.

Deaconu, G., Louembet, C., & Théron, A. (2015). Designing continuously constrained spacecraft relative trajectories for proximity operations. *Journal of Guidance, Control, and Dynamics*, *38*(7), 1208–1217. http://dx.doi.org/10.2514/1.g000283.

Detra, R. W., Kemp, N. H., & Riddell, F. R. (1957). Addendum to "Heat Transfer to Satellite Vehicles Re-entering the Atmosphere". *Jet Propulsion*, *27*(12), 1256–1257. http://dx.doi.org/10.2514/8.12520.

Di Cairano, S., & Kolmanovsky, I. V. (2018). Real-time optimization and model predictive control for aerospace and automotive applications. In *2018 annual American control conference (ACC)*. IEEE, http://dx.doi.org/10.23919/acc.2018.8431585.

Di Cairano, S., Park, H., & Kolmanovsky, I. (2012). Model predictive control approach for guidance of spacecraft rendezvous and proximity maneuvering. *International Journal of Robust and Nonlinear Control*, *22*(12), 1398–1427. http://dx.doi.org/10.1002/rnc.2827.

Di Lizia, P., Armellin, R., Morselli, A., & Bernelli-Zazzera, F. (2013). High order optimal feedback control of space trajectories with bounded control. *Acta Astronautica*, *94*(1), 383–394. http://dx.doi.org/10.1016/j.actaastro.2013.02.011.

Doerr, B. G., Linares, R., & Furfaro, R. (2020). Space objects maneuvering prediction via maximum causal entropy inverse reinforcement learning. In *AIAA SciTech 2020 forum*. American Institute of Aeronautics and Astronautics, http://dx.doi.org/10.2514/6.2020-0235.

Domahidi, A., Chu, E., & Boyd, S. (2013). ECOS: An SOCP solver for embedded systems. In *2013 European control conference (ECC)* (pp. 3071–3076). IEEE, http://dx.doi.org/10.23919/ecc.2013.6669541.

Dong, H., Hu, Q., & Akella, M. R. (2017). Safety control for spacecraft autonomous rendezvous and docking under motion constraints. *Journal of Guidance, Control, and Dynamics*, *40*(7), 1680–1692. http://dx.doi.org/10.2514/1.g002322.

Dong, K., Luo, J., Dang, Z., & Wei, L. (2020). Tube-based robust output feedback model predictive control for autonomous rendezvous and docking with a tumbling target. *Advances in Space Research*, *65*(4), 1158–1181. http://dx.doi.org/10.1016/j.asr.2019.11.014.

Downes, R., & Rose, S. (2001). *Hubble space telescope primer for cycle 11: Technical Report*, Greenbelt, MD: Goddard Space Flight Center.

D'Souza, C. (1997). An optimal guidance law for planetary landing. In *Guidance, navigation, and control conference*. American Institute of Aeronautics and Astronautics, http://dx.doi.org/10.2514/6.1997-3709.

D'Souza, C., Hannak, C., Spehar, P., Clark, F., & Jackson, M. (2007). Orion rendezvous, proximity operations and docking design and analysis. In *AIAA guidance, navigation and control conference and exhibit*. American Institute of Aeronautics and Astronautics, http://dx.doi.org/10.2514/6.2007-6683.

D'Souza, S. N., & Sarigul-Klijn, N. (2014). Survey of planetary entry guidance algorithms. *Progress in Aerospace Sciences*, *68*, 64–74. http://dx.doi.org/10.1016/j.paerosci.2014.01.002.

Dueri, D. A. (2018). *Real-time optimization in aerospace systems* (Ph.D. thesis), Seattle: University of Washington.

Dueri, D., Açıkmeşe, B., Scharf, D. P., & Harris, M. W. (2017). Customized real-time interior-point methods for onboard powered-descent guidance. *Journal of Guidance, Control, and Dynamics*, *40*(2), 197–212. http://dx.doi.org/10.2514/1.g001480.

Dueri, D., Raković, S. V., & Açıkmeşe, B. (2016). Consistently improving approximations for constrained controllability and reachability. In *2016 European control conference (ECC)*. IEEE, http://dx.doi.org/10.1109/ecc.2016.7810523.

Dueri, D., Zhang, J., & Açıkmeşe, B. (2014). Automated custom code generation for embedded, real-time second order cone programming. *IFAC Proceedings Volumes*, *47*(3), 1605–1612. http://dx.doi.org/10.3182/20140824-6-za-1003.02736.

Dunham, W., Petersen, C., & Kolmanovsky, I. (2016). Constrained control for soft landing on an asteroid with gravity model uncertainty. In *2016 American control conference (ACC)*. IEEE, http://dx.doi.org/10.1109/acc.2016.7526585.

Dunning, I., Huchette, J., & Lubin, M. (2017). JuMP: A Modeling language for mathematical optimization. *SIAM Review*, *59*(2), 295–320. http://dx.doi.org/10.1137/15M1020575.

Dwyer-Cianciolo, A. M., Karlgaard, C. D., Woffinden, D., Lugo, R. A., Tynis, J., Sostaric, R. R., et al. (2019). Defining navigation requirements for future missions. In *AIAA SciTech 2019 forum*. American Institute of Aeronautics and Astronautics, http://dx.doi.org/10.2514/6.2019-0661.

Englander, J. A. (2020). Evolutionary mission trajectory generator (EMTG). *GitHub Repository*, https://github.com/JacobEnglander/EMTG.

Englander, J. A., Conway, B. A., & Williams, T. (2012). Automated mission planning via evolutionary algorithms. *Journal of Guidance, Control, and Dynamics*, *35*(6), 1878–1887. http://dx.doi.org/10.2514/1.54101.

Eren, U., Açıkmeşe, B., & Scharf, D. P. (2015). A mixed integer convex programming approach to constrained attitude guidance. In *2015 European control conference (ECC)*. IEEE, http://dx.doi.org/10.1109/ecc.2015.7330690.

Eren, U., Dueri, D., & Açıkmeşe, B. (2015). Constrained reachability and controllability sets for planetary precision landing via convex optimization. *Journal of Guidance, Control, and Dynamics*, *38*(11), 2067–2083. http://dx.doi.org/10.2514/1.g000882.

Eren, U., Prach, A., Koçer, B. B., Raković, S. V., Kayacan, E., & Açıkmeşe, B. (2017). Model predictive control in aerospace systems: Current state and opportunities. *Journal of Guidance, Control, and Dynamics*, *40*(7), 1541–1566. http://dx.doi.org/10.2514/1.g002507.

Team, E. S. (2012). *Europa study 2012 report: Europa lander mission: Technical Report Task Order NMO711062 Outer Planets Flagship Mission*, Jet Propulsion Laboratory.

Fahroo, F., Doman, D., & Ngo, A. (2003a). Footprint generation for reusable launch vehicles using a direct pseudospectral method. In *Proceedings of the 2003 American control conference*. IEEE, http://dx.doi.org/10.1109/acc.2003.1243394.

Fahroo, F., Doman, D., & Ngo, A. (2003b). Modeling issues in footprint generation for reusable vehicles. In *2003 IEEE aerospace conference proceedings (Cat. No.03TH8652)*. IEEE, http://dx.doi.org/10.1109/aero.2003.1235205.

Fahroo, F., & Ross, I. M. (2002). Direct trajectory optimization by a Chebyshev pseudospectral method. *Journal of Guidance, Control, and Dynamics*, *25*(1), 160–166. http://dx.doi.org/10.2514/2.4862.

Fares, B., Noll, D., & Apkarian, P. (2002). Robust control via sequential semidefinite programming. *SIAM Journal on Control and Optimization*, *40*(6), 1791–1820. http://dx.doi.org/10.1137/s0363012900373483.

Faulders, C. R. (1958). Low-thrust rocket steering program for minimum time transfer between planetary orbits. In *SAE technical paper series*. SAE International, http://dx.doi.org/10.4271/580376.

Fehse, W. (2003). *Automated rendezvous and docking of spacecraft*. Cambridge University Press, http://dx.doi.org/10.1017/cbo9780511543388.

Fernandes, S. D. S. (1995). Optimum low-thrust limited power transfers between neighbouring elliptic non-equatorial orbits in a non-central gravity field. *Acta Astronautica*, *35*(12), 763–770. http://dx.doi.org/10.1016/0094-5765(95)00002-h.

Floudas, C. A., & Pardalos, P. M. (2009). *Encyclopedia of optimization* (2nd ed.). Springer.

Forsgren, A., Gill, P. E., & Wright, M. H. (2002). Interior methods for nonlinear optimization. *SIAM Review*, *44*(4), 525–597. http://dx.doi.org/10.1137/s0036144502414942.

Fourer, R., Gay, D. M., & Kernighan, B. W. (1990). A modeling language for mathematical programming. *Management Science*, *36*(5), 519–554. http://dx.doi.org/10.1287/mnsc.36.5.519.

Françolin, C. C., Benson, D. A., Hager, W. W., & Rao, A. V. (2014). Costate approximation in optimal control using integral Gaussian quadrature orthogonal collocation methods. *Optimal Control Applications & Methods*, *36*(4), 381–397. http://dx.doi.org/10.1002/oca.2112.

Frey, G. R., Petersen, C. D., Leve, F. A., Kolmanovsky, I. V., & Girard, A. R. (2017). Constrained spacecraft relative motion planning exploiting periodic natural motion trajectories and invariance. *Journal of Guidance, Control, and Dynamics*, *40*(12), 3100–3115. http://dx.doi.org/10.2514/1.g002914.

Furfaro, R., Scorsoglio, A., Linares, R., & Massari, M. (2020). Adaptive generalized ZEM-ZEV feedback guidance for planetary landing via a deep reinforcement learning approach. *Acta Astronautica*, *171*, 156–171. http://dx.doi.org/10.1016/j.actaastro.2020.02.051.

Furfaro, R., Selnick, S., Cupples, M. L., & Cribb, M. W. (2011). Non-linear sliding guidance algorithms for precision Lunar landing. In *21st AAS/AIAA space flight mechanics meeting* (pp. 945–964). AAS Paper 09-334.

Ganet-Schoeller, M., & Brunel, A. (2019). Optimal guidance for 1st stage launcher recovery. *IFAC-PapersOnLine*, *52*(12), 532–537. http://dx.doi.org/10.1016/j.ifacol.2019.11.298.

García, C. E., Prett, D. M., & Morari, M. (1989). Model predictive control: Theory and practice—A survey. *Automatica*, *25*(3), 335–348. http://dx.doi.org/10.1016/0005-1098(89)90002-2.

Garcia-Sanz, M. (2019). Control co-design: An engineering game changer. *Advanced Control for Applications: Engineering and Industrial Systems*, *1*(1), Article e18. http://dx.doi.org/10.1002/adc2.18.

Garg, D., Patterson, M., Hager, W. W., Rao, A. V., Benson, D. A., & Huntington, G. T. (2010). A unified framework for the numerical solution of optimal control problems using pseudospectral methods. *Automatica*, *46*(11), 1843–1851. http://dx.doi.org/10.1016/j.automatica.2010.06.048.

Garrett, L. B., & Pitts, J. I. (1970). A general transient heat-transfer computer program for thermally thick walls. (NASA-TM-X-2058, L-6850), NASA Langley Research Center Hampton, VA.

Garstka, M., Cannon, M., & Goulart, P. (2019). COSMO: A conic operator splitting method for large convex problems. In *European Control Conference*. http://dx.doi.org/10.23919/ECC.2019.8796161, arXiv:1901.10887.

Gath, P. F., & Calise, A. J. (2001). Optimization of launch vehicle ascent trajectories with path constraints and coast arcs. *Journal of Guidance, Control, and Dynamics*, *24*(2), 296–304. http://dx.doi.org/10.2514/2.4712.

Gaudet, B., & Linares, R. (2019). Adaptive guidance with reinforcement meta-learning. arXiv:1901.04473.

Gaudet, B., Linares, R., & Furfaro, R. (2018). Spacecraft rendezvous guidance in cluttered environments via artificial potential functions and reinforcement learning. In *AAS/AIAA astrodynamics specialist conference*.

Gaudet, B., Linares, R., & Furfaro, R. (2019). Seeker based adaptive guidance via reinforcement meta-learning applied to asteroid close proximity operations. arXiv:1907.06098.

