

# Ignition Point Reachability for Aerodynamically-Controlled Reusable Launch Vehicles

Benjamin Chung <sup>\*a</sup>, Kazuya Echigo <sup>†b</sup>, and Behçet Açıkmeşe <sup>‡b</sup>

<sup>a</sup>*JuliaHub, Inc*

<sup>b</sup>*William E. Boeing Department of Aeronautics and Astronautics, University of Washington, Seattle, WA 98195, USA.*

We describe a successive convex programming (Sequential Convex Programming (SCP)) based approach for estimate the set of points where a 5-degree of freedom (5-DoF) reusable launch vehicle (RLV) returning to a landing site can transition from aerodynamic to propulsive descent. Determining the set of feasible ignition points that a RLV can use and then safely land is important for mission planning and range safety. However, past trajectory optimization approaches for RLVs consider substantially simplified versions of the vehicle dynamics. Furthermore, prior reachability analysis methods either do not extend to the full constraint set needed for an RLV or are too beset by the curse of dimensionality to handle the full 5-DoF dynamics.

To solve this problem, we describe an algorithm that approximates the projection of a high dimensional reachable set onto a low dimensional space. Instead of computing all parts of the reachable space, we only calculate reachability in the projected space of interest by using repeated trajectory optimization to sample the reachable polytope in the reduced space. The optimization can take into account initial and terminal constraints as well as state and control constraints.

We show that our algorithm is able to compute the projection of a reachable set into a low dimensional space by calculating the feasible ignition points for a two-phase aerodynamic/propulsive RLV landing trajectory, while also demonstrating the aerodynamic divert enabled by our body and fin actuator model.

## I. Introduction

Returning to land is an important part of the lifecycle of a reusable launch vehicle. Such return trajectories include both aerodynamic and propulsive descent guidance phases due to minimum throttle and relight count limitations. Most reusable launch vehicles have minimum thrust-to-weight ratio greater than 1 on touchdown, placing substantial pressure on trajectory generation.

As part of returning to a launch site a reusable vehicle must first reach a state from which it may propulsively land: too far away and it runs out of fuel before it touches down, too close and it hits the ground at substantial velocity. Moreover, range safety requirements frequently require that an aero-ballistic returning stage not be aimed at the landing site until it has successfully relit its engine and transitioned from aerodynamic to propulsive guidance [1]. Consequently, a returning booster must maneuver aerodynamically to put its ignition point into a safe location and yet still be able to reach the desired landing site under power.

The next question that arises is: given an initial state and a landing site where can the divert maneuver be performed? The returning stage can maneuver aerodynamically to reach the point from which it must be able to land without running out of fuel or shutting its engine down again as illustrated in Figure 1. If we can compute such a reachable volume we can then choose where in the volume reignition occurs and powered descent guidance begins based on range safety and other concerns.

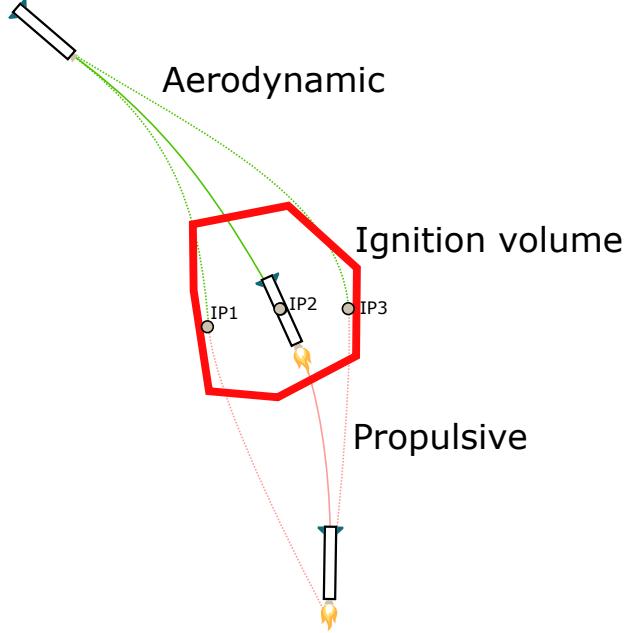
Trajectory optimization for low-fidelity 6-degree of freedom (6-DoF) reusable launch vehicle models has remained immature, with either simplistic drag-only aerodynamic models [2] that are unsuitable for modeling aerodynamic maneuvers or reduced models with limited aerodynamic control modeling [3].

Reachability analysis adds a further challenge to the problem. Superficially, we wish to compute a 11-dimensional reachability analysis for a system subject to initial, final, and intermediate state and control constraints. This is untractable for most reachability techniques. For example, Hamilton-Jacobi methods are all beset by the curse of dimensionality [4]

<sup>\*</sup>Modeling & Simulation Engineer, JuliaHub, Inc. Corresponding author email: benjamin.chung@juliahub.com

<sup>†</sup>Doctoral Student, William E. Boeing Department of Aeronautics and Astronautics, University of Washington.

<sup>‡</sup>AIAA Fellow. Professor, William E. Boeing Department of Aeronautics and Astronautics, University of Washington.



**Fig. 1** Schematic representation of two-phase aerodynamic descent guidance.

due to having to build and propagate representations of past reachable states while direct set computation approaches [5] cannot guarantee satisfaction of future state constraints. Similarly, sampling-based methods [6] are limited in their ability to handle state and control constraints.

Reachability of all dimensions and all time is intractable for our problem. However, we only care about a subset of the dimensions—the rocket’s position—at a specific time: ignition. Accordingly, we are only truly interested in a 3-dimensional reachability problem for a single point. Reachability for our problem can therefore be reduced to identifying the volume in 3-dimensional space which is somehow reachable from the initial and final conditions in 11-dimensional space. We can do this using trajectory optimization.

We propose a trajectory optimization approach based on Sequential Convex Programming (SCP) for identifying the reachable set of ignition points for an aerodynamically-controlled launch vehicle. Previous work on trajectory optimization based reachability required a convex dynamics [7] or only considered landing site reachability for the simpler exoatmospheric lander dynamics [8]. By combining a new formulation of nonlinear axisymmetric body aerodynamics for better problem space exploration with trajectory optimization based sampling we can approximate reachable intermediate points even with highly nonlinear dynamics.

## II. Problem Description

We aim to approximate the reachable volume for an intermediate point on a trajectory that satisfies initial and terminal conditions. In this section, we will describe the continuous-time nonlinear powered landing problem beginning with the vehicle dynamics, followed by the continuous-time objective, boundary conditions, and state and control constraints.

We are focused on the two-phase powered descent guidance problem, where an aerodynamically and propulsively actuated rotationally axisymmetric vehicle undergoes two phases:

- 1) Aerodynamic flight where the engine is shut off, and
- 2) Propulsive flight after engine ignition.

The time spent in each phase may vary so long as the boundary conditions and path constraints are satisfied. We wish to identify the volume in which the transition can occur while satisfying all other requirements.

