Real-Time Sequential Conic Optimization
for Multi-Phase Rocket Landing Guidance[★]

Abhinav G. Kamath[*] **Purnanand Elango**[*] **Yue Yu**[**]
Skye Mceowen[*] **Govind M. Chari**[*] **John M. Carson III**[***]
Behçet Açıkmeşe[*]

[*] *William E. Boeing Department of Aeronautics & Astronautics,*
University of Washington, Seattle, WA 98195, USA
(e-mail: {`agkamath, pelango, skye95, gchari, behcet`}`@uw.edu`).
[**] *Oden Institute for Computational Engineering and Sciences, The*
University of Texas at Austin, Austin, TX 78712, USA
(e-mail: `yueyu@utexas.edu`).
[***] *NASA Johnson Space Center, Houston, TX 77058, USA*
(e-mail: `john.m.carson@nasa.gov`).

Abstract: We introduce a multi-phase rocket landing guidance framework that can handle nonlinear dynamics and does not mandate any additional mixed-integer or nonconvex constraints to handle discrete temporal events/switching. To achieve this, we first introduce *sequential conic optimization* (SeCO), a new paradigm for solving nonconvex optimal control problems that is entirely devoid of matrix factorizations and inversions. This framework combines sequential convex programming (SCP) and first-order conic optimization and can solve unified multi-phase trajectory optimization problems in real-time. The novel features of this framework are: (1) *time-interval dilation*, which enables multi-phase trajectory optimization with free-transition-time; (2) *single-crossing compound state-triggered constraints*, which are entirely convex if the trigger and constraint conditions are convex; (3) *virtual state*, which is a new approach to handling artificial infeasibility in SCP methods that preserves the shapes of the constraint sets; and, (4) the use of the proportional-integral projected gradient method (PIPG), a high-performance first-order conic optimization solver, in tandem with the penalized trust region (PTR) SCP algorithm. We demonstrate the efficacy and real-time capability of SeCO by solving a relevant multi-phase rocket landing guidance problem with nonlinear dynamics and convex constraints only, and observe that our solver is 2.7 times faster than a state-of-the-art convex optimization solver.

Keywords: Real-time optimal control; convex optimization; rocket landing; guidance & control

1. INTRODUCTION

Rocket landing guidance can be considered to be the generalization of powered-descent guidance (PDG) to include the unpowered phase(s) of flight. Precision landing techniques for orbital rockets harness convex optimization for real-time trajectory generation (Açıkmeşe and Ploen, 2007; Blackmore, 2016). One such method, known as *lossless convexification*, was the first convex optimization-based algorithm to compute a rocket landing guidance trajectory for a mid-flight large divert maneuver onboard the vehicle. Sequential convex programming (SCP) techniques for rocket landing have recently emerged as a way to handle more generalized nonconvexities in the dynamics, state constraints, environmental constraints, and path constraints (Mao et al. (2016)). Such SCP algorithms have been demonstrated to successfully solve a wide range of 6-DoF rocket landing problems (Szmuk et al. (2019); Reynolds et al. (2020b); Szmuk et al. (2020)).

Pseudospectral methods have been developed in the last decade for solving multi-phase trajectory optimization problems, particularly in aerospace applications. These methods typically parameterize the state time-history with Chebyshev and Legendre polynomials, operate on nonuniform time-grids with node points that are roots of these polynomials, and use linking conditions to tie together different phases of flight. A multi-phase Radau pseudospectral method was introduced in (Garrido and Sagliano, 2021) for ascent and powered-descent guidance. The SPARTAN software package described in (Sagliano et al., 2021) has been demonstrated to solve multi-phase problems arising in space applications. Further, relevant details showcasing pseudospectral approaches are provided in (Hwang and Ahn, 2022; Ma et al., 2019; Zhang and Zhang, 2022). Despite their ability to solve multi-phase problems, pseudospectral methods are typically untenable for real-time implementations (Malyuta et al., 2019).

In this paper, we present sequential conic optimization (SeCO), a novel matrix-inverse-free paradigm for solving nonconvex optimal control problems in real-time, using which we formulate and solve a multi-phase rocket landing guidance problem in a unified manner. The guidance algorithm has the ability to perform mid-burn engine

switching after ignition. The primary features of the SeCO framework are as follows: (1) time-interval dilation, which allows for nonuniform time-grids and free-phase-transition-time; (2) a new formulation of compound state-triggered constraints—called *single-crossing* compound state-triggered constraints—that is convex, provided the trigger and constraint conditions are convex; (3) a new approach to handling artificial infeasibility, i.e., *virtual state*, that preserves the shapes of the constraint sets and does not alter the dynamics manifold; and (4) PIPG (proportional-integral projected gradient), a high-performance first-order solver that effectively exploits the structure of trajectory optimization problems, making it well-suited for embedded applications (Yu et al. (2020)). Further, we use an inverse-free exact discretization method to generate dynamically feasible solutions. Despite the layers of apparent complexity described, the method that we present is implemented via a low-footprint codebase that is easy to verify and validate.

2. DYNAMICS

2.1 Time-interval dilation

The original nonlinear dynamics, governing the evolution of state $x(t) \in \mathbb{R}^{n_x}$ with control input $u(t) \in \mathbb{R}^{n_u}$, over the entire time-horizon, are given by Equation (1).

$$\dot{x}(t) = f(t, x(t), u(t)), \quad t \in [0, t_f) \qquad (1)$$

Now, we consider the dynamics in the sub-interval $[t_k, t_{k+1})$, where $0 < t_k < t_{k+1} < t_N = t_f^-$, $k = 1 : N-1$—where $a : b$ denotes the range of integers between (and including) integers a and b—and define an affine map, $\tau_k(t)$, as shown in Equation (2).

$$\tau_k(t) := \frac{t - t_k}{t_{k+1}^- - t_k} \ni \tau_k(t) : [t_k, t_{k+1}) \to [0, 1) \qquad (2)$$

This mapping is referred to as *time-interval dilation*, as it normalizes the wall-clock time-interval to a known fixed interval—in our case $[0, 1)$—by either shrinking or expanding—and hence *dilating*—the original time-interval. Next, we apply the derivative operator with respect to the dilated time τ_k, denoted by $\mathring{\Box}$, to Equation (1), and invoke the chain-rule, as shown in Equation (3), where $t \in [t_k, t_{k+1})$.

$$\mathring{x}(t) = \frac{d}{d\tau_k} x(t) = \frac{dt}{d\tau_k} \frac{d}{dt} x(t) = \frac{dt}{d\tau_k} \dot{x}(t) = \underbrace{(t_{k+1}^- - t_k)}_{s_k} \dot{x}(t)$$

$$= s_k f(t, x(t), u(t)) := F(t, x(t), u(t), s_k) \qquad (3)$$

The multiplier in Equation (3), $s_k := t_{k+1}^- - t_k \in \mathbb{R}_+$, which is nothing but the length of the kth wall-clock time-interval, is termed the *dilation factor*. By treating a phase-based subset of s_k, $k = 1 : N-1$, i.e., such that $k \subseteq 1 : N-1$, as decision variables and discretizing the system over them, we allow the optimizer to decide what the temporal spacing of discrete nodes should be in each phase rather than use a uniform temporal grid over the entire horizon. In the approach we propose, this is the key to enabling free-transition-time multi-phase trajectory optimization within a single-shot optimization framework, without requiring any mixed-integer or nonconvex constraints to handle the discrete temporal events/switching.