Gaudet, B., Linares, R., & Furfaro, R. (2020a). Six degree-of-freedom body-fixed hovering over unmapped asteroids via LIDAR altimetry and reinforcement meta-learning. *Acta Astronautica*, *172*, 90–99. http://dx.doi.org/10.1016/j.actaastro.2020.03.026.

Gaudet, B., Linares, R., & Furfaro, R. (2020b). Terminal adaptive guidance via reinforcement meta-learning: Applications to autonomous asteroid close-proximity operations. *Acta Astronautica*, *171*, 1–13. http://dx.doi.org/10.1016/j.actaastro.2020.02.036.

Gavilan, F., Vazquez, R., & Camacho, E. F. (2012). Chance-constrained model predictive control for spacecraft rendezvous with disturbance estimation. *Control Engineering Practice*, *20*(2), 111–122. http://dx.doi.org/10.1016/j.conengprac.2011.09.006.

Gil-Fernandez, J., & Gomez-Tierno, M. A. (2010). Practical method for optimization of low-thrust transfers. *Journal of Guidance, Control, and Dynamics*, *33*(6), 1927–1931. http://dx.doi.org/10.2514/1.50739.

Gill, P. E., Murray, W., & Saunders, M. A. (2005). SNOPT: An SQP algorithm for large-scale constrained optimization. *SIAM Review*, *47*(1), 99–131. http://dx.doi.org/10.1137/s0036144504446096.

Gill, P. E., & Wong, E. (2011). Sequential quadratic programming methods. In *Mixed integer nonlinear programming* (pp. 147–224). Springer New York, http://dx.doi.org/10.1007/978-1-4614-1927-3_6.

Gilz, P. R. A., Joldes, M., Louembet, C., & Camps, F. (2019). Stable model predictive strategy for rendezvous hovering phases allowing for control saturation. *Journal of Guidance, Control, and Dynamics*, *42*(8), 1658–1675. http://dx.doi.org/10.2514/1.g003558.

Gong, Q., Fahroo, F., & Ross, I. M. (2008). Spectral algorithm for pseudospectral methods in optimal control. *Journal of Guidance, Control, and Dynamics*, *31*(3), 460–471. http://dx.doi.org/10.2514/1.32908.

Gong, Q., Ross, I. M., & Fahroo, F. (2010). Costate computation by a Chebyshev pseudospectral method. *Journal of Guidance, Control, and Dynamics*, *33*(2), 623–628. http://dx.doi.org/10.2514/1.45154.

Gong, Q., Ross, I. M., Kang, W., & Fahroo, F. (2007). Connections between the covector mapping theorem and convergence of pseudospectral methods for optimal control. *Computational Optimization and Applications*, *41*(3), 307–335. http://dx.doi.org/10.1007/s10589-007-9102-4.

Goodman, J. L. (2006). History of space shuttle rendezvous and proximity operations. *Journal of Spacecraft and Rockets*, *43*(5), 944–959. http://dx.doi.org/10.2514/1.19653.

Goodyear, A., Petersen, C., Pierre, J., Zagaris, C., Baldwin, M., & Kolmanovsky, I. (2015). Hardware implementation of model predictive control for relative motion maneuvering. In *2015 American control conference (ACC)*. IEEE, http://dx.doi.org/10.1109/acc.2015.7171077.

Graham, K. F., & Rao, A. V. (2015). Minimum-time trajectory optimization of multiple revolution low-thrust earth-orbit transfers. *Journal of Spacecraft and Rockets*, *52*(3), 711–727. http://dx.doi.org/10.2514/1.a33187.

Graham, K. F., & Rao, A. V. (2016). Minimum-time trajectory optimization of low-thrust earth-orbit transfers with eclipsing. *Journal of Spacecraft and Rockets*, *53*(2), 289–303. http://dx.doi.org/10.2514/1.a33416.

Grant, M., & Boyd, S. (2008). Graph implementations for nonsmooth convex programs. In V. Blondel, S. Boyd, & H. Kimura (Eds.), *Lecture Notes in Control and Information Sciences*, *Recent Advances in Learning and Control* (pp. 95–110). Springer-Verlag Limited, http://dx.doi.org/10.1007/978-1-84800-155-8_7.

Guo, Y., Hawkins, M., & Wie, B. (2013). Waypoint-optimized zero-effort-miss/zero-effort-velocity feedback guidance for mars landing. *Journal of Guidance, Control, and Dynamics*, *36*(3), 799–809. http://dx.doi.org/10.2514/1.58098.

Haberkorn, T., Martinon, P., & Gergaud, J. (2004). Low thrust minimum-fuel orbital transfer: A homotopic approach. *Journal of Guidance, Control, and Dynamics*, *27*(6), 1046–1060. http://dx.doi.org/10.2514/1.4022.

Haeussermann, W. (1970). Description and performance of the Saturn launch vehicle's navigation, guidance, and control system. *IFAC Proceedings Volumes*, *3*(1), 275–312. http://dx.doi.org/10.1016/s1474-6670(17)68785-8.

Halbe, O., Mathavaraj, S., & Padhi, R. (2010). Energy based suboptimal reentry guidance of a reusable launch vehicle using model predictive static programming. In *AIAA guidance, navigation, and control conference*. American Institute of Aeronautics and Astronautics, http://dx.doi.org/10.2514/6.2010-8311.

Halbe, O., Raja, R. G., & Padhi, R. (2014). Robust reentry guidance of a reusable launch vehicle using model predictive static programming. *Journal of Guidance, Control, and Dynamics*, *37*(1), 134–148. http://dx.doi.org/10.2514/1.61615.

Han, H., Qiao, D., Chen, H., & Li, X. (2019). Rapid planning for aerocapture trajectory via convex optimization. *Aerospace Science and Technology*, *84*, 763–775. http://dx.doi.org/10.1016/j.ast.2018.11.009.

Hanson, J., Shrader, M., & Cruzen, C. (1994). Ascent guidance comparisons. In *Guidance, navigation, and control conference*. American Institute of Aeronautics and Astronautics, http://dx.doi.org/10.2514/6.1994-3568.

Harris, M. W. (2021). Optimal control on disconnected sets using extreme point relaxations and normality approximations. *IEEE Transactions on Automatic Control*, 1–8. http://dx.doi.org/10.1109/tac.2021.3059682.

Harris, M. W., & Açıkmeşe, B. (2013a). Lossless convexification for a class of optimal control problems with linear state constraints. In *52nd IEEE conference on decision and control*. IEEE, http://dx.doi.org/10.1109/cdc.2013.6761017.

Harris, M. W., & Açıkmeşe, B. (2013b). Lossless convexification for a class of optimal control problems with quadratic state constraints. In *2013 American control conference*. IEEE, http://dx.doi.org/10.1109/acc.2013.6580359.

Harris, M. W., & Açıkmeşe, B. (2013c). Maximum divert for planetary landing using convex optimization. *Journal of Optimization Theory and Applications*, *162*(3), 975–995. http://dx.doi.org/10.1007/s10957-013-0501-7.

Harris, M. W., & Açıkmeşe, B. (2014). Lossless convexification of non-convex optimal control problems for state constrained linear systems. *Automatica*, *50*(9), 2304–2311. http://dx.doi.org/10.1016/j.automatica.2014.06.008.

Hartl, R. F., Sethi, S. P., & Vickson, R. G. (1995). A survey of the maximum principles for optimal control problems with state constraints. *SIAM Review*, *37*(2), 181–218. http://dx.doi.org/10.1137/1037043.

Hartley, E. N. (2015). A tutorial on model predictive control for spacecraft rendezvous. In *2015 European control conference (ECC)*. IEEE, http://dx.doi.org/10.1109/ecc.2015.7330727.

Hartley, E. N., Gallieri, M., & Maciejowski, J. M. (2013). Terminal spacecraft rendezvous and capture with LASSO model predictive control. *International Journal of Control*, *86*(11), 2104–2113. http://dx.doi.org/10.1080/00207179.2013.789608.

Hartley, E. N., & Maciejowski, J. M. (2014). Field programmable gate array based predictive control system for spacecraft rendezvous in elliptical orbits. *Optimal Control Applications & Methods*, *36*(5), 585–607. http://dx.doi.org/10.1002/oca.2117.

Hartley, E. N., Trodden, P. A., Richards, A. G., & Maciejowski, J. M. (2012). Model predictive control system design and implementation for spacecraft rendezvous. *Control Engineering Practice*, *20*(7), 695–713. http://dx.doi.org/10.1016/j.conengprac.2012.03.009.

Hicks, K. D. (2009). Introduction to astrodynamic reentry. (AFIT/EN/TR-09-03), Air Force Institute of Technology Wright-Patterson Air Force Base, OH.

Holt, H., Armellin, R., Baresi, N., Hashida, Y., Turconi, A., Scorsoglio, A., et al. (2021). Optimal Q-laws via reinforcement learning with guaranteed stability. *Acta Astronautica*, *187*, 511–528. http://dx.doi.org/10.1016/j.actaastro.2021.07.010.

Horn, H. J., Chandler, D. C., & Buckelew, V. L. (1969). Iterative guidance applied to generalized missions. *Journal of Spacecraft and Rockets*, *6*(1), 4–8. http://dx.doi.org/10.2514/3.29523.

Houska, B., Ferreau, H. J., & Diehl, M. (2010). ACADO Toolkit – An open-source framework for automatic control and dynamic optimization. *Optimal Control Applications and Methods*, *32*(3), 298–312. http://dx.doi.org/10.1002/oca.939.

Howell, T. A., Jackson, B. E., & Manchester, Z. (2019). ALTRO: A fast solver for constrained trajectory optimization. In *2019 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, http://dx.doi.org/10.1109/iros40897.2019.8967788.

Hu, Q., Xie, J., & Liu, X. (2018). Trajectory optimization for accompanying satellite obstacle avoidance. *Aerospace Science and Technology*, *82–83*, 220–233. http://dx.doi.org/10.1016/j.ast.2018.08.033.

Hu, H., Zhu, S., & Cui, P. (2016). Desensitized optimal trajectory for landing on small bodies with reduced landing error. *Aerospace Science and Technology*, *48*, 178–185. http://dx.doi.org/10.1016/j.ast.2015.11.006.

Hutao, C., Xiaojun, C., Rui, X., & Pingyuan, C. (2011). RHC-based attitude control of spacecraft under geometric constraints. *Aircraft Engineering and Aerospace Technology*, *83*(5), 296–305. http://dx.doi.org/10.1108/00022661111159906.

Izzo, D., Märtens, M., & Pan, B. (2019). A survey on artificial intelligence trends in spacecraft guidance dynamics and control. *Astrodynamics*, *3*(4), 287–299. http://dx.doi.org/10.1007/s42064-018-0053-6.

Jacobson, D. H. (1968). New second-order and first-order algorithms for determining optimal control: A differential dynamic programming approach. *Journal of Optimization Theory and Applications*, *2*(6), 411–440. http://dx.doi.org/10.1007/bf00925746.

Jennings, L., Yu, C., Li, B., Rehbock, V., Loxton, R., Teo, K. L., et al. (2015). VISUAL MISER: An efficient user-friendly visual program for solving optimal control problems. *Journal of Industrial and Management Optimization*, *12*(2), 781–810. http://dx.doi.org/10.3934/jimo.2016.12.781.

Jewison, C., Erwin, R. S., & Saenz-Otero, A. (2015). Model predictive control with ellipsoid obstacle constraints for spacecraft rendezvous. *IFAC-PapersOnLine*, *48*(9), 257–262. http://dx.doi.org/10.1016/j.ifacol.2015.08.093.

Jewison, C., & Miller, D. W. (2018). Probabilistic trajectory optimization under uncertain path constraints for close proximity operations. *Journal of Guidance, Control, and Dynamics*, *41*(9), 1843–1858. http://dx.doi.org/10.2514/1.g003152.

Jiang, X., Li, S., & Tao, T. (2018). Computational guidance for planetary powered descent using collaborative optimization. *Aerospace Science and Technology*, *76*, 37–48. http://dx.doi.org/10.1016/j.ast.2018.02.009.

Jin, Z., Chen, J., Sheng, Y., & Liu, X. (2016). Neural network based adaptive fuzzy PID-type sliding mode attitude control for a reentry vehicle. *International Journal of Control, Automation and Systems*, *15*(1), 404–415. http://dx.doi.org/10.1007/s12555-015-0181-1.