Our model uses three reference frames:

- 1)  $F_I$ , a surface fixed (rotating around the center of the planet) North-East-Up reference frame with the origin at the landing site.
- 2)  $F_B$ , a vehicle-fixed frame centered at the vehicle center-of-mass, with its Z axis pointing out the nose of the

vehicle and the X and Y axes chosen to be jointly perpendicular to Z and aligning with the axes in  $F_I$  if the vehicle is unrotated.

3)  $F_W$ , a vehicle-fixed frame centered at the vehicle center-of-mass, with its Z axis pointing opposite the air-relative velocity vector and the X and Y axes constructed from  $F_I$  by the minimum rotation around  $F_I$ 's N/E axes such that the new Z axis is opposite the velocity vector.

We note the frame that a variable inhabits with a subscript; for example,  $r_I(t)$  is the vehicle's position in the inertial frame. The construction of  $F_B$  and  $F_W$  will be covered in more detail later in this section.

### A. Vehicle Model

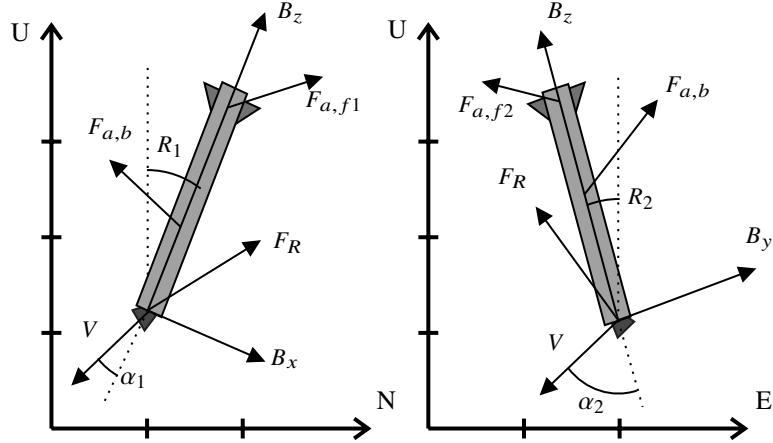


Fig. 2 Vehicle dynamics

The vehicle is modeled as a rigid body. It possesses the states:

- Mass:  $m(t) \in \mathbb{R}$
- Position:  $r_I(t) \in \mathbb{R}^3$
- Velocity:  $v_I(t) \in \mathbb{R}^3$
- Body-to-inertial rotation:  $R_{B/I}(t) \in \mathbb{R}^2$
- Angular rate:  $\omega_B(t) \in \mathbb{R}^2$

Since our vehicle is axisymmetric we only allow it to rotate along the north and east axes in the inertial frame; the rotation about the body up axis is fixed. Accordingly, our model is 5-degree of freedom (5-DoF) rather than 6-DoF. Consequently, we can use a simple pair of Euler angles to represent the vehicle's attitude. In order to reduce the number of needed discretization nodes we also apply a damping coefficient  $C_{\text{damp}}$  to its attitude dynamics that approximates the affects of  $C_{m,q}$  and  $C_{m,\dot{\alpha}}$  though the same effect could be achieved with a feedback controller.

Denoting the forces applied to the vehicle as  $F_I(t)$  and torques  $M_B(t)$ , our basic equations of motion then follow as:

$$\dot{m}(t) = \frac{\|u_{B,R}(t)\|_2}{gI_{sp}} \quad (1)$$

$$\dot{r}_I(t) = v_I(t) \quad (2)$$

$$\dot{v}_I(t) = \frac{F_I(t)}{m(t)} + A_{I,g}(t) + A_{I,c}(t) + A_{I,r}(t) \quad (3)$$

$$F_I(t) = T_{B/I}(R_{B/I}(t))(u_{B,R}(t)) + F_{I,fins}(t) + F_{I,body}(t) \quad (4)$$

$$\dot{R}_{B/I}(t) = \omega_B(t) \quad (5)$$

$$\omega_B(t) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} (J_B^{-1}(t) \times M_B(t)) - C_{\text{damp}}\omega_B(t) \quad (6)$$

$$M_B(t) = r_{B,engine} \times u_B(t) + T_{I/B}(R_{B/I})M_{I,body}(t) + M_{B,fins}(t) \quad (7)$$

$$J_B(t) = J_{B,dry} + \frac{m(t) - m_{dry}}{m_{wet} - m_{dry}} (J_{B,wet} - J_{B,dry}) \quad (8)$$

where  $\times$  is the cross product. We denote the transformation matrix from frame  $B$  to  $I$  as  $T_{B/I}(R_{B/I}(t))$ . Reusable launch vehicles tend to have a large fuel fraction relative to their structural mass and therefore their moment of inertia changes meaningfully with fuel depletion. We approximate the dependence with a linear interpolation between the wet and dry moments of inertia, as depicted in Eq. (8).

We have two types of control:

- Aerodynamic, represented as a force  $u_{W,a}(t)$  applied to the vehicle at a fixed position relative to its center of mass, and
- Propulsive, represented as a force  $u_{B,R}(t)$  applied by the rocket at an offset  $e_B$  from the vehicle's center of mass. Propulsive control is dispatched with easily through inclusion in the mass depletion dynamics Eq. (1), in the translational dynamics Eq. 4 as transformed by the the body to inertial rotation, and by inclusion in the body rotational moment Eq. (7). Aerodynamic controls will be discussed later.

**Environmental Modeling** We adopt the international standard atmosphere (ISA) [9] as our atmospheric model, which is used to determine the local atmospheric density  $\rho(r_I)$ , temperature, and speed of sound  $c(r_I)$ . Aerodynamic data is then normalized to the reference pressure  $\rho_0$  which we define as sea level pressure and then scaled in the simulation by  $\rho_r = \rho(r_I/\rho_0)$ . Similarly, mach number is computed as  $\|v_I\|_2/c(r_I)$ . We assume that the surface-level temperature is unchanged from the ISA reference value.

We use a spherical gravity model  $A_{I,g}(t) = \frac{\mu}{\|r_I(t) - r_{I,ls}\|_2^3} (r_I(t) - r_{I,ls})$  where  $r_{I,ls}$  is the location of the center of the planet as expressed in the inertial reference frame. Furthermore, the planet (and thus our reference frame) are rotating with angular velocity  $\omega_{I,p}$  inducing Coriolis  $A_{I,c} = 2\omega_{I,p} \times v_I(t)$  and centrifugal accelerations  $A_{I,r} = \omega_{I,p} \times (\omega_{I,p} \times (r_I(t) - r_{I,ls}))$ .

**Body Aerodynamics** Aerodynamic modeling of axisymmetric bodies for optimization is challenging due to a singularity in the directions of lift and moment when the body is parallel to the velocity vector. Wings or other lifting devices would establish directions from which we define lift and slip axes but axisymmetric bodies offer no such affordances.