Although it is possible to allow each dilation factor to be an independent decision variable, we choose to partition the

temporal grid based on the phases of flight, and evenly space the temporal nodes within each phase. This measure is taken to mitigate extreme inter-sample constraint violation, which tends to occur when fully adaptive grids are used. The time-dilated dynamics given by Equation (3) will be used henceforth. Note that using the time-dilated dynamics given by Equation (3) in lieu of Equation (1) converts the original *free-final-time* optimal control problem to an equivalent *fixed-final-time* optimal control problem, with the effective horizon being $[0, N-1)$.

2.2 Linearization

A convex approximation of the original nonconvex optimal control problem is obtained by linearizing the nonlinear dynamics (Equation (3))—which leads to a linear time-varying (LTV) system—and keeping the convex constraints intact. The state, control, and parameter constraint sets are assumed to belong to a set that has separable closed-form projection operations. We stress that most of the common constraints in trajectory optimization problems naturally fit this template (Malyuta et al., 2022). Nonconvex constraints can be handled using this framework too, by means of either linearization or convex approximation.

2.3 Discretization

We assume a first-order hold (FOH) on the control input signal throughout the horizon and a zero-order hold (ZOH) on the control input signal only during discrete switching events such as engine startup and downselection. Both of these control parameterizations have two key properties that make them attractive for optimal control applications: (1) inter-sample satisfaction of the convex control constraints is guaranteed (provided they are satisfied at the discrete temporal nodes), which is in contrast to pseudospectral methods; and, (2) the resulting conic subproblem has a sparsity pattern that is amenable to real-time implementation (Malyuta et al., 2019; Szmuk et al., 2020).

In the FOH case, the control profile is parameterized as shown in Equation (4), where $t \in [t_k, t_{k+1})$ and $\tau_k \in [0, 1)$, as given by Equation (2).

$$u(\tau_k) = (1 - \tau_k) u_k + \tau_k u_{k+1}, \quad k = 1 : N-1 \qquad (4)$$

The LTV dynamics can now be written: (1) using the piecewise-affine control input parameterization given by Equation (4); and, (2) in terms of deviations from the reference, as shown in Equation (5). The reference quantities are denoted by $\overline{\Box}$, and $\Delta\Box$ denotes the deviation of a quantity from its reference, i.e., $\Delta\Box := \Box - \overline{\Box}$, and $\Delta\mathring{x}(\tau_k) := \mathring{x}(\tau_k) - F(\tau_k, \overline{x}(\tau_k), \overline{u}(\tau_k), \overline{s}_k)$. The approximate nature of the equation is an artifact of linearization of the original nonlinear dynamics via truncation of the higher-order (≥ 2) terms in the Taylor series expansion.

$$\Delta\mathring{x}(\tau_k) \approx A(\tau_k) \Delta x(\tau_k) + B(\tau_k) (1 - \tau_k) \Delta u_k \\ + B(\tau_k) \tau_k \Delta u_{k+1} + S(\tau_k) \Delta s_k \qquad (5)$$

The *state transition matrix* (STM) associated with Equation (5), denoted by $\Phi(\tau_k, 0)$, $\tau_k \in [0, 1)$, satisfies the following matrix differential equation: $\mathring{\Phi}(\tau_k, 0) = A(\tau_k) \Phi(\tau_k, 0)$, with $\Phi(0, 0) = I_{n_x}$. The unique solution to Equation (5) is given by Equation (6) (Antsaklis and Michel, 2006; Malyuta et al., 2022).

$$\Delta x(\tau_k) = \Phi(\tau_k, 0)\,\Delta x(0) + \int_0^{\tau_k} \Phi(\tau_k, \zeta)\,\cdot \qquad (6)$$

$$\cdot\,\{B(\zeta)\,(1-\tau_k)\,\Delta u_k + B(\zeta)\,\tau_k\,\Delta u_{k+1} + S(\zeta)\,\Delta s_k\}\,\mathrm{d}\zeta$$
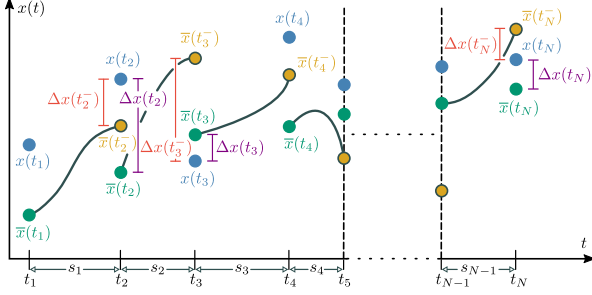

Fig. 1. Propagation of the state. $\Delta x(t_1) := 0$, where $t_1 := 0$. The *stitching condition* for $k = 1\!:\!N-1$ is given by $\Delta x(t_{k+1}^-) + \overline{x}(t_{k+1}^-) = \Delta x(t_{k+1}) + \overline{x}(t_{k+1}) = x(t_{k+1})$.

Evaluating Equation (6) at $\tau_k = 1^-$, we get Equation (7). We replace the limits 0 and 1 with 0_k and 1_k, respectively, to explicitly indicate the dependence on index k, i.e., $\tau_k(t_k) := 0_k$ and $\tau_k(t_{k+1}^-) := 1_k^-$.

$$\Delta x(1_k^-) = A_k \Delta x(0_k) + B_k^- \Delta u_k + B_k^+ \Delta u_{k+1} + S_k \Delta s_k \quad (7)$$

A_k, B_k^-, B_k^+, and S_k can be computed as the solution to the initial value problem (IVP) given by Equations (8), respectively, integrated from 0_k to 1_k^-.

$$\mathring{\Psi}_A(\zeta) = A(\zeta)\,\Psi_A(\zeta) \tag{8a}$$

$$\mathring{\Psi}_{B^-}(\zeta) = A(\zeta)\,\Psi_{B^-}(\zeta) + B(\zeta)\,(1-\zeta) \tag{8b}$$

$$\mathring{\Psi}_{B^+}(\zeta) = A(\zeta)\,\Psi_{B^+}(\zeta) + B(\zeta)\,\zeta \tag{8c}$$

$$\mathring{\Psi}_S(\zeta) = A(\zeta)\,\Psi_S(\zeta) + S(\zeta) \tag{8d}$$

where function $\Psi_A(\zeta)$ is defined such that $\zeta \mapsto \Phi(\zeta, 0_k)$, and the initial conditions for Equations (8) are: $\Psi_A(0_k) = I_{n_x}$, $\Psi_{B^-}(0_k) = \Psi_{B^+}(0_k) = 0_{n_x \times n_u}$, and $\Psi_S(0_k) = 0_{n_x}$. Note that Equation (9), which we refer to as the *stitching condition*, holds, as is evident from Figure 1.