Jin, K., Geller, D. K., & Luo, J. (2020). Robust trajectory design for rendezvous and proximity operations with uncertainties. *Journal of Guidance, Control, and Dynamics*, *43*(4), 741–753. http://dx.doi.org/10.2514/1.g004121.

Jo, J.-W., & Prussing, J. E. (2000). Procedure for applying second-order conditions in optimal control problems. *Journal of Guidance, Control, and Dynamics*, *23*(2), 241–250. http://dx.doi.org/10.2514/2.4546.

Johnson, B. J., Braden, E. M., Sostaric, R. R., Cerimele, C. J., & Lu, P. (2018). Entry, descent, and landing performance for a mid-lift-to-drag ratio vehicle at mars. In *2018 American astronautical society guidance and control conference, Breckenridge, CO*.

Johnson, W. R., Lu, P., & Stachowiak, S. (2017). Automated reentry system using FNPEG. In *AIAA guidance, navigation, and control conference*. American Institute of Aeronautics and Astronautics, http://dx.doi.org/10.2514/6.2017-1899.

Johnson, B. J., Rocca-Bejar, D., Lu, P., Nikaido, B., Hays, Z. B., D'Souza, S., et al. (2020). Pterodactyl: Development and performance of guidance algorithms for a mechanically deployed entry vehicle. In *AIAA SciTech 2020 forum*. American Institute of Aeronautics and Astronautics, http://dx.doi.org/10.2514/6.2020-1011.

Jones, J. W. (2018). The recent large reduction in space launch cost. In *48th international conference on environmental systems*, Albuquerque, NM.

Josselyn, S., & Ross, I. M. (2003). Rapid verification method for the trajectory optimization of reentry vehicles. *Journal of Guidance, Control, and Dynamics*, *26*(3), 505–508. http://dx.doi.org/10.2514/2.5074.

JPL, Masten Space Systems (2012a). 500 meter divert xombie test flight for G-FOLD, guidance for fuel optimal large divert, validation. http://www.youtube.com/watch?v=1GRwimo1AwY.

JPL, Masten Space Systems (2012b). 750 meter divert xombie test flight for G-FOLD, guidance for fuel optimal large divert, validation. http://www.youtube.com/watch?v=jl6pw2oossU.

Kamath, A. G., Assadian, F. F., & Robinson, S. K. (2020). Multivariable robust control for the powered-descent of a multibody lunar landing system. In *AIAA/AAS Astrodynamics Specialist Conference*. American Institute of Aeronautics and Astronautics.

Kawaguchi, J., Fujiwara, A., & Uesugi, T. (2008). Hayabusa—its technology and science accomplishment summary and Hayabusa-2. *Acta Astronautica*, *62*(10–11), 639–647. http://dx.doi.org/10.1016/j.actaastro.2008.01.028.

Kechichian, J. A. (1995). Optimal low-thrust transfer using variable bounded thrust. *Acta Astronautica*, *36*(7), 357–365. http://dx.doi.org/10.1016/0094-5765(95)00112-3.

Kelly, M. (2017). An introduction to trajectory optimization: How to do your own direct collocation. *SIAM Review*, *59*(4), 849–904. http://dx.doi.org/10.1137/16m1062569.

Kim, Y., & Mesbahi, M. (2004). Quadratically constrained attitude control via semidefinite programming. *IEEE Transactions on Automatic Control*, *49*(5), 731–735. http://dx.doi.org/10.1109/tac.2004.825959.

Kim, Y., Mesbahi, M., Singh, G., & Hadaegh, F. Y. (2010). On the convex parameterization of constrained spacecraft reorientation. *IEEE Transactions on Aerospace and Electronic Systems*, *46*(3), 1097–1109. http://dx.doi.org/10.1109/taes.2010.5545176.

Kirk, D. E. (1970). *Optimal control theory: An introduction*. Englewood Cliffs, NJ: Prentice-Hall.

Klumpp, A. R. (1974). Apollo Lunar descent guidance. *Automatica*, *10*(2), 133–146. http://dx.doi.org/10.1016/0005-1098(74)90019-3.

Koeppen, N., Ross, I. M., Wilcox, L. C., & Proulx, R. J. (2019). Fast mesh refinement in pseudospectral optimal control. *Journal of Guidance, Control, and Dynamics*, *42*(4), 711–722. http://dx.doi.org/10.2514/1.g003904.

Kunhippurayil, S., Harris, M. W., & Jansson, O. (2021). Lossless convexification of optimal control problems with annular control constraints. *Automatica*.

LaFarge, N. B., Miller, D., Howell, K. C., & Linares, R. (2020). Guidance for closed-loop transfers using reinforcement learning with application to libration point orbits. In *AIAA SciTech 2020 forum*. American Institute of Aeronautics and Astronautics, http://dx.doi.org/10.2514/6.2020-0458.

Lantoine, G., & Russell, R. P. (2012a). A hybrid differential dynamic programming algorithm for constrained optimal control problems. Part 1: Theory. *Journal of Optimization Theory and Applications*, *154*(2), 382–417. http://dx.doi.org/10.1007/s10957-012-0039-0.

Lantoine, G., & Russell, R. P. (2012b). A hybrid differential dynamic programming algorithm for constrained optimal control problems. Part 2: Application. *Journal of Optimization Theory and Applications*, *154*(2), 418–442. http://dx.doi.org/10.1007/s10957-012-0038-1.

Larsson, R., Berge, S., Bodin, P., & Jönsson, U. T. (2006). Fuel efficient relative orbit control strategies for formation flying and rendezvous within PRISMA. In *29th annual AAS guidance and control conference* (pp. 25–40).

Lauretta, D. S., Balram-Knutson, S. S., Beshore, E., Boynton, W. V., d'Aubigny, C. D., DellaGiustina, D. N., et al. (2017). OSIRIS-REx: Sample return from asteroid (101955) bennu. *Space Science Reviews*, *212*(1–2), 925–984. http://dx.doi.org/10.1007/s11214-017-0405-1.

Lawden, D. (1963). *Optimal trajectories for space navigation*. London: Butterworths.

Lee, D. Y., Gupta, R., Kalabić, U. V., Di Cairano, S., Bloch, A. M., Cutler, J. W., et al. (2017). Geometric mechanics based nonlinear model predictive spacecraft attitude control with reaction wheels. *Journal of Guidance, Control, and Dynamics*, *40*(2), 309–319. http://dx.doi.org/10.2514/1.g001923.

Lee, U., & Mesbahi, M. (2013). Quaternion-based optimal spacecraft reorientation under complex attitude constrained zones. In *AAS/AIAA astrodynamics specialist conference. AAS/AIAA*.

Lee, U., & Mesbahi, M. (2014). Dual quaternion based spacecraft rendezvous with rotational and translational field of view constraints. In *AIAA/AAS astrodynamics specialist conference*. American Institute of Aeronautics and Astronautics, http://dx.doi.org/10.2514/6.2014-4362.

Lee, U., & Mesbahi, M. (2017). Constrained autonomous precision landing via dual quaternions and model predictive control. *Journal of Guidance, Control, and Dynamics*, *40*(2), 292–308. http://dx.doi.org/10.2514/1.g001879.

Li, Y., Guan, Y., Wei, C., & Hu, R. (2019). Optimal control of ascent trajectory for launch vehicles: A convex approach. *IEEE Access*, *7*, 186491–186498. http://dx.doi.org/10.1109/access.2019.2960864.

Li, Y., Pang, B., Wei, C., Cui, N., & Liu, Y. (2020). Online trajectory optimization for power system fault of launch vehicles via convex programming. *Aerospace Science and Technology*, *98*, Article 105682. http://dx.doi.org/10.1016/j.ast.2020.105682.

Li, Q., Yuan, J., Zhang, B., & Gao, C. (2017). Model predictive control for autonomous rendezvous and docking with a tumbling target. *Aerospace Science and Technology*, *69*, 700–711. http://dx.doi.org/10.1016/j.ast.2017.07.022.

Li, Q., Zhang, B., Yuan, J., & Wang, H. (2018). Potential function based robust safety control for spacecraft rendezvous and proximity operations under path constraint. *Advances in Space Research*, *62*(9), 2586–2598. http://dx.doi.org/10.1016/j.asr.2018.08.003.

Li, P., & Zhu, Z. H. (2017). Line-of-sight nonlinear model predictive control for autonomous rendezvous in elliptical orbit. *Aerospace Science and Technology*, *69*, 236–243. http://dx.doi.org/10.1016/j.ast.2017.06.030.

Li, P., & Zhu, Z. H. (2018a). Model predictive control for spacecraft rendezvous in elliptical orbit. *Acta Astronautica*, *146*, 339–348. http://dx.doi.org/10.1016/j.actaastro.2018.03.025.

Li, P., & Zhu, Z. H. (2018b). Pulse-width pulse-frequency modulated nonlinear model predictive control for spacecraft rendezvous. In *2018 space flight mechanics meeting*. American Institute of Aeronautics and Astronautics, http://dx.doi.org/10.2514/6.2018-2218.

Li, P., Zhu, Z. H., & Meguid, S. (2016). State dependent model predictive control for orbital rendezvous using pulse-width pulse-frequency modulated thrusters. *Advances in Space Research*, *58*(1), 64–73. http://dx.doi.org/10.1016/j.asr.2016.04.022.

Liao-McPherson, D., Dunham, W. D., & Kolmanovsky, I. (2016). Model predictive control strategies for constrained soft landing on an asteroid. In *AIAA/AAS astrodynamics specialist conference*. American Institute of Aeronautics and Astronautics, http://dx.doi.org/10.2514/6.2016-5507.

Linares, R., & Raquepas, J. B. (2018). Physically-constrained inverse optimal control for satellite maneuver detection. In *2018 AAS/AIAA astrodynamics specialist conference*.

Liu, X. (2019). Fuel-optimal rocket landing with aerodynamic controls. *Journal of Guidance, Control, and Dynamics*, *42*(1), 65–77. http://dx.doi.org/10.2514/1.g003537.

Liu, J., & Li, H. (2019). Artificial potential function safety and obstacle avoidance guidance for autonomous rendezvous and docking with noncooperative target. *Mathematical Problems in Engineering*, *2019*, 1–17. http://dx.doi.org/10.1155/2019/3451864.

Liu, X., & Lu, P. (2014). Solving nonconvex optimal control problems by convex optimization. *Journal of Guidance, Control, and Dynamics*, *37*(3), 750–765. http://dx.doi.org/10.2514/1.62110.

Liu, X., Lu, P., & Pan, B. (2017). Survey of convex optimization for aerospace applications. *Astrodynamics*, *1*(1), 23–40. http://dx.doi.org/10.1007/s42064-017-0003-8.

Liu, X., & Shen, Z. (2015). Rapid smooth entry trajectory planning for high lift/drag hypersonic glide vehicles. *Journal of Optimization Theory and Applications*, *168*(3), 917–943. http://dx.doi.org/10.1007/s10957-015-0831-8.

Liu, X., Shen, Z., & Lu, P. (2015). Solving the maximum-crossrange problem via successive second-order cone programming with a line search. *Aerospace Science and Technology*, *47*, 10–20. http://dx.doi.org/10.1016/j.ast.2015.09.008.

Liu, X., Shen, Z., & Lu, P. (2016a). Entry trajectory optimization by second-order cone programming. *Journal of Guidance, Control, and Dynamics*, *39*(2), 227–241. http://dx.doi.org/10.2514/1.g001210.

Liu, X., Shen, Z., & Lu, P. (2016b). Exact convex relaxation for optimal flight of aerodynamically controlled missiles. *IEEE Transactions on Aerospace and Electronic Systems*, *52*(4), 1881–1892. http://dx.doi.org/10.1109/taes.2016.150741.

Löfberg, J. (2004). YALMIP : A toolbox for modeling and optimization in MATLAB. In *2004 IEEE International Conference on Robotics and Automation*. IEEE, http://dx.doi.org/10.1109/cacsd.2004.1393890.

Long, K. (2004). *James Webb space telescope project mission operations concept document: Technical Report JWST-OPS-002018*, Greenbelt, MD: Goddard Space Flight Center.

Longuski, J. M., Guzmán, J. J., & Prussing, J. E. (2014). *Optimal control with aerospace applications*. New York: Springer, http://dx.doi.org/10.1007/978-1-4614-8945-0.