We define drag to lie parallel to the flight path vector and lift to be pointed away from the flight path in the direction that the vehicle is tilted, thereby setting sideslip to be zero. Non-axisymmetric vehicles need to roll to maintain an explicit zero-sideslip orientation [10] as their rotation around their forward vector changes the applied aerodynamic forces. However, our vehicle has no aerodynamic dependence on roll angle since it is axisymmetric.

The aerodynamic moment on a cylinder will be applied on an axis perpendicular to the body up and flight path vectors or  $\hat{d}_{I,m}(t) = \frac{1}{\|v_I(t)\|_2 \sin(\alpha(t))} \hat{b}_{I,z}(t) \times v_I(t)$ . Lift is applied perpendicular to the flight path vector and the aerodynamic moment vector or  $\hat{d}_{I,L}(t) = \frac{1}{\|v_I(t)\|_2} v_I(t) \times \hat{d}_{I,m}(t)$ . The flight path direction  $\hat{v}_I(t)$ , the moment direction  $\hat{d}_{I,M}$ , and the lift direction  $\hat{d}_{I,L}$  now forms an orthogonal basis.

Accordingly, basic equations for aerodynamic lift, drag, and moment are

$$L_I(t) = \frac{1}{2} \rho(t) C_l(\alpha(t), M(t)) \|v_I(t)\|_2^2 \hat{d}_{I,L}(t) \quad (9)$$

$$D_I(t) = -\frac{1}{2} \rho(t) C_d(\alpha(t), M(t)) \|v_I(t)\|_2 v_I(t) \quad (10)$$

$$M_I(t) = \frac{1}{2} \rho(t) C_m(\alpha(t), M(t)) \|v_I(t)\|_2^2 \hat{d}_{I,m}(t) \quad (11)$$

$$F_{I,body}(t) = L_I(t) + D_I(t). \quad (12)$$

This version of drag has no singularities and can thus readily be used for optimization. Observe, however, that the computation of  $\hat{d}_{I,m}(t)$  normalizes the cross product  $\hat{b}_{I,z} \times v_I(t)$  by dividing by the angle of attack  $\alpha(t)$ . Accordingly, a singularity exists at 0 degrees angle of attack when there is no tilt angle to reference our frame to. This is undesirable as the reusable launch vehicle must land at a very low angle of attack.

We simplify the lift formulation to solve this problem with two observations:

- 1) At  $\alpha(t) = 0$  the value of  $C_l(\alpha(t), M(t))$  is not relevant since the body-up vector  $\hat{b}_{I,z}$  is parallel to  $v_I(t)$  and thus  $\|v_I(t) \times (\hat{b}_{I,z}(t) \times v_I(t))\|_2 = 0$ .
- 2) Lift of axisymmetric bodies typically has a nearly linear relationship to  $\alpha$  at low angles of attack.

These properties can be exploited to smooth out the singularity by factoring  $\frac{1}{\sin(\alpha)} = \csc(\alpha)$  into the lookup table for  $C_l$  to form the lookup table  $C_{l,mod}$ .

$$C_{l,mod}(\alpha, M) = \begin{cases} C_l(\alpha, M) \csc(\alpha) & \alpha \neq 0 \\ 0 & \alpha = 0 \end{cases} \quad (13)$$

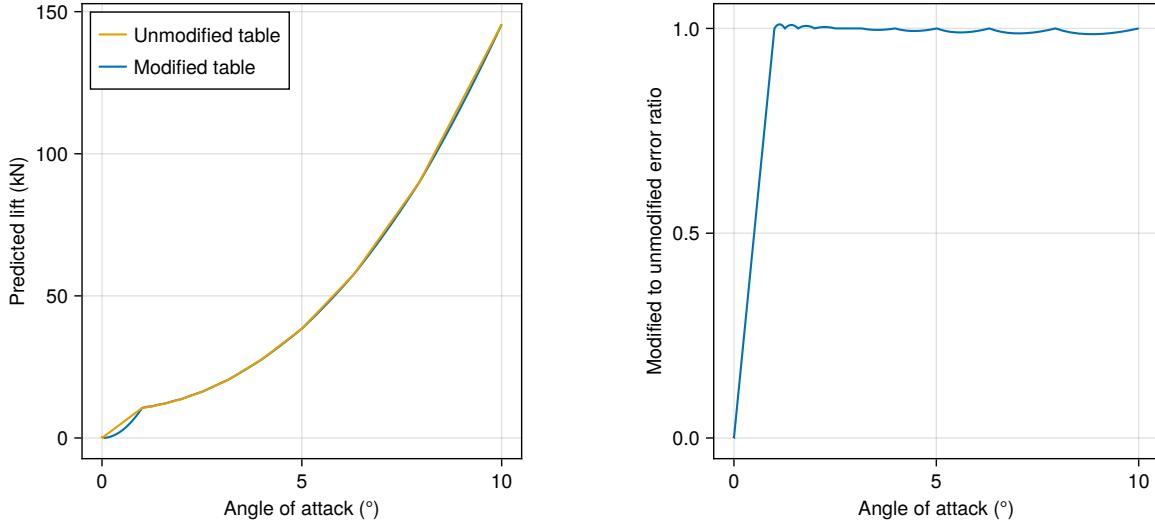
For nonzero angles of attack, we premultiply the value in the lookup table by  $\csc(\alpha)$ , while at 0 angle of attack we simply let the lookup table value be 0. The linearity of lift and torque at small  $\alpha$  causes error to remain small between 0 and the first nonzero data point, while no lift or torque is produced at 0 angle of attack.

Including the normalization  $\csc(\alpha)$  in the lookup table then lets us define body lift and moment as follows:

$$L_I(t) = \frac{1}{2} \rho(t) C_{l,mod}(\alpha(t), M(t)) v_I(t) \times (\hat{b}_{I,z}(t) \times v_I(t)) \quad (14)$$

$$M_{I,body} = \frac{1}{2} \rho(t) C_{m,mod}(\alpha(t), M(t)) \|v_I(t)\|_2 (\hat{b}_{I,z}(t) \times v_I(t)). \quad (15)$$

The velocity normalization factors  $\|v_I(t)\|_2^2$  are factored singly and doubly into the cross products for torque and lift, respectively, and including  $\csc(\alpha(t))$  in the lookup table  $C_{m,mod}(\alpha(t), m)$  allows us to use the cross product magnitude to calculate lift and drag directly.



**Fig. 3 Modified vs. unmodified lift predictions at mach 0.9.** **Fig. 4 Relative lift force error vs. original lookup table at mach 0.9.**

Figures 3 and 4 show the relative and absolute force error as a function of angle of attack at mach 0.9. We evaluated the reference medium-fidelity lift and drag models for the vehicle in a fixed 0 degree fin actuation down to 1 deg angle of attack. Out to the limit of the underlying look up table, the modified lookup table maintains good agreement (< 2%) to the linear interpolation of the original look up table. However, below the minimum available angle of attack the approximation loses validity. Absolute error below the end of the lookup table is small in absolute terms and proportional to the force at the cutoff minimum nonzero angle of attack in the reference lookup table.