$$\Delta x(1_k^-) + \overline{x}(1_k^-) = \Delta x(1_k) + \overline{x}(1_k) \tag{9}$$

The discretized dynamics can now be given by Equation (10), where $\Delta x_k := \Delta x(0_k)$, $x_{k+1}^{\mathrm{prop}} := \overline{x}(1_k^-)$, $\overline{x}_{k+1} := \overline{x}(1_k)$, and $\overline{u}(\tau_k) := (1-\tau_k)\,\overline{u}_k + \tau_k\,\overline{u}_{k+1}$, for $1 \leq k \leq N-1$.

$$\Delta x_{k+1} = A_k \Delta x_k + B_k^- \Delta u_k + B_k^+ \Delta u_{k+1} + S_k \Delta s_k \\ + x_{k+1}^{\mathrm{prop}} - \overline{x}_{k+1} \tag{10}$$

The discretized dynamics in terms of the absolute variables are recovered from Equation (10), as shown in Equation (11).

$$x_{k+1} = A_k\,x_k + B_k^-\,u_k + B_k^+\,u_{k+1} + S_k\,s_k + \tag{11}$$
$$x_{k+1}^{\mathrm{prop}} - \left(A_k\,\overline{x}_k + B_k^-\,\overline{u}_k + B_k^+\,\overline{u}_{k+1} + S_k\,\overline{s}_k\right)$$

Equation (11) represents an *exact* discretization of the LTV dynamics, which means that the error between the continuous-time trajectory and the discrete-time trajectory at the discrete temporal nodes is analytically zero.

In order to obtain the corresponding expressions for the ZOH case, the following changes are incorporated: (1) Equation (8b) is replaced by Equation (12) (and accordingly, B_k^- is replaced by B_k); and, (2) B_k^+ is set to a zero matrix with the same dimensions.

$$\mathring{\Psi}_B(\zeta) = A(\zeta)\,\Psi_B(\zeta) + B(\zeta) \tag{12}$$

3. STATE-TRIGGERED CONSTRAINTS

We introduce a specialized formulation of compound state-triggered constraints (STCs) for problems in which the trigger functions are activated only once and are strictly monotonic in a neighborhood around which they are activated. We refer to these constraints as *single-crossing* compound state-triggered constraints.

Let $g(\cdot)$ be a trigger function that is said to be *activated* on the set $\{x \mid g(x) \leq g^\star\}$ for some trigger value $g^\star$. Further, let $g^{-1}(g^\star)$ be a well-defined pre-image. Then, $g(\cdot)$ is called a *single-crossing* trigger function if $x^\star := g^{-1}(g^\star)$ is a singleton and $g(\cdot)$ is strictly monotonic in a neighborhood around $x^\star$. Such a formulation is especially useful in applications such as rocket landing that require certain STCs to be satisfied for mission success, wherein it is reasonable to expect the trigger conditions to be activated once and only once. For instance, it is reasonable to expect/require a rocket in descent from a certain initial altitude, h_i, with its target landing location at the origin, to certainly cross a trigger altitude $0 < h_{\mathrm{trigger}} < h_i$ once during its descent, and not surpass that altitude after.

For the purpose of demonstration, we consider a compound STC with one trigger condition and multiple constraint conditions to be imposed with the AND logic. Let $g(\cdot)$ be the trigger function and $c^j(\cdot)$, $j = 1\!:\!n_c$, be the constraint functions, where n_c is the number of constraint conditions. The purpose of the compound STC (Szmuk et al., 2019, Section II.B) is to satisfy the condition given in Equation (13), where $x(t)$ is the state. $\forall t \in [0, t_f)$,

$$g(x(t)) \leq 0 \Rightarrow \bigwedge_{j=1}^{n_c} c^j(x(t)) \leq 0 \tag{13}$$

In maneuvers with a single-crossing trigger condition, the trigger condition is activated once and only once, i.e., $g(x(t)) = 0$ is guaranteed to activate at some $t = t_{\mathrm{trigger}}$, $g(x(t)) > 0 \ \forall t \in [0, t_{\mathrm{trigger}})$, and $g(x(t)) < 0$ $\forall t \in (t_{\mathrm{trigger}}, t_f)$. Using this fact, the single-crossing compound STC is formulated as shown in Equation (14). We emphasize that t_{trigger} itself is free, and hence the compound constraint is state-triggered and not time-triggered.

$$g(x(t)) \geq 0 \qquad \forall t \in [0, t_{\mathrm{trigger}}) \tag{14a}$$

$$g(x(t)) = 0, \qquad t = t_{\mathrm{trigger}} \tag{14b}$$

$$\left.\begin{array}{c} g(x(t)) \leq 0 \\ c^1(x(t)) \leq 0 \\ \vdots \\ c^{n_c}(x(t)) \leq 0 \end{array}\right\} \ \forall t \in (t_{\mathrm{trigger}}, t_f) \tag{14c}$$

If the trigger conditions and the constraint conditions above are individually convex, the compound STC is entirely convex. This is in contrast to existing formulations of STCs in the literature that are inherently nonconvex, regardless of the convexity of the trigger and constraint conditions (Szmuk et al., 2019, 2020; Reynolds et al., 2020b). However, those methods are more general, in that they do not mandate the trigger condition to be single-crossing—we trade off this generality for convexity/simplicity in our method. Further, we note that the single-crossing STC formulation bears resemblance to the method adopted in (Bhasin, 2016).

The aforementioned approaches in the literature use uniform temporal spacing of discrete nodes over the entire horizon. As a result, along with the fact that inter-sample constraint satisfaction is typically not guaranteed in general, these approaches do not guarantee the imposition of the constraints exactly at the specified trigger conditions. The triggering of these constraints (in time) is only accurate up to the spacing of the grid, and the corresponding solutions usually demonstrate violation of these constraints with respect to the triggers. This issue becomes more prevalent in maneuvers over very long time horizons, especially when the set of feasible trigger windows is much smaller in comparison, and can be detrimental to mission success if accurate triggering is required.

We leverage time-interval dilation to impose single-crossing STCs, and treat the windows within which these constraints need to be imposed as distinct phases of flight. This allows for a fine grid in phases that involve critical constraints that need to be satisfied to ensure mission success, and a coarser grid in the more benign phases of flight—thus enabling one-shot multi-phase trajectory optimization. If the solution converges to a feasible trajectory, the STCs are guaranteed to be satisfied at the triggers (since the triggers are imposed as waypoints as in Equation (14b)), and at every discrete temporal node within the trigger window.

4. SEQUENTIAL CONIC OPTIMIZATION

4.1 Virtual state

Artificial infeasibility refers to the phenomenon wherein a subproblem can become infeasible as a result of linearization of the nonconvex constraints even if there exists a feasible solution to the original problem. Typically, unconstrained yet heavily penalized slack variables are added to the linearized constraints, so as to ensure that the subproblem is always feasible.