Louembet, C., Arzelier, D., & Deaconu, G. (2015). Robust rendezvous planning under maneuver execution errors. *Journal of Guidance, Control, and Dynamics*, *38*(1), 76–93. http://dx.doi.org/10.2514/1.g000391.

Lu, P. (2008). Predictor-corrector entry guidance for low-lifting vehicles. *Journal of Guidance, Control, and Dynamics*, *31*(4), 1067–1075. http://dx.doi.org/10.2514/1.32055.

Lu, P. (2014). Entry guidance: A unified method. *Journal of Guidance, Control, and Dynamics*, *37*(3), 713–728. http://dx.doi.org/10.2514/1.62605.

Lu, P. (2017). Introducing computational guidance and control. In P. Lu (Ed.), *Journal of Guidance, Control, and Dynamics*, *40*(2), 193. http://dx.doi.org/10.2514/1.g002745.

Lu, P. (2018). Propellant-optimal powered descent guidance. *Journal of Guidance, Control, and Dynamics*, *41*(4), 813–826. http://dx.doi.org/10.2514/1.g003243.

Lu, P. (2019). Augmented Apollo powered descent guidance. *Journal of Guidance, Control, and Dynamics*, *42*(3), 447–457. http://dx.doi.org/10.2514/1.g004048.

Lu, P. (2020). Theory of fractional-polynomial powered descent guidance. *Journal of Guidance, Control, and Dynamics*, *43*(3), 398–409. http://dx.doi.org/10.2514/1.g004556.

Lu, P., Cerimele, C. J., Tigges, M. A., & Matz, D. A. (2015). Optimal aerocapture guidance. *Journal of Guidance, Control, and Dynamics*, *38*(4), 553–565. http://dx.doi.org/10.2514/1.g000713.

Lu, P., & Liu, X. (2013). Autonomous trajectory planning for rendezvous and proximity operations by conic optimization. *Journal of Guidance, Control, and Dynamics*, *36*(2), 375–389. http://dx.doi.org/10.2514/1.58436.

Lu, P., & Pan, B. (2010). Highly constrained optimal launch ascent guidance. *Journal of Guidance, Control, and Dynamics*, *33*(2), 404–414. http://dx.doi.org/10.2514/1.45632.

Lu, P., Sun, H., & Tsai, B. (2003). Closed-loop endoatmospheric ascent guidance. *Journal of Guidance, Control, and Dynamics*, *26*(2), 283–294. http://dx.doi.org/10.2514/2.5045.

Lu, P., & Xue, S. (2010). Rapid generation of accurate entry landing footprints. *Journal of Guidance, Control, and Dynamics*, *33*(3), 756–767. http://dx.doi.org/10.2514/1.46833.

Lu, P., Zhang, L., & Sun, H. (2005). Ascent guidance for responsive launch: A fixed-point approach. In *AIAA guidance, navigation, and control conference and exhibit*. American Institute of Aeronautics and Astronautics, http://dx.doi.org/10.2514/6.2005-6453.

Luo, Y.-Z., Lei, Y.-J., & Tang, G.-J. (2007). Optimal multi-objective nonlinear impulsive rendezvous. *Journal of Guidance, Control, and Dynamics*, *30*(4), 994–1002. http://dx.doi.org/10.2514/1.27910.

Luo, Y., Serrani, A., Yurkovich, S., Oppenheimer, M. W., & Doman, D. B. (2007). Model-predictive dynamic control allocation scheme for reentry vehicles. *Journal of Guidance, Control, and Dynamics*, *30*(1), 100–113. http://dx.doi.org/10.2514/1.25473.

Luo, Y.-Z., Tang, G.-J., & Lei, Y.-J. (2007). Optimal multi-objective linearized impulsive rendezvous. *Journal of Guidance, Control, and Dynamics*, *30*(2), 383–389. http://dx.doi.org/10.2514/1.21433.

Luo, Y., Tang, G., & Parks, G. (2008). Multi-objective optimization of perturbed impulsive rendezvous trajectories using physical programming. *Journal of Guidance, Control, and Dynamics*, *31*(6), 1829–1832. http://dx.doi.org/10.2514/1.35409.

Luo, Y., Zhang, J., & Tang, G. (2014). Survey of orbital dynamics and control of space rendezvous. *Chinese Journal of Aeronautics*, *27*(1), 1–11. http://dx.doi.org/10.1016/j.cja.2013.07.042.

Lurie, B. J., & Enright, P. J. (2000). *Classical feedback control with MATLAB*. New York: Marcel Dekker.

Majumdar, A., Hall, G., & Ahmadi, A. A. (2020). Recent scalability improvements for semidefinite programming with applications in machine learning, control, and robotics. *Annual Review of Control, Robotics, and Autonomous Systems*, *3*(1), 331–360. http://dx.doi.org/10.1146/annurev-control-091819-074326.

Majumdar, A., & Tedrake, R. (2017). Funnel libraries for real-time robust feedback motion planning. *International Journal of Robotics Research*, *36*(8), 947–982. http://dx.doi.org/10.1177/0278364917712421.

Mall, K., & Grant, M. J. (2017). Epsilon-trig regularization method for bang-bang optimal control problems. *Journal of Optimization Theory and Applications*, *174*(2), 500–517. http://dx.doi.org/10.1007/s10957-017-1129-9.

Mall, K., Grant, M. J., & Taheri, E. (2020). Uniform trigonometrization method for optimal control problems with control and state constraints. *Journal of Spacecraft and Rockets*, *57*(5), 995–1007. http://dx.doi.org/10.2514/1.a34624.

Malladi, B. P., Di Cairano, S., & Weiss, A. (2019). Nonlinear model predictive control of coupled rotational-translational spacecraft relative motion. In *2019 American control conference (ACC)*. IEEE, http://dx.doi.org/10.23919/acc.2019.8814345.

Malyuta, D., & Açıkmeşe, B. (2020a). Approximate multiparametric mixed-integer convex programming. *IEEE Control Systems Letters*, *4*(1), http://dx.doi.org/10.1109/lcsys.2019.2922639, arXiv:1902.10994.

Malyuta, D., & Açıkmeşe, B. (2020b). Lossless convexification of optimal control problems with semi-continuous inputs. *IFAC-PapersOnLine*, *53*(2), 6843–6850. http://dx.doi.org/10.1016/j.ifacol.2020.12.341.

Malyuta, D., & Açıkmeşe, B. (2021). Fast homotopy for spacecraft rendezvous trajectory optimization with discrete logic. *Journal of Guidance, Control, and Dynamics (in review)*, arXiv:2107.07001.

Malyuta, D., Reynolds, T. P., Szmuk, M., Açıkmeşe, B., & Mesbahi, M. (2020). Fast trajectory optimization via successive convexification for spacecraft rendezvous with integer constraints. In *AIAA SciTech 2020 forum*. American Institute of Aeronautics and Astronautics, http://dx.doi.org/10.2514/6.2020-0616.

Malyuta, D., Reynolds, T. P., Szmuk, M., Lew, T., Bonalli, R., Pavone, M., et al. (2021). Convex optimization for trajectory generation. *IEEE Control Systems Magazine (in review)*, arXiv:2106.09125.

Malyuta, D., Reynolds, T. P., Szmuk, M., Mesbahi, M., Açıkmeşe, B., & Carson III, J. M. (2019). Discretization performance and accuracy analysis for the rocket powered descent guidance problem. In *AIAA SciTech 2019 forum, January* (pp. 1–20). http://dx.doi.org/10.2514/6.2019-0925.

Mammarella, M., Capello, E., Park, H., Guglieri, G., & Romano, M. (2018). Tube-based robust model predictive control for spacecraft proximity operations in the presence of persistent disturbance. *Aerospace Science and Technology*, *77*, 585–594. http://dx.doi.org/10.1016/j.ast.2018.04.009.

Mammarella, M., Lorenzen, M., Capello, E., Park, H., Dabbene, F., Guglieri, G., et al. (2020). An offline-sampling SMPC framework with application to autonomous space maneuvers. *IEEE Transactions on Control Systems Technology*, *28*(2), 388–402. http://dx.doi.org/10.1109/tcst.2018.2879938.

Mao, Y., Dueri, D., Szmuk, M., & Açıkmeşe, B. (2018). Convexification and real-time optimization for MPC with aerospace applications. In *Handbook of model predictive control* (pp. 335–358). Springer International Publishing, http://dx.doi.org/10.1007/978-3-319-77489-3_15.

Mao, Y., Szmuk, M., Xu, X., & Açıkmeşe, B. (2018). Successive convexification: A superlinearly convergent algorithm for non-convex optimal control problems. arXiv:1804.06539.

Marcucci, T., & Tedrake, R. (2019). Mixed-integer formulations for optimal control of piecewise-affine systems. In *Proceedings of the 22nd ACM international conference on hybrid systems computation and control - HSCC '19* (pp. 230–239). ACM Press, http://dx.doi.org/10.1145/3302504.3311801.

Marec, J.-P. (1979). *Optimal space trajectories*. Amsterdam: Elsevier Scientific Publishing Company, http://dx.doi.org/10.1002/oca.4660040210.

Mattingley, J., & Boyd, S. (2011). CVXGEN: A code generator for embedded convex optimization. *Optimization and Engineering*, *13*(1), 1–27. http://dx.doi.org/10.1007/s11081-011-9176-9.

Mayne, D. (1966). A second-order gradient method for determining optimal trajectories of non-linear discrete-time systems. *International Journal of Control*, *3*(1), 85–95. http://dx.doi.org/10.1080/00207176608921369.

Mayne, D. Q. (2014). Model predictive control: Recent developments and future promise. *Automatica*, *50*(12), 2967–2986. http://dx.doi.org/10.1016/j.automatica.2014.10.128.

Mayne, D. (2015). Robust and stochastic MPC: Are we going in the right direction? *IFAC-PapersOnLine*, *48*(23), 1–8. http://dx.doi.org/10.1016/j.ifacol.2015.11.255.

Mayne, D., Rawlings, J., Rao, C., & Scokaert, P. (2000). Constrained model predictive control: Stability and optimality. *Automatica*, *36*(6), 789–814. http://dx.doi.org/10.1016/s0005-1098(99)00214-9.

McCue, G. A. (1967). Quasilinearization determination of optimum finite-thrust orbital transfers. *AIAA Journal*, *5*(4), 755–763. http://dx.doi.org/10.2514/3.4058.

McDonald, S., Grizzel, T., & Wang, Z. (2020). A real-time approach to minimum-energy reorientation of an asymmetric rigid body spacecraft. In *AIAA SciTech 2020 forum*. American Institute of Aeronautics and Astronautics, http://dx.doi.org/10.2514/6.2020-1203.

McHenry, R., Long, A., Cockrell, B., Thibodeau III, J., & Brand, T. (1979). Space shuttle ascent guidance, navigation, and control. *Journal of the Astronautical Sciences*, *27*, 1–38.

Meditch, J. (1964). On the problem of optimal thrust programming for a Lunar soft landing. *IEEE Transactions on Automatic Control*, *9*(4), 477–484. http://dx.doi.org/10.1109/tac.1964.1105758.

Mendeck, G., & Craig, L. (2011). Entry guidance for the 2011 Mars Science Laboratory mission. In *AIAA atmospheric flight mechanics conference*. American Institute of Aeronautics and Astronautics, http://dx.doi.org/10.2514/6.2011-6639.

Miller, D., Englander, J. A., & Linares, R. (2019). Interplanetary low-thrust design using proximal policy optimization. In *AAS/AIAA astrodynamics specialist conference*. American Astronautical Society.

Miller, D., & Linares, R. (2019). Low-thrust optimal control via reinforcement learning. In *29th AAS/AIAA space flight mechanics meeting*.

Mindell, D. A. (2008). *Digital apollo*. MIT Press.

Morgan, D., Chung, S.-J., Blackmore, L., Açıkmeşe, B., Bayard, D., & Hadaegh, F. Y. (2012). Swarm-keeping strategies for spacecraft under J2 and atmospheric drag perturbations. *Journal of Guidance, Control, and Dynamics*, *35*(5), 1492–1506. http://dx.doi.org/10.2514/1.55705.

Murillo, O., & Lu, P. (2010). Fast ascent trajectory optimization for hypersonic air-breathing vehicles. In *AIAA guidance, navigation, and control conference*. American Institute of Aeronautics and Astronautics, http://dx.doi.org/10.2514/6.2010-8173.