**Actuator Aerodynamics** Most reusable launch vehicles use four fins in a cross pattern at the top of the first stage for aerodynamic control. For simplicity, we consider a vehicle with fins located 90 deg around the circumference of the vehicle; skewed configurations can be analyzed with the same framework but with different force vectors, as discussed later.

Ultimately, we wish to abstract over exactly how fins produce lift and drag. To reduce the dimensionality of the control vector, the aerodynamic control vector is expressed as a force applied to the vehicle on a plane perpendicular to

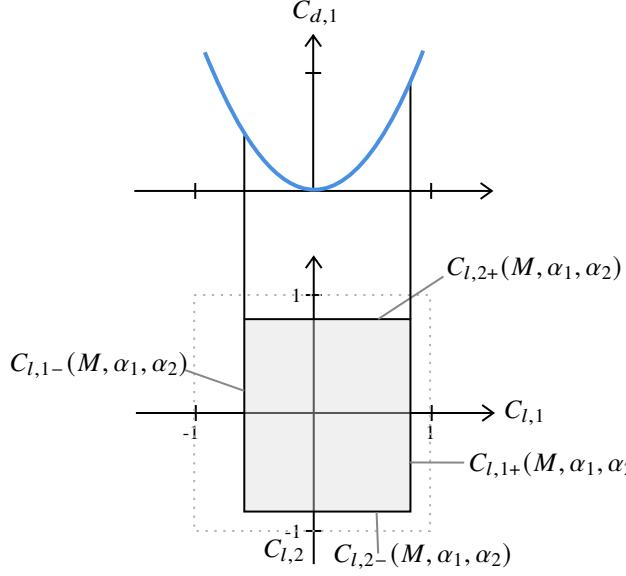


Fig. 5 Fin control constraints and drag polar coupling.

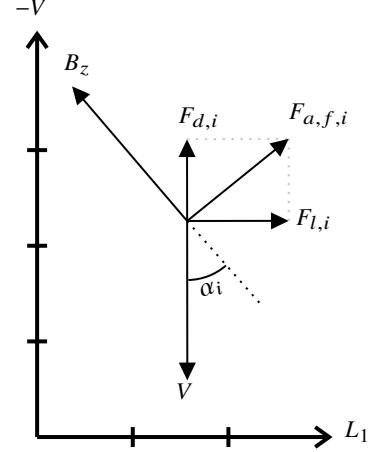


Fig. 6 Fin force frame.

the velocity vector, and is thus two dimensional. This lift produced by the fins then incurs an induced drag on the vehicle. Furthermore, for any given attitude and mach number the fins can only produce so much lift before they begin stalling.

Our aerodynamic actuators apply a force calculated in the wind frame; our control is a vector in the plane perpendicular to the velocity vector that then induces a force opposite the velocity vector (drag). The geometry in the plane of a single fin pair is shown in Figure 6. Aerodynamic lift is scheduled through a table-derived scale factor from a normalized -1 to 1 input on each axis. The scale factor is determined by lookup table as a function of the current mach number and per-plane angles of attack  $C_{fin,l,i}(M(t), \alpha_1(t), \alpha_2(t))$  that is normalized by the squared velocity. The angles of

To account for state-dependent lift control asymmetries like actuator rotation limits and critical angles of attack we apply an additional table-derived nonlinear box constraint on the control as shown in Figure 5. For each condition the normalized control  $u_a$  must satisfy the constraint

$$C_{l,i-}(M, \alpha_1, \alpha_2) \leq u_{a,i} \leq C_{l,i+}(M, \alpha_1, \alpha_2) \text{ for } i \in 1, 2. \quad (16)$$

Induced drag is then coupled to lift through a table-derived drag polar as a function of mach. We observe that for a given mach and AoA pair induced drag is proportional to squared lift plus a sweep offset that is linear in the cosine of the opposing angle of attack and the lift squared.

Accordingly, our raw aerodynamic control dynamics in the wind frame are

$$f_{f,l,i}(t) = u_{a,i}(t) \|v(t)\|_2^2 C_{fin,l,i}(M(t), \alpha_1(t), \alpha_2(t)) \quad (17)$$

$$f_{f,d}(t) = C_{fin,d,lin}(M(t)) \begin{bmatrix} \cos(\alpha_2(t)) & \cos(\alpha_1(t)) \end{bmatrix} \begin{bmatrix} f_{f,l,1}^2(t) \\ f_{f,l,2}^2(t) \end{bmatrix} + C_{fin,d,cst}(M(t)) f_{f,l}^T(t) f_{f,l}(t) \quad (18)$$

$$F_{W,f}(t) = \begin{bmatrix} f_{f,l,1}(t) \\ f_{f,l,2}(t) \\ f_{f,d}(t) \end{bmatrix}. \quad (19)$$

The actuator forces must then be mapped from the wind frame to the inertial frame. The drag vector is easy: it merely points reverse of the velocity vector. The lift vectors are harder, as their clock angle around the velocity vector is determined by the vehicle roll angle. In our problem we fix the vehicle roll angle to align the body axes with the axes of the inertial frame, so the actuator lift basis vectors are a function only of the velocity.

We orient the lift vectors by constructing a reference frame whose Z axis is aligned with the velocity vector. Our construction takes the unrotated body basis vectors (which are initially expressed in the NED inertial reference frame) and then rotates them so that the Z axis points through the velocity vector. We construct the rotation matrix  $R_{W/I}$  using Möller and Hughes's construction [11] that rotates  $-\hat{Z}$  around the  $-Z \times \hat{v}$  axis by the smallest angle such that  $-R_{W/I}\hat{Z}$  is aligned with  $\hat{v}$ . This construction avoids any normalization beyond the computation of  $\hat{v}$ , thereby limiting the singularities to those that occur at zero velocity and at 180 degrees angle of attack.  $R_{W/I}$  then relates the forces calculated in the unrotated inertial axis aligned body frame to the forces applied in the inertial frame from the wind frame.

The moment from the actuators is calculated trivially as the cross product of the force and the (assumed to be fixed) mean position of the actuators relative to the body center of mass. The net body-frame contribution of the aerodynamic actuators to force and moment is therefore:

$$F_{I,fins}(t) = R_{W/I}(t)f_{f,l}(t) - v_I(t) * f_{f,d}(t) \quad (20)$$

$$M_{B,fins}(t) = \sum_i r_{B,fins} \times R_{I/B}(t)F_{I,fins}(t). \quad (21)$$

Where  $r_{B,fins}$  is the mean position of the fins in the body frame.