We propose a new approach to handling artificial infeasibility by means of a *virtual state* variable, which serves as a copy of the original state. This approach helps decouple the dynamics and control constraints from the state constraints and exactly satisfy all the path constraints at each solver iteration, while ensuring that the subproblem never turns infeasible. If x is the actual state variable, u is the control variable, and ξ is the virtual state variable, the dynamics constraint is imposed on x and u, the control constraints are imposed on u, and all the state constraints are imposed on ξ.

To ensure that the dynamics and all other constraints are satisfied at convergence, we minimize the error between x and ξ by heavily penalizing the squared distance between them in the objective function. The virtual state does not alter the dynamics manifold (unlike the virtual control approach (Szmuk et al., 2020; Reynolds et al., 2020b)), and preserves the shapes of the conic state constraint sets (unlike the virtual buffer approach (Malyuta et al., 2022)).

4.2 Conic subproblem

We impose a soft trust region on the decision variable and use the penalized trust region (PTR) algorithm (Szmuk et al., 2020; Reynolds et al., 2020a,b). The discretized conic subproblem with virtual state(s) and a soft trust region is shown in Problem (15), which is strongly convex.

$$\min_{u,s} \quad w_{\mathrm{c}}J(x_N) + \tfrac{1}{2}(w_{\mathrm{tr}}J_{\mathrm{tr}} + w_{\mathrm{vse}}J_{\mathrm{vse}}) \tag{15a}$$

$$\text{s.t.} \quad x_{k+1} = \text{RHS of Eq. (11)}, \ k=1:N-1 \tag{15b}$$

$$\xi_k \in \mathcal{X}_\nabla, \qquad\qquad k=1:N \tag{15c}$$

$$u_k \in \mathcal{U}_\nabla, \qquad\qquad k=1:N \tag{15d}$$

$$s_k \in \mathcal{S}_\nabla, \qquad\qquad k=1:n_{\mathrm{phase}} \tag{15e}$$

where $x \in \mathbb{R}^{n_x}$ is the state vector, $\xi \in \mathbb{R}^{n_x}$ is the *virtual state* vector, $u \in \mathbb{R}^{n_u}$ is the control input vector, and $s \in \mathbb{R}_{++}$ is the dilation factor (vector); $\mathcal{X}_\nabla$, $\mathcal{U}_\nabla$, and $\mathcal{S}_\nabla$ are the state, control, and temporal constraint sets, respectively, which are assumed to be closed and convex; $J(x_N)$ is the original cost function, assumed to be in the Mayer form (Berkovitz, 2013), $J_{\mathrm{tr}} := \sum_{k=1}^{N}\left(\|x_k - \overline{x}_k\|_2^2 + \|u_k - \overline{u}_k\|_2^2 \right) + \sum_{k=1}^{n_{\mathrm{phase}}} \|s_k - \overline{s}_k\|_2^2$ is the trust region penalty, and $J_{\mathrm{vse}} := \sum_{k=1}^{N} \|x_k - \xi_k\|_2^2$ is the virtual state error penalty, w_c, w_{tr}, and w_{vse} being their respective weights.

Proposition 1. The virtual state error penalty, given by $\sum_{k=1}^{N} \|x_k - \xi_k\|_2^2$, is convex.

Proof. Let $y_k := (x_k, \xi_k)$. $\therefore \|x_k - \xi_k\|_2^2 = y_k^\top M y_k$, where $M = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} \otimes I_{n_x}$. Since $\operatorname{spec} M \in \{0,2\}$, M is positive semidefinite (PSD). Therefore, the quadratic form, $y_k^\top M y_k$, is convex. Since the sum of convex functions is convex, $\sum_{k=1}^{N} y_k^\top M y_k = \sum_{k=1}^{N} \|x_k - \xi_k\|_2^2$ is also convex. ∎

Problem (15) can be vectorized, i.e., assembled into the form of Problem (16), by stacking all the decision variables into a single vector, z. For more details, see (Yu et al., 2022a).

$$\min_{z} \quad \tfrac{1}{2}z^\top Q z + \langle q, z \rangle \tag{16a}$$

$$\text{s.t.} \quad Hz - h = 0 \tag{16b}$$

$$z \in \mathbb{D} \tag{16c}$$

5. HIGH-PERFORMANCE SOLVER

The proportional-integral projected gradient method (PIPG) (Yu et al., 2020, 2022b) is a first-order primal-dual optimization algorithm for solving conic optimization problems such as the one given by Equations (16). PIPG is compatible with extrapolation, which has been shown to improve its practical convergence behavior. Here, we use the extrapolated PIPG algorithm, also denoted by xPIPG, given by the iterative sequence in Equation (17) (Yu et al., 2022a).

$$z^{j+1} = \pi_{\mathbb{D}}\big[\zeta^j - \alpha\big(Q\zeta^j + q + H^\top\eta^j\big)\big] \tag{17a}$$

$$w^{j+1} = \eta^j + \beta\big(H\big(2z^{j+1} - \zeta^j\big) - h\big) \tag{17b}$$

$$\zeta^{j+1} = (1-\rho)\zeta^j + \rho z^{j+1} \tag{17c}$$

$$\eta^{j+1} = (1-\rho)\eta^j + \rho w^{j+1} \tag{17d}$$

where the primal variable, z^j, converges to an optimum of Problem (16) as $j \to \infty$. The step-sizes, α and β, are computed in accordance with (Yu et al., 2022a, Lemma 2), and $\rho \in [1,2)$ is the extrapolation factor. We leverage the warm-starting capability of PIPG and use the ℓ_2-hypersphere preconditioning technique described in (Kamath et al., 2023) to further accelerate convergence.

6. MULTI-PHASE ROCKET LANDING GUIDANCE

As a representative example for multi-phase rocket landing guidance via SeCO, we consider the problem of terrestrial precision landing of a vehicle akin to Starship, which is designed to be a fully reusable rocket and currently in development (Shotwell and Blackmore, 2021).

We consider a nonlinear planar model of the vehicle, given by Equations (18), with $\hat{x}$ being the vertical axis and $\hat{z}$ being the horizontal axis in the inertial frame (and the longitudinal and lateral axes in the body frame, respectively). We use a simple, tractable aerodynamic model to account for the aerodynamic effects on the vehicle during descent (Szmuk et al., 2020). In practice, however, numerical databases can be used along with SeCO (Mceowen and Açıkmeşe, 2022). Further, with our approach, aerodynamic-free polynomial coast-phase ballistic trajectory predictions, such as the one provided in (Szmuk et al., 2020), are not required. We note that (Lee and Lee, 2022) provide an SCP-based 6-DoF multi-phase rocket landing guidance implementation, but impose nonconvex STCs and implement inexact (trapezoidal) discretization, unlike our approach.

The *original* state vector is defined as follows: $x(t) := (m(t), r(t), v(t), \theta(t), \omega(t))$, where $m(t) \in \mathbb{R}_+$ is the mass, $r(t) \in \mathbb{R}^2$ is the position, $v(t) \in \mathbb{R}^2$ is the velocity, $\theta(t) \in \mathbb{R}$ is the body tilt angle with respect to the inertial vertical, and $\omega(t) \in \mathbb{R}$ is the angular velocity of the body.