NASA Science (2019). Moon's south pole in NASA's landing sites. https://solarsystem.nasa.gov/news/907/moons-south-pole-in-nasas-landing-sites/. Accessed: 11/26/2019.

Nemhauser, G. L., & Wolsey, L. A. (1999). *Integer and combinatorial optimization*. Hoboken, NJ: Wiley.

Nocedal, J., & Wright, S. (1999). *Numerical optimization*. Springer.

Ono, M., Pavone, M., Kuwata, Y., & Balaram, J. (2015). Chance-constrained dynamic programming with application to risk-aware robotic space exploration. *Autonomous Robots*, *39*(4), 555–571. http://dx.doi.org/10.1007/s10514-015-9467-7.

Palacios-Gomez, F., Lasdon, L., & Engquist, M. (1982). Nonlinear optimization by successive linear programming. *Management Science*, *28*(10), 1106–1120. http://dx.doi.org/10.1287/mnsc.28.10.1106.

Palumbo, N. F., Blauwkamp, R. A., & Lloyd, J. M. (2010). Modern homing missile guidance theory and techniques. *Johns Hopkins APL Technical Digest*, *29*(1), 42–59.

Pan, B., Chen, Z., Lu, P., & Gao, B. (2013). Reduced transversality conditions in optimal space trajectories. *Journal of Guidance, Control, and Dynamics*, *36*(5), 1289–1300. http://dx.doi.org/10.2514/1.60181.

Pan, B., & Lu, P. (2010). Improvements to optimal launch ascent guidance. In *AIAA guidance, navigation, and control conference*. American Institute of Aeronautics and Astronautics, http://dx.doi.org/10.2514/6.2010-8174.

Pan, B., Lu, P., & Chen, Z. (2012). Coast arcs in optimal multiburn orbital transfers. *Journal of Guidance, Control, and Dynamics*, *35*(2), 451–461. http://dx.doi.org/10.2514/1.54655.

Pan, B., Lu, P., Pan, X., & Ma, Y. (2016). Double-homotopy method for solving optimal control problems. *Journal of Guidance, Control, and Dynamics*, *39*(8), 1706–1720. http://dx.doi.org/10.2514/1.g001553.

Pan, X., & Pan, B. (2020). Practical homotopy methods for finding the best minimum-fuel transfer in the circular restricted three-body problem. *IEEE Access*, *8*, 47845–47862. http://dx.doi.org/10.1109/access.2020.2978246.

Pan, B., Pan, X., & Lu, P. (2019). Finding best solution in low-thrust trajectory optimization by two-phase homotopy. *Journal of Spacecraft and Rockets*, *56*(1), 283–291. http://dx.doi.org/10.2514/1.a34144.

Pan, B., Pan, X., & Zhang, S. (2018). A new probability-one homotopy method for solving minimum-time low-thrust orbital transfer problems. *Astrophysics and Space Science*, *363*(9), http://dx.doi.org/10.1007/s10509-018-3420-0.

Park, H., Di Cairano, S., & Kolmanovsky, I. (2011). Model predictive control for spacecraft rendezvous and docking with a rotating/tumbling platform and for debris avoidance. In *Proceedings of the 2011 American control conference*. IEEE, http://dx.doi.org/10.1109/acc.2011.5991151.

Park, H., Kolmanovsky, I., & Sun, J. (2013). Model predictive control of spacecraft relative motion maneuvers using the IPA-SQP approach. In *ASME 2013 dynamic systems and control conference*. American Society of Mechanical Engineers, http://dx.doi.org/10.1115/dscc2013-3854.

Park, H., Zagaris, C., Llop, J. V., Zappulla, R., Kolmanovsky, I., & Romano, M. (2016). Analysis and experimentation of model predictive control for spacecraft rendezvous and proximity operations with multiple obstacle avoidance. In *AIAA/AAS astrodynamics specialist conference*. American Institute of Aeronautics and Astronautics, http://dx.doi.org/10.2514/6.2016-5273.

Pascucci, C. A., Bennani, S., & Bemporad, A. (2015). Model predictive control for powered descent guidance and control. In *2015 European control conference (ECC)*. IEEE, http://dx.doi.org/10.1109/ecc.2015.7330732.

Patterson, M. A., & Rao, A. V. (2014). GPOPS-II: A MATLAB software for solving multiple-phase optimal control problems using hp-adaptive Gaussian quadrature collocation methods and sparse nonlinear programming. *ACM Transactions on Mathematical Software*, *41*(1), 1–37. http://dx.doi.org/10.1145/2558904.

Pavlov, A., Shames, I., & Manzie, C. (2020). Interior point differential dynamic programming. arXiv:2004.12710.

Pellegrini, E., & Russell, R. P. (2020a). A multiple-shooting differential dynamic programming algorithm. Part 1: Theory. *Acta Astronautica*, *170*, 686–700. http://dx.doi.org/10.1016/j.actaastro.2019.12.037.

Pellegrini, E., & Russell, R. P. (2020b). A multiple-shooting differential dynamic programming algorithm. Part 2: Applications. *Acta Astronautica*, *173*, 460–472. http://dx.doi.org/10.1016/j.actaastro.2019.12.038.

Peng, J., Roos, C., & Terlaky, T. (2002). Primal-dual interior-point methods for second-order conic optimization based on self-regular proximities. *SIAM Journal on Optimization*, *13*(1), 179–203. http://dx.doi.org/10.1137/s1052623401383236.

Petersen, C., Jaunzemis, A., Baldwin, M., Holzinger, M., & Kolmanovsky, I. (2014). Model predictive control and extended command governor for improving robustness of relative motion guidance and control. In *Proc. AAS/AIAA space flight mechanics meeting*. American Institute of Aeronautics and Astronautics (AIAA).

Petropoulos, A., & Russell, R. (2008). Low-thrust transfers using primer vector theory and a second-order penalty method. In *AIAA/AAS astrodynamics specialist conference and exhibit*. American Institute of Aeronautics and Astronautics, http://dx.doi.org/10.2514/6.2008-6955.

Phogat, K. S., Chatterjee, D., & Banavar, R. N. (2018a). A discrete-time Pontryagin maximum principle on matrix lie groups. *Automatica*, *97*, 376–391. http://dx.doi.org/10.1016/j.automatica.2018.08.026.

Phogat, K. S., Chatterjee, D., & Banavar, R. (2018b). Discrete-time optimal attitude control of a spacecraft with momentum and control constraints. *Journal of Guidance, Control, and Dynamics*, *41*(1), 199–211. http://dx.doi.org/10.2514/1.g002861.

Picot, G. (2012). Shooting and numerical continuation methods for computing time-minimal and energy-minimal trajectories in the Earth-Moon system using low propulsion. *Discrete & Continuous Dynamical Systems - B*, *17*(1), 245–269. http://dx.doi.org/10.3934/dcdsb.2012.17.245.

Pinson, R., & Lu, P. (2015). Rapid generation of optimal asteroid powered descent trajectories via convex optimization. In *Proceedings of the AAS/AIAA astrodynamics specialist conference AAS 15-616*.

Pinson, R. M., & Lu, P. (2018). Trajectory design employing convex optimization for landing on irregularly shaped asteroids. *Journal of Guidance, Control, and Dynamics*, *41*(6), 1243–1256. http://dx.doi.org/10.2514/1.g003045.

Ploen, S., Açıkmeşe, B., & Wolf, A. (2006). A comparison of powered descent guidance laws for Mars pinpoint landing. In *AIAA/AAS astrodynamics specialist conference and exhibit*. American Institute of Aeronautics and Astronautics, http://dx.doi.org/10.2514/6.2006-6676.

Pontani, M., & Conway, B. (2013). Optimal low-thrust orbital maneuvers via indirect swarming method. *Journal of Optimization Theory and Applications*, *162*(1), 272–292. http://dx.doi.org/10.1007/s10957-013-0471-9.

Pontryagin, L. S., Boltyanskii, V. G., Gamkrelidze, R. V., & Mishchenko, E. F. (1986). *The mathematical theory of optimal processes*. Montreux: Gordon and Breach Science Publishers, http://dx.doi.org/10.1201/9780203749319.

Qin, S., & Badgwell, T. A. (2003). A survey of industrial model predictive control technology. *Control Engineering Practice*, *11*(7), 733–764. http://dx.doi.org/10.1016/s0967-0661(02)00186-7.

Rahmani, A., Bandyopadhyay, S., Rossi, F., de la Croix, J.-P., Hook, J. V., & Wolf, M. (2019). Space vehicle swarm exploration missions: A study of key enabling technologies and gaps. In *70th international astronautical congress*. International Astronautical Federation.

Rao, A. V. (2010). A survey of numerical methods for optimal control. *Advances in the Astronautical Sciences*, *135*, 497–528.

Rasotto, M., Armellin, R., & Lizia, P. D. (2015). Multi-step optimization strategy for fuel-optimal orbital transfer of low-thrust spacecraft. *Engineering Optimization*, *48*(3), 519–542. http://dx.doi.org/10.1080/0305215x.2015.1025773.

Rawlings, J. B., Mayne, D. Q., & Diehl, M. M. (2017). *Model predictive control: Theory, computation and design* (2nd ed.). Madison, WI: Nob Hill Publishing.

Rea, J. (2003). Launch vehicle trajectory optimization using a Legendre pseudospectral method. In *AIAA guidance, navigation, and control conference and exhibit*. American Institute of Aeronautics and Astronautics, http://dx.doi.org/10.2514/6.2003-5640.

Rea, J., & Putnam, Z. (2007). A comparison of two Orion skip entry guidance algorithms. In *AIAA guidance, navigation and control conference and exhibit*. American Institute of Aeronautics and Astronautics, http://dx.doi.org/10.2514/6.2007-6424.

Recasens, J., Chu, Q., & Mulder, J. (2005). Robust model predictive control of a feedback linearized system for a lifting-body reentry vehicle. In *AIAA guidance, navigation, and control conference and exhibit*. American Institute of Aeronautics and Astronautics, http://dx.doi.org/10.2514/6.2005-6147.

Restrepo, R. L., & Russell, R. P. (2017). Shadow trajectory model for fast low-thrust indirect optimization. *Journal of Spacecraft and Rockets*, *54*(1), 44–54. http://dx.doi.org/10.2514/1.a33611.

Reynolds, T. P. (2020). *Computational guidance and control for aerospace systems*. (Ph.D. thesis), Seattle: University of Washington.

Reynolds, T. P., Malyuta, D., Mesbahi, M., & Açıkmeşe, B. (2020). Temporally-interpolated funnel synthesis for nonlinear systems. In *2nd RSS workshop on robust autonomy*. RSS, URL: https://openreview.net/forum?id=cMoxvmU22Zg.

Reynolds, T. P., Malyuta, D., Mesbahi, M., Açıkmeşe, B., & Carson III, J. M. (2020). A real-time algorithm for non-convex powered descent guidance. In *AIAA SciTech 2020 forum*. American Institute of Aeronautics and Astronautics, http://dx.doi.org/10.2514/6.2020-0844.

Reynolds, T. P., & Mesbahi, M. (2017). Small body precision landing via convex model predictive control. In *AIAA SPACE and astronautics forum and exposition*. American Institute of Aeronautics and Astronautics, http://dx.doi.org/10.2514/6.2017-5179.

Reynolds, T. P., & Mesbahi, M. (2020a). Crawling phenomena in sequential convexification for solving nonconvex problems. In *2020 American control conference (ACC)*. IEEE.

Reynolds, T. P., & Mesbahi, M. (2020b). Optimal planar powered descent with independent thrust and torque. *Journal of Guidance, Control, and Dynamics*, *43*(7), 1225–1231. http://dx.doi.org/10.2514/1.g004701.

Reynolds, T. P., Szmuk, M., Malyuta, D., Mesbahi, M., Açıkmeşe, B., & Carson III, J. M. (2019a). A state-triggered line of sight constraint for 6-DoF powered descent guidance problems. In *AIAA SciTech 2019 forum*. American Institute of Aeronautics and Astronautics, http://dx.doi.org/10.2514/6.2019-0924.