## B. Dynamic constraints

We enforce initial and final position, velocity, orientation, and angular rate constraints:

$$r_I(0) = r_{I,i}, v_I(0) = v_{I,i}, R_{B/I}(0) = R_{B/I,i}, \omega_B(0) = \omega_{B,i}, m(0) = m_i \quad (22)$$

$$r_I(t_f) = r_{I,f}, v_I(t_f) = v_{I,f}, R_{B/I}(t_f) = R_{B/I,f}, \omega_B(t_f) = \omega_{B,f} \quad (23)$$

We similarly enforce a mass lower bound:

$$m(t) \geq m_{dry} \quad (24)$$

The thrust vector is constrained to be within the gimbal range of the engine  $\theta$

$$\left\| \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} u_{B,R}(t) \right\|_2 \leq \tan(\theta) u_{B,R}(t)^T \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (25)$$

less than the maximum thrust,

$$\|u_{B,R}(t)\|_2^2 \leq u_{\max} \quad (26)$$

and greater than the minimum thrust for the propulsive trajectory.

$$u_{\min} \leq \|u_{B,R}(t)\|_2^2 \quad (27)$$

We impose a glideslope constraint of angle  $\gamma$  to the propulsive component trajectory.

$$\left\| \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} r_I(t) \right\|_2 \leq \tan(\gamma) r_I(t)^T \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (28)$$

We enforce a nonlinear net angle of attack with rolloff constraint:

$$\tanh\left(\frac{\|v_I(t)\|_2}{v_{\text{small}}}\right) \alpha(t) \leq \alpha_{\max} \quad (29)$$

to ensure lookup table validity. We use a hyperbolic rolloff at velocities below  $v_{\text{small}}$  as the calculation of  $\alpha$  has a singularity at  $\|v\|_2 = 0$ . As a result, numerical error near landing can therefore create very large yet practically meaningless values of  $\alpha$ .

Similarly, we enforce a maximum angular rate constraint

$$\|\omega_B(t)\|_2 \leq \omega_{\max} \quad (30)$$

to limit centrifugal loads on the vehicle's structure.

Additionally, we impose maximum dynamic pressure and maximum dynamic pressure-angle of attack product limits:

$$\frac{1}{2}\rho(t)||v_I(t)||_2^2 \leq q_{\max} \quad (31)$$

$$\frac{1}{2}\rho(t)\alpha(t)||v_I(t)||_2^2 \leq \chi_{\max}. \quad (32)$$

### III. Methods

We aim to identify the reachable volume for the transition point between unpowered aerodynamic flight and powered descent guidance for a reusable launch vehicle. The 5-DoF dynamics and constraints we have described are highly nonlinear and the system has excessive dimensionality for traditional state-space exploration methods such as Hamilton-Jacobi methods while the imposed constraints make it unsuitable for set propagation or sampling-based methods.

The key observation is that we do not care about nonposition state at the ignition points so long as all dynamic constraints remain satisfied. We can thus avoid the curse of dimensionality if we can avoid examining vehicle attitude as part of our reachability analysis. We do so using a successive convexification-based trajectory optimization approach to "fill in" for the unconsidered states.

Unlike trajectory optimization based approaches for convex problems [12], we cannot provide a guarantee that our computed sets over or underapproximates the reachable space. The reachable set for such a nonlinear problem may exhibit "holes," concavities, consist of several disconnected sets, or have protuberances. Our algorithm cannot certify that the produced volume is either a super or subset of the true reachable space. What the approach does guarantee is that all of the explored points are actually reachable; we can always identify a feasible trajectory through a given previously identified point, but cannot always do so for interpolations thereof.

We first describe our method for solving the trajectory optimization subproblems, then examine our approach for using them to approximate the reachable volume.

**Trajectory Optimization** Our basic trajectory optimization subproblem is

Problem 1: Reference nonlinear trajectory optimization problem

$$\begin{aligned} \min_{x,u} \quad & L(x, u) \\ \text{s.t.} \quad & \text{Eq. (1), (2), (3), (5), (6), (16), (22), (23), (24),} \\ & (25), (26), (27), (28), (29), (30), (31), (32) . \end{aligned} \quad (\mathbf{P1})$$

Where  $L(x, u)$  is a nonlinear objective function of the state and control trajectories.

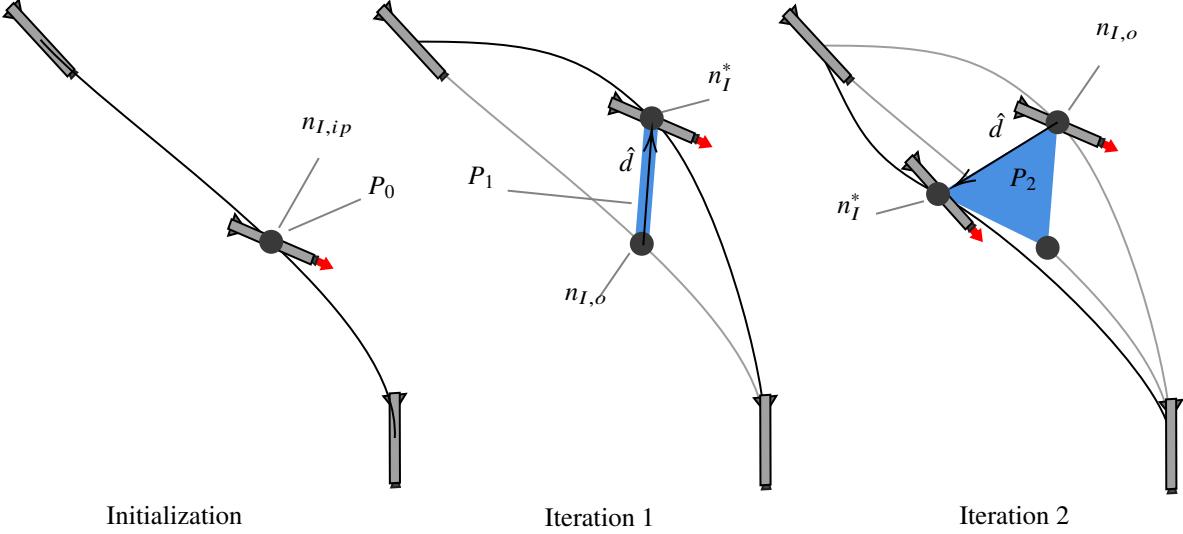
We perform trajectory optimization by transcribing a nondimensionalized continuous-time model in Modeling-Toolkit.jl [13] into a discrete transition system via multiple shooting which is then solved using the prox-linear method [14] with successive convex optimization. Our implementation adopts the penalized trust region method using the exact  $L_1$  penalty [15] with the adaptive weighting scheme described by Mao et al [16]. More details on the algorithm can be found in [17].

Our multiple shooting method is the same as that used by Dukeman [10] or Szmuk [2]. The primary distinction is that we use two different time dilation parameters  $\tau_a$  and  $\tau_p$  (for aerodynamic and propulsive, respectively) to "stretch" nondimensionalized time  $\tau$  into physical time  $t$ . Varying  $\tau_a$  and  $\tau_p$  then varies the overall flight time and the flight time of the segments as follows:

$$t = \begin{cases} \tau_a \tau & : \tau < \frac{1}{2} \\ \tau_p(\tau - \frac{1}{2}) + \frac{1}{2}\tau_a & : \tau \geq \frac{1}{2} \end{cases}$$

The total time of flight is then  $\frac{1}{2}\tau_a + \frac{1}{2}\tau_p$ .