The control input vector is defined as follows: $u(t) := (T(t), \delta(t))$, where $T(t) \in \mathbb{R}$ is the thrust magnitude and $\delta(t) \in \mathbb{R}$ is the gimbal deflection angle.

Further, we define a *virtual state* vector and impose all the state constraints, including the boundary conditions, on this variable, in accordance with Subsection 4.1: $\xi(t) := \left(m^\xi(t), r^\xi(t), v^\xi(t), \theta^\xi(t), \omega^\xi(t) \right)$.

$$\dot{m}(t) = -\alpha_e T(t) \tag{18a}$$

$$\dot{r}(t) = v(t) \tag{18b}$$

$$\dot{v}(t) = \frac{1}{m(t)} (F_\mathcal{I}(t) + A_\mathcal{I}(t)) + g \tag{18c}$$

$$\dot{\theta}(t) = \omega(t) \tag{18d}$$

$$\dot{\omega}(t) = \frac{1}{J(t)} \left(F_{\mathcal{B}_{\hat{z}}}(t)\, l_{cm} - A_{\mathcal{B}_{\hat{z}}}(t)\, l_{cp}^j \right) \tag{18e}$$

where

$$\alpha_e := \frac{1}{I_{sp} g_0} \tag{19a}$$

$$F_\mathcal{I}(t) := T(t) \begin{pmatrix} \cos(\theta(t) + \delta(t)) \\ -\sin(\theta(t) + \delta(t)) \end{pmatrix} \tag{19b}$$

$$A_\mathcal{I}(t) := R_{\mathcal{I}\leftarrow\mathcal{B}}(t) A_\mathcal{B}(t) \tag{19c}$$

$$F_\mathcal{B}(t) := T(t) \begin{pmatrix} \cos(\delta(t)) \\ -\sin(\delta(t)) \end{pmatrix} \tag{19d}$$

$$A_\mathcal{B}(t) := -\frac{\rho_{air} S_{area} \|v(t)\|_2}{2} C_{aero} R_{\mathcal{I}\leftarrow\mathcal{B}}^\top(t) v(t) \tag{19e}$$

$$R_{\mathcal{I}\leftarrow\mathcal{B}}(t) := \begin{pmatrix} \cos\theta(t) & \sin\theta(t) \\ -\sin\theta(t) & \cos\theta(t) \end{pmatrix} \tag{19f}$$

Here, $\alpha_e \in \mathbb{R}_{++}$ is the thrust-specific fuel consumption (TSFC), $I_{sp} \in \mathbb{R}_{++}$ is the specific impulse of the rocket engine, $g_0 \in \mathbb{R}_{++}$ is standard Earth gravitational acceleration, $g := (-g_0, 0)$, $R_{\mathcal{I}\leftarrow\mathcal{B}}(t) \in \mathrm{SO}(2)$ is the rotation matrix that maps coordinates in the body frame to the inertial frame, $\rho_{air} \in \mathbb{R}_{++}$ is the ambient atmospheric density, $S_{area} \in \mathbb{R}_{++}$ is the reference area, $C_{aero} := \mathrm{diag}\{c_{\hat{x}}, c_{\hat{z}}\}$ is the aerodynamic coefficient matrix, where $c_{\hat{x}}, c_{\hat{z}} \in \mathbb{R}_{++}$ are the aerodynamic coefficients along the body $\hat{x}$ and $\hat{z}$ axes, respectively, and $J(t) \in \mathbb{R}_{++}$ is the moment of inertia of the vehicle about the body $\hat{y}$ axis (out-of-plane). The body of the vehicle is assumed to be a uniform solid cylinder, and hence, the moment of inertia about its central diameter is given by $J(t) := m(t)\left(\frac{l_r^2}{4} + \frac{l_h^2}{12}\right)$, where $l_r \in \mathbb{R}_{++}$ and $l_h \in \mathbb{R}_{++}$ are the radius and height of the fuselage, respectively.

The engines are assumed to be co-located, and the location of the vehicle mass-center is assumed to be fixed in the body frame. The thrust moment-arm (the distance between the vehicle mass-center and the engine gimbal hinge point) is denoted by $l_{cm} \in \mathbb{R}_{++}$, and $l_{cp}^j \in \mathbb{R}_+$ is the aerodynamic moment-arm (the distance between the vehicle mass-center and the center-of-pressure), where $j \in \{0,1\}$; $j = 0$ for the unpowered phase of flight and $j = 1$ for the powered phases of flight—l_{cp}^0 is assumed to be maintained at zero via independent aerodynamic controls by means of forward and aft flaps (aerodynamic control surfaces), i.e., the center-of-pressure and the mass-center are assumed to be coincident when the vehicle is in the coast phase.

At the engine ignition (PDI) epoch, it is assumed that the following events occur: (1) the forward flaps are fully extended (to maximize drag towards the nose-cone); and, (2) the aft flaps are fully folded (to minimize drag towards the aft section of the vehicle). As a result, the center-of-pressure shifts away from the mass-center, towards the nose-cone, and induces an aerodynamic torque (pitching moment) on the vehicle. Hence, l_{cp}^1 is set to a nonzero value, and it is kept fixed for the remainder of the trajectory. This, along with gimbaling of the rocket engines, is used to induce the flip maneuver to get the vehicle upright in preparation for terminal descent.

The guidance problem is partitioned into four phases: (1) the unpowered, subsonic coast phase; (2) the high-thrust (3-engine) burn phase; (3) the low-thrust (1-engine) burn phase; and, (4) the altitude-triggered terminal descent phase, separated by the following important discrete events/epochs: (1) powered-descent initiation (PDI) or engine ignition, $t_{ignition}$; (2) engine downselection or switching from a triple-engine burn to a single-engine burn, t_{switch}; and, (3) altitude-based triggering of the terminal descent phase, $t_{trigger}$. We emphasize that putting all of these phases together implicitly leads to a free-ignition-time, free-engine-switching-time, *and* free-final-time optimal control problem (subject to the temporal constraints imposed).

For the discretized problem, the temporal grid we choose is as given by Equation (20), where N is the number of discrete temporal nodes and $k_{ignition}$, k_{switch}, and $k_{trigger}$ are the nodes at which the discrete events occur.

$$k \in \{1, \ldots, k_{ignition}, \ldots, k_{switch}, \ldots, k_{trigger}, \ldots, N\} \tag{20}$$