Reynolds, T. P., Szmuk, M., Malyuta, D., Mesbahi, M., Açıkmeşe, B., & Carson III, J. M. (2019b). Dual quaternion based powered descent guidance with state-triggered constraints. *AIAA Journal of Guidance, Control, and Dynamics*, Accepted. arXiv:1904.09248.

Richards, A., & How, J. (2003a). Model predictive control of vehicle maneuvers with guaranteed completion time and robust feasibility. In *Proceedings of the 2003 American control conference*. IEEE, http://dx.doi.org/10.1109/acc.2003.1240467.

Richards, A., & How, J. (2003b). Performance evaluation of rendezvous using model predictive control. In *AIAA guidance, navigation, and control conference and exhibit*. American Institute of Aeronautics and Astronautics, http://dx.doi.org/10.2514/6.2003-5507.

Richards, A., & How, J. P. (2006). Robust variable horizon model predictive control for vehicle maneuvering. *International Journal of Robust and Nonlinear Control*, *16*(7), 333–351. http://dx.doi.org/10.1002/rnc.1059.

Richards, A., Schouwenaars, T., How, J. P., & Feron, E. (2002). Spacecraft trajectory planning with avoidance constraints using mixed-integer linear programming. *Journal of Guidance, Control, and Dynamics*, *25*(4), 755–764. http://dx.doi.org/10.2514/2.4943.

Ridderhof, J., & Tsiotras, P. (2018). Uncertainty quantication and control during Mars powered descent and landing using covariance steering. In *2018 AIAA guidance, navigation, and control conference*. American Institute of Aeronautics and Astronautics, http://dx.doi.org/10.2514/6.2018-0611.

Ridderhof, J., & Tsiotras, P. (2019). Minimum-fuel powered descent in the presence of random disturbances. In *AIAA SciTech 2019 forum*. American Institute of Aeronautics and Astronautics, http://dx.doi.org/10.2514/6.2019-0646.

Robertson, E. A. (2017). Synopsis of precision landing and hazard avoidance (PL&HA) capabilities for space exploration. In *AIAA guidance, navigation, and control conference*. American Institute of Aeronautics and Astronautics, http://dx.doi.org/10.2514/6.2017-1897.

Robinson, M. (2018). Lunar reconnaissance orbiter camera (LROC). http://lroc.sese.asu.edu/posts/993. Accessed: 11/26/2019.

Rockafellar, R. T. (1970). *Convex analysis*. Princeton, NJ: Princeton University Press, http://dx.doi.org/10.2307/j.ctvcm4g0s.

Rockafellar, R. T. (1993). Lagrange multipliers and optimality. *SIAM Review*, *35*(2), 183–238. http://dx.doi.org/10.1137/1035044.

Ross, I., & D'Souza, C. (2004). Rapid trajectory optimization of multi-agent hybrid systems. In *AIAA guidance, navigation, and control conference and exhibit*. American Institute of Aeronautics and Astronautics, http://dx.doi.org/10.2514/6.2004-5422.

Ross, I. M., & D'Souza, C. N. (2005). Hybrid optimal control framework for mission planning. *Journal of Guidance, Control, and Dynamics*, *28*(4), 686–697. http://dx.doi.org/10.2514/1.8285.

Ross, I. M., Gong, Q., & Sekhavat, P. (2007). Low-thrust, high-accuracy trajectory optimization. *Journal of Guidance, Control, and Dynamics*, *30*(4), 921–933. http://dx.doi.org/10.2514/1.23181.

Ross, I. M., & Karpenko, M. (2012). A review of pseudospectral optimal control: From theory to flight. *Annual Reviews in Control*, *36*(2), 182–197. http://dx.doi.org/10.1016/j.arcontrol.2012.09.002.

Russell, R. P. (2007). Primer vector theory applied to global low-thrust trade studies. *Journal of Guidance, Control, and Dynamics*, *30*(2), 460–472. http://dx.doi.org/10.2514/1.22984.

Sagliano, M. (2018a). Generalized hp pseudospectral convex programming for powered descent and landing. In *2018 AIAA guidance, navigation, and control conference*. American Institute of Aeronautics and Astronautics, http://dx.doi.org/10.2514/6.2018-1870.

Sagliano, M. (2018b). Pseudospectral convex optimization for powered descent and landing. *Journal of Guidance, Control, and Dynamics*, *41*(2), 320–334. http://dx.doi.org/10.2514/1.g002818.

Sagliano, M. (2019). Generalized hp pseudospectral-convex programming for powered descent and landing. *Journal of Guidance, Control, and Dynamics*, *42*(7), 1562–1570. http://dx.doi.org/10.2514/1.g003731.

Sagliano, M., & Mooij, E. (2018). Optimal drag-energy entry guidance via pseudospectral convex optimization. In *2018 AIAA guidance, navigation, and control conference*. American Institute of Aeronautics and Astronautics, http://dx.doi.org/10.2514/6.2018-1315.

Sagliano, M., Mooij, E., & Theil, S. (2017). Onboard trajectory generation for entry vehicles via adaptive multivariate pseudospectral interpolation. *Journal of Guidance, Control, and Dynamics*, *40*(2), 466–476. http://dx.doi.org/10.2514/1.g001817.

San Martín, A. M., Lee, S. W., & Wong, E. C. (2013). The development of the MSL guidance, navigation, and control system for entry, descent, and landing. In *23rd space flight mechanics meeting*.

Sanchez, J. C., Gavilan, F., & Vazquez, R. (2018). A predictive guidance algorithm for autonomous asteroid soft landing. *IFAC-PapersOnLine*, *51*(12), 6–11. http://dx.doi.org/10.1016/j.ifacol.2018.07.080.

Sanchez, J. C., Gavilan, F., Vazquez, R., & Louembet, C. (2020). A flatness-based predictive controller for six-degrees of freedom spacecraft rendezvous. *Acta Astronautica*, *167*, 391–403. http://dx.doi.org/10.1016/j.actaastro.2019.11.026.

Sánchez-Sánchez, C., & Izzo, D. (2018). Real-time optimal control via deep neural networks: Study on landing problems. *Journal of Guidance, Control, and Dynamics*, *41*(5), 1122–1135. http://dx.doi.org/10.2514/1.g002357.

Saranathan, H., & Grant, M. J. (2018). Relaxed autonomously switched hybrid system approach to indirect multiphase aerospace trajectory optimization. *Journal of Spacecraft and Rockets*, *55*(3), 611–621. http://dx.doi.org/10.2514/1.a34012.

Scharf, D. P., Açıkmeşe, B., Dueri, D., Benito, J., & Casoliva, J. (2017). Implementation and experimental demonstration of onboard powered-descent guidance. *Journal of Guidance, Control, and Dynamics*, *40*(2), 213–229. http://dx.doi.org/10.2514/1.g000399.

Scharf, D. P., Açıkmeşe, A. B., Ploen, S. R., & Hadaegh, F. Y. (2006). A direct solution for fuel-optimal reactive collision avoidance of collaborating spacecraft. In *2006 American Control Conference (ACC)*. IEEE, http://dx.doi.org/10.1109/ACC.2006.1657548.

Scharf, D., Hadaegh, F., & Ploen, S. (2003). A survey of spacecraft formation flying guidance and control (part 1): Guidance. In *Proceedings of the 2003 American control conference, 2003.* IEEE, http://dx.doi.org/10.1109/acc.2003.1239845.

Scharf, D. P., Ploen, S. R., & Açıkmeşe, B. A. (2015). Interpolation-enhanced powered descent guidance for onboard nominal, off-nominal, and multi-x scenarios. In *AIAA guidance, navigation, and control conference*. American Institute of Aeronautics and Astronautics, http://dx.doi.org/10.2514/6.2015-0850.

Scharf, D. P., Regehr, M. W., Vaughan, G. M., Benito, J., Ansari, H., Aung, M., et al. (2014). ADAPT demonstrations of onboard large-divert guidance with a VTVL rocket. In *2014 IEEE aerospace conference*. IEEE, http://dx.doi.org/10.1109/aero.2014.6836462.

Scheeres, D., Ostro, S., Hudson, R., DeJong, E., & Suzuki, S. (1998). Dynamics of orbits close to asteroid 4179 Toutatis. *Icarus*, *132*(1), 53–79. http://dx.doi.org/10.1006/icar.1997.5870.

Schouwenaars, T., Richards, A., Feron, E., & How, J. (2001). Plume avoidance maneuver planning using mixed integer linear programming. In *AIAA guidance, navigation, and control conference and exhibit*. American Institute of Aeronautics and Astronautics, http://dx.doi.org/10.2514/6.2001-4091.

Schulman, J., Duan, Y., Ho, J., Lee, A., Awwal, I., Bradlow, H., et al. (2014). Motion planning with sequential convex optimization and convex collision checking. *International Journal of Robotics Research*, *33*(9), 1251–1270. http://dx.doi.org/10.1177/0278364914528132.

Scorsoglio, A., Furfaro, R., Linares, R., & Massari, M. (2019). Actor-critic reinforcement learning approach to relative motion guidance in near-rectilinear orbit. In *AAS/AIAA space flight mechanics meeting*.

Seywald, K., & Seywald, H. (2019). Desensitized optimal control. In *AIAA SciTech 2019 forum*. American Institute of Aeronautics and Astronautics, http://dx.doi.org/10.2514/6.2019-0651.

Shen, H., Seywald, H., & Powell, R. W. (2010). Desensitizing the minimum-fuel powered descent for Mars pinpoint landing. *Journal of Guidance, Control, and Dynamics*, *33*(1), 108–115. http://dx.doi.org/10.2514/1.44649.

Shi, Y., & Wang, Z. (2020). A deep learning-based approach to real-time trajectory optimization for hypersonic vehicles. In *AIAA SciTech 2020 forum*. American Institute of Aeronautics and Astronautics, http://dx.doi.org/10.2514/6.2020-0023.

Shirazi, A., Ceberio, J., & Lozano, J. A. (2018). Spacecraft trajectory optimization: A review of models, objectives, approaches and solutions. *Progress in Aerospace Sciences*, *102*, 76–98. http://dx.doi.org/10.1016/j.paerosci.2018.07.007.

Simplício, P., Marcos, A., & Bennani, S. (2019). Guidance of reusable launchers: Improving descent and landing performance. *Journal of Guidance, Control, and Dynamics*, *42*(10), 2206–2219. http://dx.doi.org/10.2514/1.g004155.

Simplício, P., Marcos, A., Joffre, E., Zamaro, M., & Silva, N. (2018). Review of guidance techniques for landing on small bodies. *Progress in Aerospace Sciences*, *103*, 69–83. http://dx.doi.org/10.1016/j.paerosci.2018.10.005.

Singh, G., Macala, G., Wong, E., Rasmussen, R., Singh, G., Macala, G., et al. (1997). A constraint monitor algorithm for the Cassini spacecraft. In *Guidance, navigation, and control conference*. American Institute of Aeronautics and Astronautics, http://dx.doi.org/10.2514/6.1997-3526.

Skogestad, S., & Postlethwaite, I. (2005). *Multivariable feedback control: Analysis and design* (2nd ed.). Wiley.

van Soest, W., Chu, Q. P., & Mulder, J. A. (2006). Combined feedback linearization and constrained model predictive control for entry flight. *Journal of Guidance, Control, and Dynamics*, *29*(2), 427–434. http://dx.doi.org/10.2514/1.14511.

Song, Y., & Gong, S. (2019). Solar-sail deep space trajectory optimization using successive convex programming. *Astrophysics and Space Science*, *364*(7), http://dx.doi.org/10.1007/s10509-019-3597-x.

Song, Z., Wang, C., Theil, S., Seelbinder, D., Sagliano, M., Liu, X., et al. (2020). Survey of autonomous guidance methods for powered planetary landing. *Frontiers of Information Technology & Electronic Engineering*, http://dx.doi.org/10.1631/fitee.1900458.

Starek, J. A., Açıkmeşe, B., Nesnas, I. A., & Pavone, M. (2015). Spacecraft autonomy challenges for next-generation space missions. In *Advances in control system technology for aerospace applications* (pp. 1–48). Springer Berlin Heidelberg, http://dx.doi.org/10.1007/978-3-662-47694-9_1.

Starek, J. A., & Kolmanovsky, I. V. (2012). Nonlinear model predictive control strategy for low thrust spacecraft missions. *Optimal Control Applications & Methods*, *35*(1), 1–20. http://dx.doi.org/10.1002/oca.2049.