Aerodynamic controls  $u_{W,a}(t)$  and propulsive controls  $u_{B,R}(t)$  are transcribed using first-order-hold. Propulsive control is only applied when  $\tau > \frac{1}{2}$ ; otherwise  $u_{B,R}(\tau) = 0$  where  $\tau < \frac{1}{2}$ .



**Fig. 7 Schematic of reachability algorithm iteration**

Constraints are either enforced nodally as part of the convex optimization problem (22), (23), (24), (25), (26), (28), (30) or through continuous time constraint satisfaction [14] (16), (27), (29), (30), (32), (31). We transcribe Eq. (30) both as a second order cone constraint on each node to improve convergence of the low frequency dynamics and with continuous time constraint satisfaction (CTCS) to reduce inter-sample constraint violation caused by the high frequency dynamics. Additionally, while Eq. 16 is a control constraint, it strongly depends on rapidly changing vehicle states so we therefore transcribe it using CTCS. Unlike Elago et al [14], we convert each CTCS constraint as an independent state variable in our augmented dynamics to improve the gradients of the sensitivity problems.

Most of the computation for the prox-linear method is incurred in sensitivity analysis of the integrated multiple shooting subproblems. We compute the Jacobians for each of the subproblems in parallel using SciMLSensitivity [18] optimized for sparsity using SparseConnectivityTracer [19] which are then solved on a single thread using the Clarabel optimizer [20] via JuMP’s disciplined convex programming interface [21].

**Reachability Analysis** We use a naive version of the defect hull algorithm proposed by Chan [8] to perform reachability analysis. We first compute a reference trajectory using a min-fuel objective begetting an ignition point  $n_{I,ip}$ . The reachability polytope  $P_i$  is then initialized from this singleton.

At each iteration the algorithm, depicted in Figure 7 picks a point  $n_{I,o}$  on the surface of  $P_i$  and selects a random direction  $\hat{d}$  to serve as an expansion vector. It then solves the defect hull problem:

Problem 2: Defect hull trajectory optimization problem

$$\begin{aligned}
 & \min_{x,u} -\mu \\
 \text{s.t.} \quad & n_I = r_I(0.5) = \mu \hat{d} + n_{I,o} \\
 & \text{Eq. (1), (2), (3), (5), (6), (16), (22), (23), (24),} \\
 & \text{(25), (26), (27), (28), (29), (30), (31), (32) .}
 \end{aligned} \tag{P2}$$

The ensuing solution  $n_I^*$  is then the ignition point farthest from  $n_{I,o}$  along  $\hat{d}$  that retains feasibility with respect to our state and control constraints. The reachability polytope is then expanded to  $P_{i+1} = \text{convexhull}(P_i, n_I^*)$  and the process is repeated.



**Fig. 8 Example reusable launch vehicle from Kerbal Space Program.**

$m_{\text{dry}}$	10.088 kg
$m_{\text{wet}}$	19.516 kg
$J_{B,\text{dry}}$	$\text{diag}(4.4, 4.4, 0.040) \times 10^5 \text{ kg m}^{-2}$
$J_{B,\text{wet}}$	$\text{diag}(5.6, 5.6, 0.053) \times 10^5 \text{ kg m}^{-2}$
$I_{\text{sp}}$	300 s
$u_{\text{max}}$	936 kN
$r_{B,\text{engine}}$	[0; 0; -4.1] m
$r_{B,\text{fin}}$	[0; 0; 9.2] m
$\theta_{\text{max}}$	10.5°
$\omega_{\text{max}}$	15°
$C_{\text{damp}}$	10

**Table 1 Vehicle parameters**

#### IV. Numerical Example

Our example vehicle is a simple vertical takeoff vertical landing reusable launch vehicle in the medium-fidelity 6-DoF simulator Kerbal Space Program (KSP) using the Ferram Aerospace Research aerodynamics extension. The detailed parameters are in Table 1, where  $\text{diag}(v)$  refers to the matrix with  $v$  on the diagonal.

Our reference scenario depicts a return to launch site trajectory for the first stage on the fictional planet Kerbin, which imposes several challenges compared to a similar scenario on Earth:

- Kerbin is small. Kerbin's radius (600 km) is similar to Charon's. As a result, the tangent plane approximation breaks down quickly, forcing us to use a spherical altitude model.
- Kerbin rotates very quickly. Kerbin completes a rotation about its axis in only 6 hours, causing rotating reference frame effects to become quickly noticeable.
- Kerbin is extremely dense. In spite of its small size, Kerbin's surface gravity is approximately the same as Earth's. The gravitational field therefore falls off very quickly, thus causing our use of a spherical gravity model.
- Kerbin has 20% shorter atmosphere than Earth's, reducing the opportunity for aerodynamic maneuvering.

All four effects would also occur in a similar trajectory on Earth but with smaller magnitudes. The most important effects of Kerbin's odd dimensions are the latter two; Kerbin's rapid gravitational falloff causes errors of approximately  $60 \text{ m s}^{-1}$  over trajectories of our duration and its shrunk atmosphere causes peak aerodynamic authority to happen very close to engine ignition for our trajectories.

Our vehicle was constructed using stock parts in KSP, consisting of five fuel tanks, landing legs, and fin actuators. Our return scenario depicts only the bottom-most fuel tank being full, with the others empty. KSP models fuel drain in the stage as occurring from all tanks in the stage at once therein causing the moment of inertia and center of mass to move rapidly towards the engine as fuel is depleted. By pre-draining the upper tanks, we approximate the real effect of ullage on the center of mass and moment of inertia.

We extracted aerodynamic data from the game using a custom version of the Kerbal Remote Procedure Call extension that allowed us to query the aerodynamic forces and torques on the vehicle at different velocities, altitudes, and control configurations. To extract the data needed to build the aerodynamic lookup tables, we performed sweeps over each angle of attack axis up to 25 degrees, each plane of aerodynamic control actuation up to 45 degrees, and in mach number up to mach 3.0. The data at each angle of attack was referenced to the control condition of minimum aerodynamic force (corresponding to when each actuator was pointed into the wind vector) and the body forces and moments were then extracted. The drag polar for the fin actuators was then developed from a least squares fit to the lower surface of a convex hull at this condition.