6.1 Common constraints

The dynamics and temporal constraints are imposed over the entire horizon. The dynamics constraint is given by Equation (15b). The constraint given by Equation (21) is imposed on the dilation factors to ensure they are bounded.
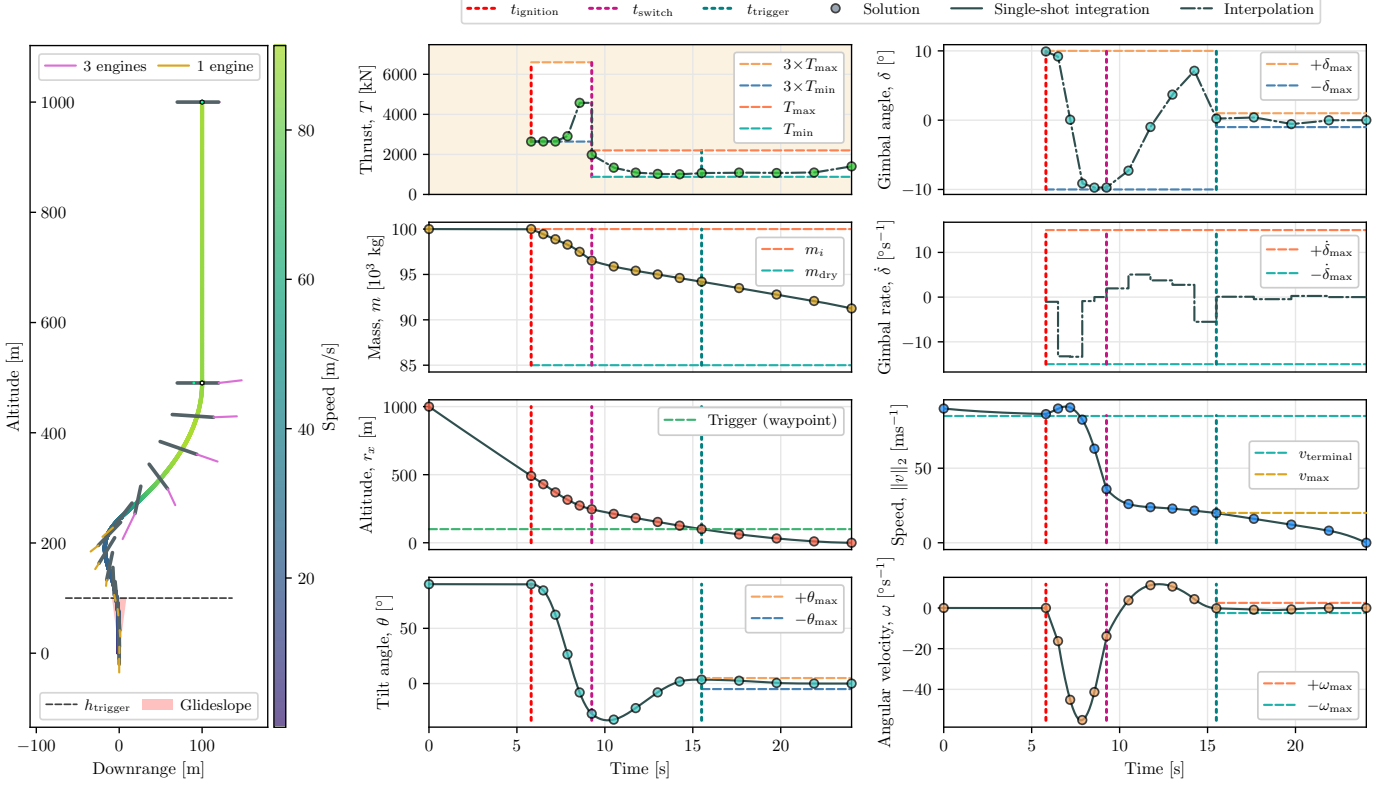
Fig. 2. A real-time multi-phase rocket landing guidance solution obtained via SeCO.

$$s_{\min} \leq s_l \leq s_{\max}, \quad l \in \{1\!:\!n_{\text{phase}}\} \tag{21}$$

In order to enable the imposition of a single-crossing compound STC in the altitude-triggered terminal descent phase, we impose a minimum altitude constraint in the first three phases, as given by Equation (22).

$$r_{x_k}^{\xi} \geq h_{\text{trigger}}, \quad k \in \{1\!:\!k_{\text{trigger}}-1\} \tag{22}$$

6.2 The unpowered coast phase, $k \in \{1\!:\!k_{\text{ignition}}-1\}$

The center-of-pressure is coincident with the mass-center, i.e., $j = 0$ and $l_{\text{cp}}^j = l_{\text{cp}}^0 = 0$ in the dynamics. In implementation, the thrust magnitude is set to zero (the gimbal angle is inconsequential), and a zero-order hold (ZOH) is assumed on the control input signal for this phase, in order to avoid control input constraint violation when the engines are ignited in the next phase. A first-order hold (FOH) can also be assumed here, if the engine startup time is significant and needs to be taken into account (with appropriate constraints on the dilation factor).

The following initial conditions are imposed:

$$\begin{aligned} m_1^{\xi} = m_i, \ r_1^{\xi} = r_i, \ v_1^{\xi} = v_i, \\ \theta_1^{\xi} = \theta_i, \ \omega_1^{\xi} = \omega_i, \ T_1 = 0 \end{aligned} \tag{23}$$

where $m_i, r_i, v_i, \theta_i,$ and ω_i are the initial values for the mass, position, velocity, body tilt angle, and angular velocity, respectively. The thrust magnitude at the first node is constrained to be zero.

6.3 The high-thrust burn phase, $k \in \{k_{\text{ignition}}\!:\!k_{\text{switch}}-1\}$

The center-of-pressure shifts towards the nose-cone of the vehicle, i.e., $j = 1$ and $l_{\text{cp}}^j = l_{\text{cp}}^1 > 0$ in the dynamics.

We model this as a discrete change in its value, which is then held constant for the remainder of the trajectory. The following constraints are imposed on the control variables:

$$3\,T_{\min} \leq T_k \leq 3\,T_{\max} \tag{24}$$

$$\begin{aligned} \max\{-\delta_{\max}, -\dot{\delta}_{\max}\bar{s}_{k-1} + \bar{\delta}_{k-1}\} \leq \delta_k \\ \leq \min\{\delta_{\max}, \dot{\delta}_{\max}\bar{s}_{k-1} + \bar{\delta}_{k-1}\} \end{aligned} \tag{25}$$

where $T_{\min}$ and $T_{\max}$ are the lower and upper bounds on the thrust magnitude for a single engine, respectively, $-\delta_{\max}$ and $\delta_{\max}$ are the lower and upper bounds on the gimbal deflection angle, respectively, and $-\dot{\delta}_{\max}$ and $\dot{\delta}_{\max}$ are the lower and upper bounds on the gimbal rate, respectively. Equation (25) combines the gimbal angle and rate constraints, by using reference values in the lower and upper bounds to make them constants and hence, avoid overlapping projections on the same variable. This constraint is, however, exact at convergence, and this formulation has been observed to work well in practice. Although there are only n_{phase} dilation factor decision variables, we consider the length of $\bar{s}$ to be $N-1$, such that the dilation factors are repeated to span each phase.