Starek, J. A., Schmerling, E., Maher, G. D., Barbee, B. W., & Pavone, M. (2017). Fast, safe, propellant-efficient spacecraft motion planning under Clohessy-Wiltshire-Hill dynamics. *Journal of Guidance, Control, and Dynamics*, *40*(2), 418–438. http://dx.doi.org/10.2514/1.g001913.

Stein, G. (2003). Respect the unstable. *IEEE Control Systems*, *23*(4), 12–25. http://dx.doi.org/10.1109/mcs.2003.1213600.

Steinfeldt, B. A., Grant, M. J., Matz, D. A., Braun, R. D., & Barton, G. H. (2010). Guidance, navigation, and control system performance trades for Mars pinpoint landing. *Journal of Spacecraft and Rockets*, *47*(1), 188–198. http://dx.doi.org/10.2514/1.45779.

Steltzner, A. D., San Martín, A. M., Rivellini, T. P., Chen, A., & Kipp, D. (2014). Mars Science Laboratory entry, descent, and landing system development challenges. *Journal of Spacecraft and Rockets*, *51*(4), 994–1003. http://dx.doi.org/10.2514/1.a32866.

Stephens, J.-P., Vos, G. A., Bilimoria, K. D., Mueller, E. R., Brazzel, J., & Spehar, P. (2013). Orion handling qualities during international space station proximity operations and docking. *Journal of Spacecraft and Rockets*, *50*(2), 449–457. http://dx.doi.org/10.2514/1.a32253.

Stevens, R., & Ross, I. M. (2005). Preliminary design of Earth-Mars cyclers using solar sails. *Journal of Spacecraft and Rockets*, *42*(1), 132–137. http://dx.doi.org/10.2514/1.2947.

Stevens, R., Ross, I. M., & Matousek, S. E. (2004). Earth-Mars Return trajectories using solar sails. In *55th international astronautical congress*. American Institute of Aeronautics and Astronautics, http://dx.doi.org/10.2514/6.iac-04-a.2.08.

Sullivan, J., Grimberg, S., & D'Amico, S. (2017). Comprehensive survey and assessment of spacecraft relative motion dynamics models. *Journal of Guidance, Control, and Dynamics*, *40*(8), 1837–1859. http://dx.doi.org/10.2514/1.g002309.

Sun, C., Dai, R., & Lu, P. (2019). Multi-phase spacecraft mission optimization by quadratically constrained quadratic programming. In *AIAA SciTech 2019 forum* (pp. 1–15). American Institute of Aeronautics and Astronautics, http://dx.doi.org/10.2514/6.2019-1667.

Sundström, O., & Guzzella, L. (2009). A generic dynamic programming Matlab function. In *2009 IEEE international conference on control applications*. IEEE, http://dx.doi.org/10.1109/cca.2009.5281131.

Szmuk, M., & Açıkmeşe, B. (2018). Successive convexification for 6-DoF Mars rocket powered landing with free-final-time. In *2018 AIAA guidance, navigation, and control conference*. American Institute of Aeronautics and Astronautics, http://dx.doi.org/10.2514/6.2018-0617.

Szmuk, M., Açıkmeşe, B., & Berning, A. W. (2016). Successive convexification for fuel-optimal powered landing with aerodynamic drag and non-convex constraints. In *AIAA guidance, navigation, and control conference*. American Institute of Aeronautics and Astronautics, http://dx.doi.org/10.2514/6.2016-0378.

Szmuk, M., Eren, U., & Açıkmeşe, B. (2017). Successive convexification for Mars 6-DoF powered descent landing guidance. In *AIAA guidance, navigation, and control conference*. American Institute of Aeronautics and Astronautics, http://dx.doi.org/10.2514/6.2017-1500.

Szmuk, M., Malyuta, D., Reynolds, T. P., Mceowen, M. S., & Acikmese, B. (2019). Real-time quad-rotor path planning using convex optimization and compound state-triggered constraints. In *2019 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, http://dx.doi.org/10.1109/iros40897.2019.8967706.

Szmuk, M., Reynolds, T. P., & Açıkmeşe, B. (2018). Successive convexification for real-time 6-DoF powered descent guidance with state-triggered constraints. In *AIAA Journal of Guidance, Control, and Dynamics*. arXiv:1811.10803, Accepted.

Szmuk, M., Reynolds, T. P., Açıkmeşe, B., Mesbahi, M., & Carson III, J. M. (2019). Successive convexification for 6-DoF powered descent guidance with compound state-triggered constraints. In *AIAA SciTech 2019 forum*. American Institute of Aeronautics and Astronautics, http://dx.doi.org/10.2514/6.2019-0926.

Tabuada, P., & Fraile, L. (2020). Data-driven stabilization of SISO feedback linearizable systems. In *59th IEEE conference on decision and control*. IEEE, p. arXiv:2003.14240.

Taheri, E., & Junkins, J. L. (2018). Generic smoothing for optimal bang-off-bang spacecraft maneuvers. *Journal of Guidance, Control, and Dynamics*, *41*(11), 2470–2475. http://dx.doi.org/10.2514/1.g003604.

Taheri, E., & Junkins, J. L. (2019). How many impulses redux. *The Journal of the Astronautical Sciences*, *67*(2), 257–334. http://dx.doi.org/10.1007/s40295-019-00203-1.

Taheri, E., Junkins, J. L., Kolmanovsky, I., & Girard, A. (2020a). A novel approach for optimal trajectory design with multiple operation modes of propulsion system, part 1. *Acta Astronautica*, *172*, 151–165. http://dx.doi.org/10.1016/j.actaastro.2020.02.042.

Taheri, E., Junkins, J. L., Kolmanovsky, I., & Girard, A. (2020b). A novel approach for optimal trajectory design with multiple operation modes of propulsion system, part 2. *Acta Astronautica*, *172*, 166–179. http://dx.doi.org/10.1016/j.actaastro.2020.02.047.

Taheri, E., Kolmanovsky, I., & Atkins, E. (2016). Enhanced smoothing technique for indirect optimization of minimum-fuel low-thrust trajectories. *Journal of Guidance, Control, and Dynamics*, *39*(11), 2500–2511. http://dx.doi.org/10.2514/1.g000379.

Taheri, E., Li, N. I., & Kolmanovsky, I. (2017). Co-state initialization for the minimum-time low-thrust trajectory optimization. *Advances in Space Research*, *59*(9), 2360–2373. http://dx.doi.org/10.1016/j.asr.2017.02.010.

Tam, M., & Lightsey, E. G. (2016). Constrained spacecraft reorientation using mixed integer convex programming. *Acta Astronautica*, *127*, 31–40. http://dx.doi.org/10.1016/j.actaastro.2016.04.003.

Tang, G., Jiang, F., & Li, J. (2018). Fuel-optimal low-thrust trajectory optimization using indirect method and successive convex programming. *IEEE Transactions on Aerospace and Electronic Systems*, *54*(4), 2053–2066. http://dx.doi.org/10.1109/taes.2018.2803558.

Tassa, Y., Mansard, N., & Todorov, E. (2014). Control-limited differential dynamic programming. In *2014 IEEE international conference on robotics and automation (ICRA)*. IEEE, http://dx.doi.org/10.1109/icra.2014.6907001.

Tavernini, L. (1987). Differential automata and their discrete simulators. *Nonlinear Analysis: Theory, Methods & Applications*, *11*(6), 665–683. http://dx.doi.org/10.1016/0362-546x(87)90034-4.

Tewari, A. (2011). *Automatic control of atmospheric and space flight vehicles*. Boston: Birkhäuser, http://dx.doi.org/10.1007/978-0-8176-4864-0.

Tian, B., & Zong, Q. (2011). Optimal guidance for reentry vehicles based on indirect Legendre pseudospectral method. *Acta Astronautica*, *68*(7–8), 1176–1184. http://dx.doi.org/10.1016/j.actaastro.2010.10.010.

Tillerson, M., Inalhan, G., & How, J. P. (2002). Co-ordination and control of distributed spacecraft systems using convex optimization techniques. *International Journal of Robust and Nonlinear Control*, *12*(2–3), 207–242. http://dx.doi.org/10.1002/rnc.683.

Topcu, U., Casoliva, J., & Mease, K. (2005). Fuel efficient powered descent guidance for Mars landing. In *AIAA guidance, navigation, and control conference and exhibit*. American Institute of Aeronautics and Astronautics, http://dx.doi.org/10.2514/6.2005-6286.

Topcu, U., Casoliva, J., & Mease, K. D. (2007). Minimum-fuel powered descent for Mars pinpoint landing. *Journal of Spacecraft and Rockets*, *44*(2), 324–331. http://dx.doi.org/10.2514/1.25023.

Trélat, E. (2012). Optimal control and applications to aerospace: Some results and challenges. *Journal of Optimization Theory and Applications*, *154*(3), 713–758. http://dx.doi.org/10.1007/s10957-012-0050-5.

Tsiotras, P., & Mesbahi, M. (2017). Toward an algorithmic control theory. *Journal of Guidance, Control, and Dynamics*, *40*(2), 194–196. http://dx.doi.org/10.2514/1.g002754.

Vazquez, R., Gavilan, F., & Camacho, E. F. (2011). Trajectory planning for spacecraft rendezvous with on/off thrusters. *IFAC Proceedings Volumes*, *44*(1), 8473–8478. http://dx.doi.org/10.3182/20110828-6-it-1002.02445.

Vazquez, R., Gavilan, F., & Camacho, E. F. (2014). Trajectory planning for spacecraft rendezvous in elliptical orbits with on/off thrusters. *IFAC Proceedings Volumes*, *47*(3), 9703–9708. http://dx.doi.org/10.3182/20140824-6-za-1003.02197.

Vazquez, R., Gavilan, F., & Camacho, E. F. (2015). Model predictive control for spacecraft rendezvous in elliptical orbits with on/off thrusters. *IFAC-PapersOnLine*, *48*(9), 251–256. http://dx.doi.org/10.1016/j.ifacol.2015.08.092.

Vazquez, R., Gavilan, F., & Camacho, E. (2017). Pulse-width predictive control for LTV systems with application to spacecraft rendezvous. *Control Engineering Practice*, *60*, 199–210. http://dx.doi.org/10.1016/j.conengprac.2016.06.017.

Ventura, J., Ciarcià, M., Romano, M., & Walter, U. (2017). Fast and near-optimal guidance for docking to uncontrolled spacecraft. *Journal of Guidance, Control, and Dynamics*, *40*(12), 3138–3154. http://dx.doi.org/10.2514/1.g001843.

Volpe, R., & Circi, C. (2019). Optical-aided, autonomous and optimal space rendezvous with a non-cooperative target. *Acta Astronautica*, *157*, 528–540. http://dx.doi.org/10.1016/j.actaastro.2019.01.020.

Wächter, A., & Biegler, L. T. (2005). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, *106*(1), 25–57. http://dx.doi.org/10.1007/s10107-004-0559-y.

Walsh, A., & Forbes, J. R. (2018). Constrained attitude control on SO(3) via semidefinite programming. *Journal of Guidance, Control, and Dynamics*, *41*(11), 2483–2488. http://dx.doi.org/10.2514/1.g003259.

Wan, C., Dai, R., & Lu, P. (2019). Alternating minimization algorithm for polynomial optimal control problems. *Journal of Guidance, Control, and Dynamics*, *42*(4), 723–736. http://dx.doi.org/10.2514/1.g003766.

Wang, Z. (2019a). Maximum-normal-load entry trajectory optimization for hypersonic glide vehicles. In *AIAA SciTech 2019 forum*. American Institute of Aeronautics and Astronautics, http://dx.doi.org/10.2514/6.2019-0262.

Wang, Z. (2019b). Optimal trajectories and normal load analysis of hypersonic glide vehicles via convex optimization. *Aerospace Science and Technology*, *87*, 357–368. http://dx.doi.org/10.1016/j.ast.2019.03.002.

Wang, J., & Cui, N. (2018). A pseudospectral-convex optimization algorithm for rocket landing guidance. In *2018 AIAA guidance, navigation, and control conference*. American Institute of Aeronautics and Astronautics, http://dx.doi.org/10.2514/6.2018-1871.

Wang, J., Cui, N., & Wei, C. (2019a). Optimal rocket landing guidance using convex optimization and model predictive control. *Journal of Guidance, Control, and Dynamics*, *42*(5), 1078–1092. http://dx.doi.org/10.2514/1.g003518.