All code, including our dynamics, convex subproblem solver, reachability analysis implementation, and plot generation utilities, as well as the modified Kerbal Remote Procedure Call extension and our post-processing scripts are available in the <https://github.com/BenChung/DualPhaseDescent.jl> repository. Additionally, we include raw and processed aerodynamic data for our vehicle alongside the KSP craft file in the repository. The version of the code and

$S_{\alpha,\max}$	1	$r_I$	[0.5; 2.5; 15]km
$S_{\text{thrust,min}}$	20	$v_I$	[0; -150; -350]m/s
$S_{\text{fin,bound}}$	10	$R_{B/I,i}$	[-0.98; 0]
$S_{\ \omega\ _2,\max}$	0.1	$\omega_{B,i}$	[0; 0]rad s <sup>-1</sup>
$S_{q,\max}$	$5 \times 10^{-4}$	$r_f$	[0; 0; 0]m
$S_{q\alpha,\max}$	$1 \times 10^{-6}$	$v_f$	[0; 0; 0]m s <sup>-1</sup>
		$R_{B/I,f}$	[0; 0]°
		$\omega_{B,f}$	[0; 0]rad s <sup>-1</sup>
		$\gamma$	60°
		$v_{\text{small}}$	100 m/s
		$\rho_{\max}$	$8 \times 10^4$ Pa
		$\chi_{\max}$	$1 \times 10^6$ Pa °

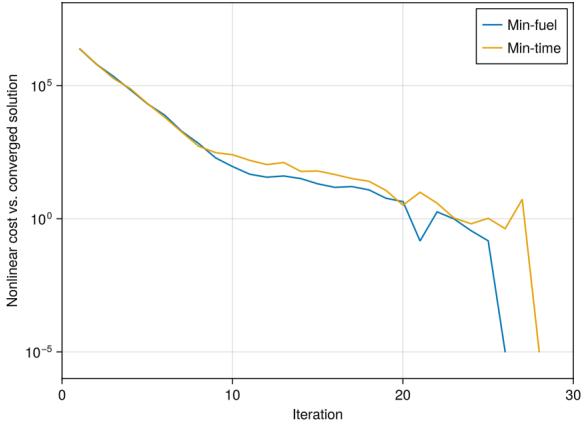
**Table 2** CTCS constraint scaling.

**Table 3** Trajectory parameters

$N$	41
$\beta$	2.0
$\alpha$	2.0
$\rho_0$	0.0
$\rho_1$	0.25
$\rho_2$	0.7
$r_{\text{init}}$	8.0
$w_m$	1000
$w_n$	50
$w_l$	100

**Table 4** Optimizer parameters

**Fig. 9** Convergence histories for the min-fuel and min-time problems.



supporting material as submitted is on Git branch `SciTech2025`.

The parameters for our reference return problem are given in Table 3. The problem was chosen to be comparable to the post-retropropulsion trajectory from Falcon 9 missions [22].

**Initialization Problem** To initialize the reachability analysis we need a reference trajectory from which to start exploring the feasible ignition points. For illustration, we will show both a min-fuel and min-time version of this problem. SCP is initialized from a zero-control trajectory propagated forward from the initial condition; for this section, we set our tolerance at 1e-5. Table 2 shows the CTCS penalty weights we used for each constraint applied in continuous time. Optimizer parameters are shown in table 4; starting from  $r_{\text{init}}$  we expand and contract the trust region weight by  $\alpha$  and  $\beta$  respectively when the relative improvement is better than  $\rho_2$  or smaller than  $\rho_1$ , and reject the iterate if it is less than  $\rho_0$ . For convergence, we apply the penalties  $w_m$  to the linearization error of the dynamics,  $w_n$  to the linearization error of the terminal constraint, and  $w_l$  on the CTCS equality state violation at each iteration.

Convergence histories for the initialization problems are shown in Figure 11.

We show a min-fuel solution in Figure 10. Two features stand out compared to vacuum or lift-free formulations such as [2]:

- The rocket maneuvers (“swerves”) in the latter phases of the aerodynamic trajectory, reducing the energy going into the propulsive phase.

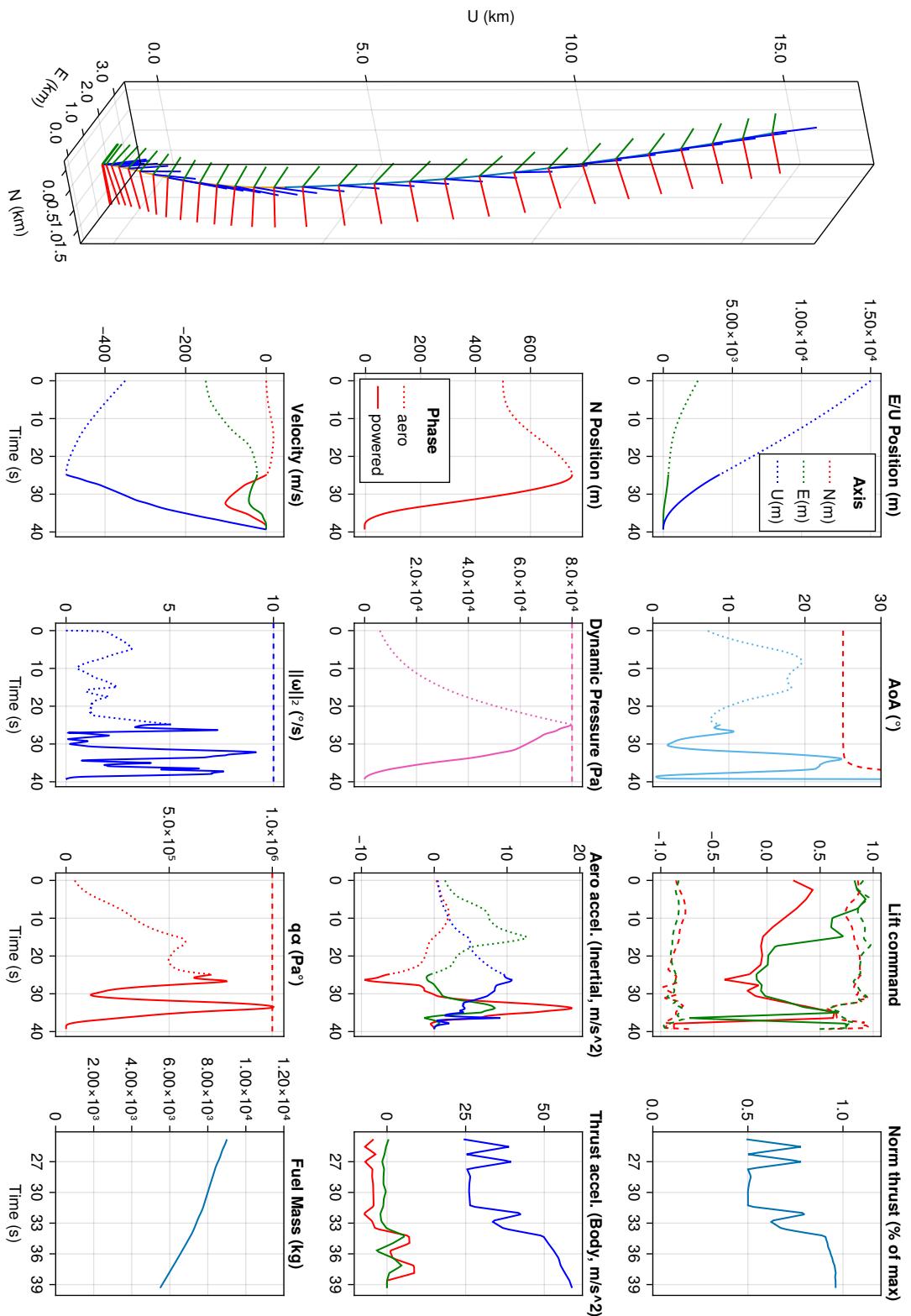
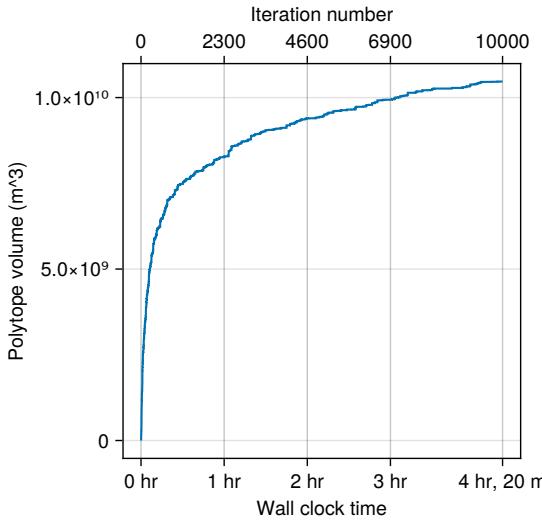
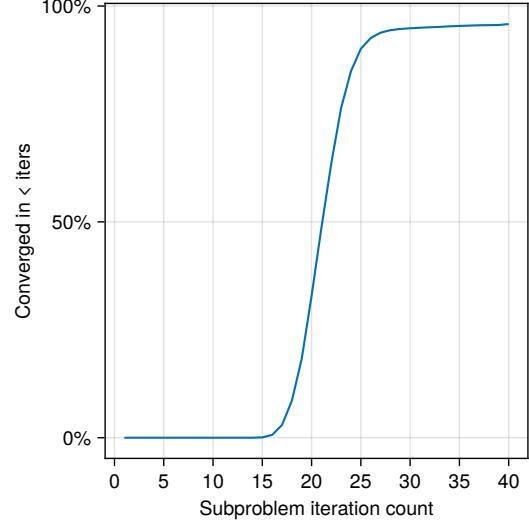