ZOH is assumed on the control input signal between $k_{\text{switch}}-1$ and k_{switch}, in preparation for engine downselection, which marks the beginning of the next phase. FOH can also be assumed here, if the engine shutdown time is significant and needs to be accounted for. In order to ensure that the gimbal deflection angle profile does not have any discontinuities, we set the gimbal rate to zero between these nodes, as shown in Equation (26).

$$\delta_{k_{\text{switch}}} = \bar{\delta}_{k_{\text{switch}}-1} \tag{26}$$

6.4 The low-thrust burn phase, $k \in \{k_{\text{switch}} : k_{\text{trigger}} - 1\}$

This phase is similar to the high-thrust burn phase, apart from the fact that the switch from 3 to 1 engines occurs at k_{switch}, and the bounds on the thrust magnitude are changed accordingly, as shown in Equation (27). The combined gimbal constraint is left unchanged from the previous phase, and is shown in Equation (28). FOH is assumed on the control input signal in this phase, and for the remainder of the trajectory.

$$T_{\min} \leq T_k \leq T_{\max} \tag{27}$$

$$\max\{-\delta_{\max}, -\dot{\delta}_{\max}\bar{s}_{k-1} + \bar{\delta}_{k-1}\} \leq \delta_k \leq \min\{\delta_{\max}, \dot{\delta}_{\max}\bar{s}_{k-1} + \bar{\delta}_{k-1}\} \tag{28}$$

6.5 The terminal descent phase, $k \in \{k_{\text{trigger}} : N\}$

The final phase, the terminal descent phase, is the most heavily and tightly constrained phase of flight. This is designed as such in order to enable closed-loop precision landing, i.e, to ensure that the generated guidance trajectories (the feedforward control input signal and the reference state profiles) are amenable to tight tracking via feedback controllers. Such a phase would be especially useful if sub-meter touchdown accuracy is required, for instance, if the vehicle is to be retrieved by the launch tower itself (Atkinson, 2022).

The constraint on the thrust magnitude is left unchanged from the previous phase, and is shown in Equation (29).

$$T_{\min} \leq T_k \leq T_{\max} \tag{29}$$

An altitude-triggered single-crossing compound STC, with five constraint conditions, is imposed, as shown in Equations (30). The constraint conditions include the following: maximum speed, maximum tilt, maximum angular speed, glideslope, and tighter gimbal deflection bounds. The altitude constraint forms the trigger condition. All of these constraints are imposed in the interval $k \in \{k_{\text{trigger}} : N-1\}$.

$$r_{x_k}^{\xi} \begin{cases} = h_{\text{trigger}}, & \text{if } k = k_{\text{trigger}} \\ \leq h_{\text{trigger}}, & \text{otherwise} \end{cases} \tag{30a}$$

$$|r_{z_k}^{\xi}| \leq \begin{cases} \tan\gamma_{\text{gs}}\, h_{\text{trigger}}, & \text{if } k = k_{\text{trigger}} \\ \tan\gamma_{\text{gs}}\, \bar{r}_{x_k}^{\xi}, & \text{otherwise} \end{cases} \tag{30b}$$

$$\|v_k^{\xi}\|_2 \leq v_{\max} \tag{30c}$$

$$|\theta_k^{\xi}| \leq \theta_{\max} \tag{30d}$$

$$|\omega_k^{\xi}| \leq \omega_{\max} \tag{30e}$$

$$\max\{-\delta_{\max_{\text{TD}}}, -\dot{\delta}_{\max}\bar{s}_{k-1} + \bar{\delta}_{k-1}\} \leq \delta_k \leq \min\{\delta_{\max_{\text{TD}}}, \dot{\delta}_{\max}\bar{s}_{k-1} + \bar{\delta}_{k-1}\} \tag{30f}$$

The altitude and glideslope constraints are treated differently at the trigger epoch and after. The altitude constraint is posed as an equality at the trigger—this ensures that the constraint conditions are exactly satisfied at the trigger. Further, the glideslope constraint is cast in the form of box constraints in terms of the reference values, in order to enable closed-form projections (without this measure, there would be two constraints on $r_{x_k}^{\xi}$ at every temporal node, thus precluding closed-form projections). Similar to the combined gimbal constraint, this constraint is exact at convergence. The bounds on the gimbal deflection angle are tightened in this phase as well, i.e., $\delta_{\max_{\text{TD}}} < \delta_{\max}$.

The following terminal boundary conditions are imposed:

$$m_N^{\xi} \geq m_{\text{dry}}, \; r_N^{\xi} = r_f, \; v_N^{\xi} = v_f, \\ \theta_N^{\xi} = \theta_f, \; \omega_N^{\xi} = \omega_f, \; \delta_N = 0 \tag{31}$$

where m_{dry} is the dry mass of the vehicle, and r_f, v_f, θ_f, and ω_f are the terminal values for the position, velocity, body tilt angle, and angular velocity, respectively. The gimbal angle at the final node is constrained to be zero to avoid plume-impingement on the retrieval structure.

6.6 The discrete SeCO subproblem

We define $J(x_N)$ in Equation (15a) to be $-m_N$, i.e., the final mass of the vehicle is maximized (thus minimizing propellant consumption). The discrete SeCO subproblem, which is a second-order cone program (SOCP), can now be given as follows:

min   *Objective function*: Equation (15a)

s.t.   *Common constraints*, Subsection 6.1:
       Equations (15b), (21), and (22)
     Coast phase, Subsection 6.2:
       Equations (23)
     High-thrust burn phase, Subsection 6.3:
       Equations (24), (25), and (26)
     Low-thrust burn phase, Subsection 6.4:
       Equations (27) and (28)
     Terminal descent phase, Subsection 6.5:
       Equations (29), (30), and (31)

7. NUMERICAL RESULTS

For our numerical implementation of the multi-phase rocket landing guidance algorithm, we choose a grid of $N = 16$ discrete temporal nodes, with one node allocated to the unpowered coast phase, and 5 nodes allocated to each of the remaining three phases of flight, i.e, $k_{\text{ignition}} = 2$, $k_{\text{switch}} = 7$, $k_{\text{trigger}} = 12$. The following parameters are used, where the ones pertaining to the vehicle/maneuver were either estimated or obtained from public sources (DeSisto, 2021; Sagliano et al., 2021; Malyuta et al., 2022).

$$g_0 = 9.81\,\text{m s}^{-2}, \; I_{\text{sp}} = 330\,\text{s}, \; l_r = 4.5\,\text{m}, \; l_h = 50\,\text{m},$$
$$l_{\text{cm}} = 0.4\,l_h, \; l_{\text{cp}}^0 = 0\,\text{m}, \; l_{\text{cp}}^1 = 0.2\,l_h, \; \rho_{\text{air}} = 1.225\,\text{kg m}^{-3},$$
$$S_{\text{area}} = 545\,\text{m}^2, \; v_{\text{terminal}} = 85\,\text{m s}^{-1}, \; m_i = 100000\,\text{kg},$$
$$c_{\hat{x}} = 0.0522, \; c_{\hat{z}} = 0.4068, \; T_{\max} = 2200\,\text{kN}, \; T_{\min} = 880\,\text{kN},$$
$$\delta_{\max} = 10°, \; \dot{\delta}_{\max} = 15°\text{s}^{-1}, \; h_{\text{trigger}} = 100\,\text{m}, \; \gamma_{\text{gs}} = 5°,$$
$$v_{\max} = 20\,\text{m s}^{-1}, \; \theta_{\max} = 5°, \; \omega_{\max} = 2.5°\text{s}^{-1}, \; \delta_{\max_{\text{TD}}} = 1°,$$
$$m_{\text{dry}} = 85000\,\text{kg}, \; r_i = (1000, 100)\,\text{m}, \; v_i = (-90, 0)\,\text{m s}^{-1},$$
$$\theta_i = 90°, \; \omega_i = 0°\text{s}^{-1}, \; r_f = (0, 0)\,\text{m}, \; v_f = (0, 0)\,\text{m s}^{-1},$$
$$\theta_f = 0°, \; \omega_f = 0°\text{s}^{-1}, \; s_{\min} = 0.6\,\text{s}, \; s_{\max} = 10\,\text{s}$$