Wang, J., Cui, N., & Wei, C. (2019b). Rapid trajectory optimization for hypersonic entry using a pseudospectral-convex algorithm. *Proceedings of the Institution of Mechanical Engineers, Part G (Journal of Aerospace Engineering)*, *233*(14), 5227–5238. http://dx.doi.org/10.1177/0954410019840839.

Wang, Z., & Grant, M. J. (2017). Constrained trajectory optimization for planetary entry via sequential convex programming. *Journal of Guidance, Control, and Dynamics*, *40*(10), 2603–2615. http://dx.doi.org/10.2514/1.g002150.

Wang, Z., & Grant, M. J. (2018a). Autonomous entry guidance for hypersonic vehicles by convex optimization. *Journal of Spacecraft and Rockets*, *55*(4), 993–1006. http://dx.doi.org/10.2514/1.a34102.

Wang, Z., & Grant, M. J. (2018b). Minimum-fuel low-thrust transfers for spacecraft: A convex approach. *IEEE Transactions on Aerospace and Electronic Systems*, *54*(5), 2274–2290. http://dx.doi.org/10.1109/taes.2018.2812558.

Wang, Z., & Grant, M. J. (2018c). Optimization of minimum-time low-thrust transfers using convex programming. *Journal of Spacecraft and Rockets*, *55*(3), 586–598. http://dx.doi.org/10.2514/1.a33995.

Wang, Z., & Grant, M. J. (2019). Improved sequential convex programming algorithms for entry trajectory optimization. In *AIAA SciTech 2019 forum*. American Institute of Aeronautics and Astronautics, http://dx.doi.org/10.2514/6.2019-0667.

Wang, X., Wang, Z., & Zhang, Y. (2018). Model predictive control to autonomously approach a failed spacecraft. *International Journal of Aerospace Engineering*, *2018*, 1–18. http://dx.doi.org/10.1155/2018/7428535.

Watson, L. T. (2002). Probability-one homotopies in computational science. *Journal of Computational and Applied Mathematics*, *140*(1–2), 785–807. http://dx.doi.org/10.1016/s0377-0427(01)00473-3.

Way, D. W., Powell, R. W., Chen, A., Steltzner, A. D., San Martín, A. M., Burkhart, P. D., et al. (2007). Mars Science Laboratory: Entry, descent, and landing system performance. In *2007 IEEE aerospace conference*. IEEE, http://dx.doi.org/10.1109/aero.2007.352821.

Weiss, A., Baldwin, M., Erwin, R. S., & Kolmanovsky, I. (2015). Model predictive control for spacecraft rendezvous and docking: Strategies for handling constraints and case studies. *IEEE Transactions on Control Systems Technology*, *23*(4), 1638–1647. http://dx.doi.org/10.1109/tcst.2014.2379639.

Weiss, A., Kolmanovsky, I., Baldwin, M., & Erwin, R. S. (2012). Model predictive control of three dimensional spacecraft relative motion. In *2012 American control conference (ACC)*. IEEE, http://dx.doi.org/10.1109/acc.2012.6314862.

Weiss, A., Petersen, C., Baldwin, M., Erwin, R. S., & Kolmanovsky, I. (2015). Safe positively invariant sets for spacecraft obstacle avoidance. *Journal of Guidance, Control, and Dynamics*, *38*(4), 720–732. http://dx.doi.org/10.2514/1.g000115.

Werner, R., & Scheeres, D. (1997). Exterior gravitation of a polyhedron derived and compared with harmonic and mascon gravitation representations of asteroid 4769 Castalia. *Celestial Mechanics & Dynamical Astronomy*, *65*(3), http://dx.doi.org/10.1007/bf00053511.

Whiffen, G. (2006). Mystic: Implementation of the static dynamic optimal control algorithm for high-fidelity, low-thrust trajectory design. In *AIAA/AAS astrodynamics specialist conference and exhibit*. American Institute of Aeronautics and Astronautics, http://dx.doi.org/10.2514/6.2006-6741.

Whiffen, G. J., & Sims, J. A. (2001). Application of a novel optimal control algorithm to low-thrust trajectory optimization. In *2001 AAS/AIAA space flight mechanics meeting* (pp. 2–6).

Woffinden, D. C., & Geller, D. K. (2007). Navigating the road to autonomous orbital rendezvous. *Journal of Spacecraft and Rockets*, *44*(4), 898–909. http://dx.doi.org/10.2514/1.30734.

Woffinden, D., Robinson, S., Williams, J., & Putnam, Z. R. (2019). Linear covariance analysis techniques to generate navigation and sensor requirements for the safe and precise landing integrated capabilities evolution (SPLICE) project. In *AIAA SciTech 2019 forum*. American Institute of Aeronautics and Astronautics, http://dx.doi.org/10.2514/6.2019-0662.

Wolf, A., Casoliva, J., Manrique, J. B., Açıkmeşe, B., & Ploen, S. (2012). Improving the landing precision of an MSL-class vehicle. In *2012 IEEE aerospace conference*. IEEE, http://dx.doi.org/10.1109/aero.2012.6187005.

Wright, M. H. (2005). The interior-point revolution in optimization: History, recent developments, and lasting consequences. *American Mathematical Society. Bulletin. New Series*, *42*, 39–56.

Wright, M. H. (2011). Fast times in linear programming: Early success, revolutions, and mysteries. URL: https://sites.math.washington.edu/mac/talks/20111021wright.pdf. Accessed: 20/03/2020.

Wu, C., & Han, X. (2019). Energy-optimal spacecraft attitude maneuver path-planning under complex constraints. *Acta Astronautica*, *157*, 415–424. http://dx.doi.org/10.1016/j.actaastro.2018.12.028.

Wu, C., Xu, R., Zhu, S., & Cui, P. (2017). Time-optimal spacecraft attitude maneuver path planning under boundary and pointing constraints. *Acta Astronautica*, *137*, 128–137. http://dx.doi.org/10.1016/j.actaastro.2017.04.004.

Xiaojun, C., Hutao, C., Pingyuan, C., & Rui, X. (2010). Large angular autonomous attitude maneuver of deep spacecraft using pseudospectral method. In *2010 3rd international symposium on systems and control in aeronautics and astronautics*. IEEE, http://dx.doi.org/10.1109/isscaa.2010.5632498.

Xie, Z., Liu, C. K., & Hauser, K. (2017). Differential dynamic programming with nonlinear constraints. In *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, http://dx.doi.org/10.1109/icra.2017.7989086.

Xu, Z., Chen, X., Huang, Y., Bai, Y., & Chen, Q. (2018). Collision prediction and avoidance for satellite ultra-close relative motion with zonotope-based reachable sets. *Proceedings of the Institution of Mechanical Engineers, Part G (Journal of Aerospace Engineering)*, *233*(11), 3920–3937. http://dx.doi.org/10.1177/0954410018810255.

Xue, S., & Lu, P. (2010). Constrained predictor-corrector entry guidance. *Journal of Guidance, Control, and Dynamics*, *33*(4), 1273–1281. http://dx.doi.org/10.2514/1.49557.

Yamanaka, K., & Ankersen, F. (2002). New state transition matrix for relative motion on an arbitrary elliptical orbit. *Journal of Guidance, Control, and Dynamics*, *25*(1), 60–66. http://dx.doi.org/10.2514/2.4875.

Yang, H., Bai, X., & Baoyin, H. (2017). Rapid generation of time-optimal trajectories for asteroid landing via convex optimization. *Journal of Guidance, Control, and Dynamics*, *40*(3), 628–641. http://dx.doi.org/10.2514/1.g002170.

Yang, H., & Baoyin, H. (2015). Fuel-optimal control for soft landing on an irregular asteroid. *IEEE Transactions on Aerospace and Electronic Systems*, *51*(3), 1688–1697. http://dx.doi.org/10.1109/taes.2015.140295.

You, S., Wan, C., Dai, R., Lu, P., & Rea, J. R. (2020). Learning-based optimal control for planetary entry, powered descent and landing guidance. In *AIAA SciTech 2020 forum*. American Institute of Aeronautics and Astronautics, http://dx.doi.org/10.2514/6.2020-0849.

You, S., Wan, C., Dai, R., & Rea, J. R. (2020). Real-time optimal control via deep neural networks: Study on landing problems. *Journal of Guidance, Control, and Dynamics*, Accepted.

Zagaris, C., Park, H., Virgili-Llop, J., Zappulla, R., Romano, M., & Kolmanovsky, I. (2018). Model predictive control of spacecraft relative motion with convexified keep-out-zone constraints. *Journal of Guidance, Control, and Dynamics*, *41*(9), 2054–2062. http://dx.doi.org/10.2514/1.g003549.

Zanelli, A., Domahidi, A., Jerez, J., & Morari, M. (2017). FORCES NLP: An efficient implementation of interior-point methods for multistage nonlinear nonconvex programs. *International Journal of Control*, *93*(1), 13–29. http://dx.doi.org/10.1080/00207179.2017.1316017.

Zarchan, P. (Ed.), (2019). *Tactical and strategic missile guidance, seventh edition - SET*. American Institute of Aeronautics and Astronautics, Inc., http://dx.doi.org/10.2514/4.105845.

Zhang, S. J., Açıkmeşe, B., Swei, S. S.-M., & Prabhu, D. (2015). Convex programming approach to real-time trajectory optimization for Mars aerocapture. In *2015 IEEE aerospace conference*. IEEE, http://dx.doi.org/10.1109/aero.2015.7119111.

Zhang, Y., Guo, Y., Ma, G., & Zeng, T. (2017). Collision avoidance ZEM/ZEV optimal feedback guidance for powered descent phase of landing on Mars. *Advances in Space Research*, *59*(6), 1514–1525. http://dx.doi.org/10.1016/j.asr.2016.12.040.

Zhang, Y., Huang, J., & Cui, H. (2020). Trajectory optimization for asteroid landing with two-phase free final time. *Advances in Space Research*, *65*(4), 1210–1224. http://dx.doi.org/10.1016/j.asr.2019.11.031.

Zhang, K., Yang, S., & Xiong, F. (2019). Rapid ascent trajectory optimization for guided rockets via sequential convex programming. *Proceedings of the Institution of Mechanical Engineers, Part G (Journal of Aerospace Engineering)*, *233*(13), 4800–4809. http://dx.doi.org/10.1177/0954410019830268.

Zhang, J., Zhao, S., Zhang, Y., & Li, Y. (2015). Optimal planning approaches with multiple impulses for rendezvous based on hybrid genetic algorithm and control method. *Advances in Mechanical Engineering*, *7*(3), Article 168781401557378. http://dx.doi.org/10.1177/1687814015573783.

Zhao, J., & Shang, T. (2018). Dynamic optimization using local collocation methods and improved multiresolution technique. *Applied Sciences*, *8*(9), 1680. http://dx.doi.org/10.3390/app8091680.

Zhao, D.-J., & Song, Z.-Y. (2017). Reentry trajectory optimization with waypoint and no-fly zone constraints using multiphase convex programming. *Acta Astronautica*, *137*, 60–69. http://dx.doi.org/10.1016/j.actaastro.2017.04.013.

Zhengxiang, S., Tao, C., Songyan, W., & Ming, Y. (2018). Convex method for ascent trajectory optimization using iterative narrowing trust region. In *2018 IEEE CSAA guidance, navigation and control conference (CGNCC)*. IEEE, http://dx.doi.org/10.1109/gncc42960.2018.9019086.

Zhou, D., Zhang, Y., & Li, S. (2019). Receding horizon guidance and control using sequential convex programming for spacecraft 6-DOF close proximity. *Aerospace Science and Technology*, *87*, 459–477. http://dx.doi.org/10.1016/j.ast.2019.02.041.

Zhu, S., Sun, R., Wang, J., Wang, J., & Shao, X. (2018). Robust model predictive control for multi-step short range spacecraft rendezvous. *Advances in Space Research*, *62*(1), 111–126. http://dx.doi.org/10.1016/j.asr.2018.03.037.

Zimmer, S., Ocampo, C., & Bishop, R. (2010). Reducing orbit covariance for continuous thrust spacecraft transfers. *IEEE Transactions on Aerospace and Electronic Systems*, *46*(2), 771–791. http://dx.doi.org/10.1109/taes.2010.5461656.