Averaged over 100 full SeCO solves, we report a mean run-time of the PIPG solver (to solve the entire nonconvex problem) of 13.7 ms. In comparison, ECOS[1] requires 37.1 ms, on average, to solve the problem. For the comparison, we assess the quality of the converged solutions in terms of the difference in propellant consumption (0.02%) and the number of SeCO iterations required to converge (7). A real-time guidance solution obtained via PIPG is shown

---

[1] For this problem, ECOS is faster than both MOSEK and GUROBI.

in Figure 2. Here, we observe that the optimizer chooses to initiate the powered-descent phase at an altitude of 490.34 m and a speed of 86.28 ms^{-1} (which is very close to the terminal velocity). Further, we note that the numerous prototype flight tests and independent analyses corroborate many of our observations (DeSisto, 2021).

8. CONCLUSIONS

We introduce SeCO, a novel matrix-inverse-free paradigm for solving nonconvex optimal control problems in real-time, and solve a multi-phase rocket landing guidance problem with free-transition-time and convex state-triggered constraints. These solutions are computed using PIPG, which is almost three times faster than ECOS, a state-of-the-art convex optimization solver.

REFERENCES

Açıkmeşe, B. and Ploen, S.R. (2007). Convex programming approach to powered descent guidance for Mars landing. *AIAA Journal of Guidance, Control and Dynamics*, 30(5), 1353–1366.

Antsaklis, P.J. and Michel, A.N. (2006). *Linear systems*. Basel, Switzerland: Birkhauser.

Atkinson, N. (2022). SpaceX tests its Starship-catching launch tower. www.universetoday.com/spacex-tests-its-starship-catching-launch-tower. Accessed: 11.18.2022.

Berkovitz, L.D. (2013). *Optimal control theory*, volume 12. Springer Science & Business Media.

Bhasin, N.S. (2016). *Fuel-optimal spacecraft guidance for landing in planetary pits*. Master's thesis, Carnegie Mellon University, Pittsburgh, PA.

Blackmore, L. (2016). Autonomous precision landing of space rockets. In *in Frontiers of Engineering: Reports on Leading-Edge Engineering from the 2016 Symposium*, volume 46, 15–20.

DeSisto, A. (2021). Starship and its belly flop maneuver. www.everydayastronaut.com/starships-belly-flop-maneuver. Accessed: 10.14.2022.

Garrido, J.V. and Sagliano, M. (2021). Ascent and descent guidance of multistage rockets via pseudospectral methods. In *AIAA SciTech 2021 Forum*.

Hwang, J. and Ahn, J. (2022). Integrated optimal guidance for reentry and landing of a rocket using multi-phase pseudo-spectral convex optimization. *International Journal of Aeronautical and Space Sciences*, 1–9.

Kamath, A.G., Elango, P., Kim, T., Mceowen, S., Yu, Y., Carson III, J.M., Mesbahi, M., and Açıkmeşe, B. (2023). Customized real-time first-order methods for onboard dual quaternion-based 6-DoF powered-descent guidance. In *AIAA SciTech 2023 Forum*.

Lee, S.D. and Lee, C.H. (2022). Multi-phase and dual aero/propulsive rocket landing guidance using successive convex programming. *Proceedings of the Institution of Mechanical Engineers, Journal of Aerospace Engineering*.

Ma, L., Wang, K., Shao, Z., Song, Z., and Biegler, L.T. (2019). Direct trajectory optimization framework for vertical takeoff and vertical landing reusable rockets: case study of two-stage rockets. *Engineering Optimization*.

Malyuta, D., Reynolds, T., Szmuk, M., Mesbahi, M., Açıkmeşe, B., and Carson, J.M. (2019). Discretization performance and accuracy analysis for the rocket powered descent guidance problem. In *AIAA Scitech 2019 Forum*.

Malyuta, D., Reynolds, T.P., Szmuk, M., Lew, T., Bonalli, R., Pavone, M., and Açıkmeşe, B. (2022). Convex optimization for trajectory generation: A tutorial on generating dynamically feasible trajectories reliably and efficiently. *IEEE Control Systems*, 42(5), 40–113.

Mao, Y., Szmuk, M., and Açıkmeşe, B. (2016). Successive convexification of non-convex optimal control problems and its convergence properties. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, 3636–3641.

Mceowen, S. and Açıkmeşe, B. (2022). Hypersonic entry trajectory optimization via successive convexification with abstracted control. In *AIAA SciTech 2022 Forum*.

Reynolds, T., Malyuta, D., Mesbahi, M., Açıkmeşe, B., and Carson, J.M. (2020a). A real-time algorithm for non-convex powered descent guidance. In *AIAA Scitech 2020 Forum*.

Reynolds, T.P., Szmuk, M., Malyuta, D., Mesbahi, M., Açıkmeşe, B., and Carson III, J.M. (2020b). Dual quaternion-based powered descent guidance with state-triggered constraints. *Journal of Guidance, Control, and Dynamics*.

Sagliano, M., Seelbinder, D., and Theil, S. (2021). SPARTAN: Rapid trajectory analysis via pseudospectral methods. In *ICATT*.

Shotwell, G.E. and Blackmore, L. (2021). Space launch in 50 years: Abundance at last? *National Academy of Engineering*.

Szmuk, M., Reynolds, T., Açıkmeşe, B., Mesbahi, M., and Carson, J.M. (2019). Successive convexification for 6-DoF powered descent guidance with compound state-triggered constraints. In *AIAA Scitech 2019 Forum*.

Szmuk, M., Reynolds, T.P., and Açıkmeşe, B. (2020). Successive convexification for real-time six-degree-of-freedom powered descent guidance with state-triggered constraints. *Journal of Guidance, Control, and Dynamics*, 43(8), 1399–1413.

Yu, Y., Elango, P., and Açıkmeşe, B. (2020). Proportional-integral projected gradient method for model predictive control. *IEEE Control Systems Letters*, 5(6), 2174–2179.

Yu, Y., Elango, P., Açıkmeşe, B., and Topcu, U. (2022a). Extrapolated proportional-integral projected gradient method for conic optimization. *IEEE Control Systems Letters*, 7, 73–78.

Yu, Y., Elango, P., Topcu, U., and Açıkmeşe, B. (2022b). Proportional–integral projected gradient method for conic optimization. *Automatica*, 142, 110359.

Zhang, D. and Zhang, Y. (2022). PySCP: A multiple-phase optimal control software using sequential convex programming. *International Journal of Aerospace Engineering*, 2022.