Fig. 10 Initialization min-time trajectory



**Fig. 11** Convergence history and wall clock time for the reachability problem.



**Fig. 12** Subproblem convergence history.

- The powered descent is done at nonzero angle of attack until the last phases to use lift to non-propulsively cancel velocity.

These features are unique to coupled aerodynamic-propulsive optimization; without combining the effects the optimizer would be unable to trade off position for reduced velocity and therefore reduced fuel consumption.

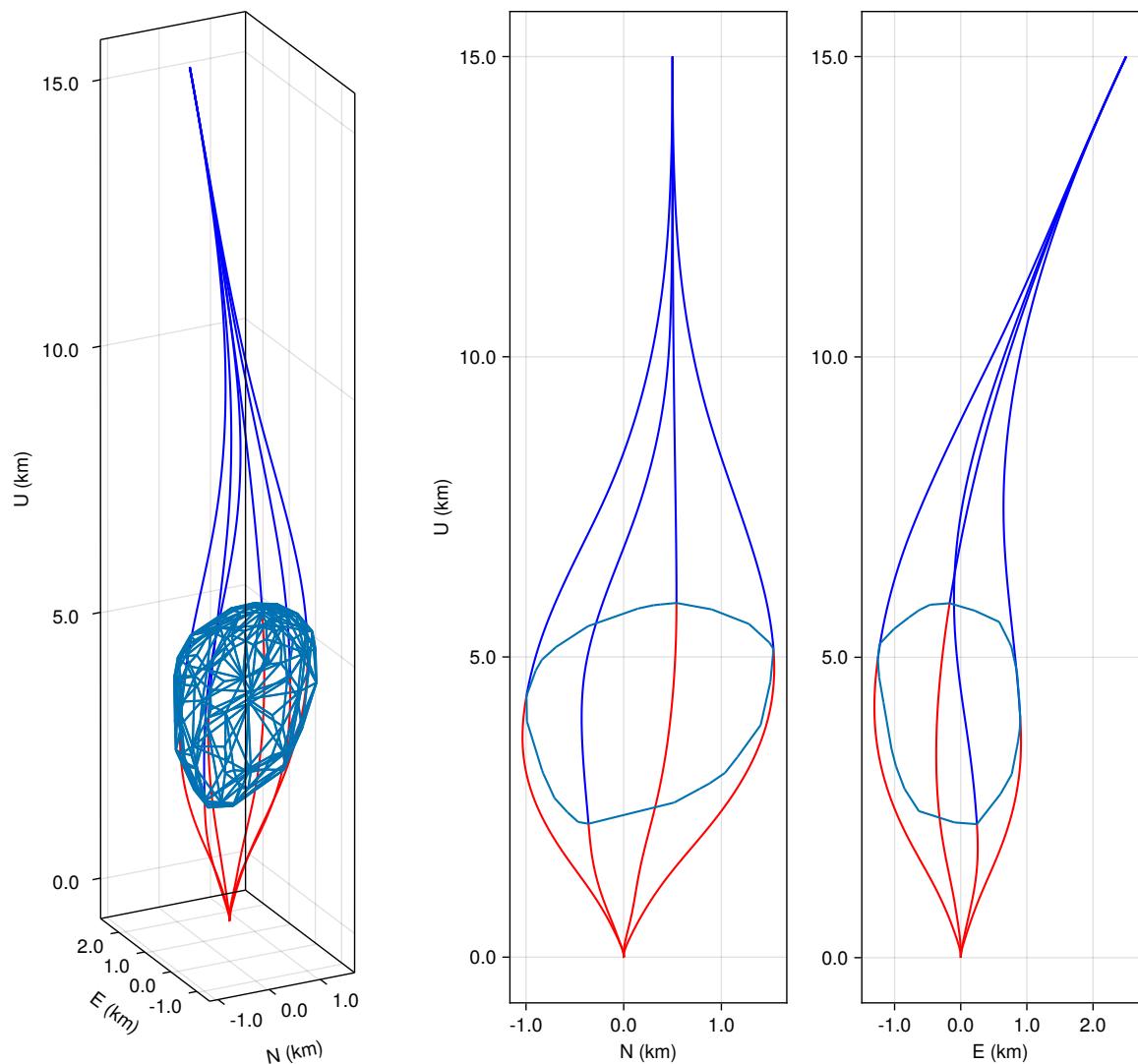
We would furthermore like to highlight the utility of the CTCS transcription for this problem. The damped attitude dynamics of the rocket are still much faster than the control or state discretization and are very wont to create inter-sample constraint violation. By using CTCS, we can minimize this inter-sample constraint violation without needing to refine our optimization time grid. In particular, we highlight the instantaneous constraint activation shown in the dynamic pressure and  $q\alpha$  histories.

The no-stall, travel-limited aerodynamic actuator restriction imposed by Eq. (16) are also tracked well with the CTCS method, with the history shown in the plot of lift commands over the trajectory, where the trajectory tracks a rapidly changing limit with one or both of its aerodynamic controls.

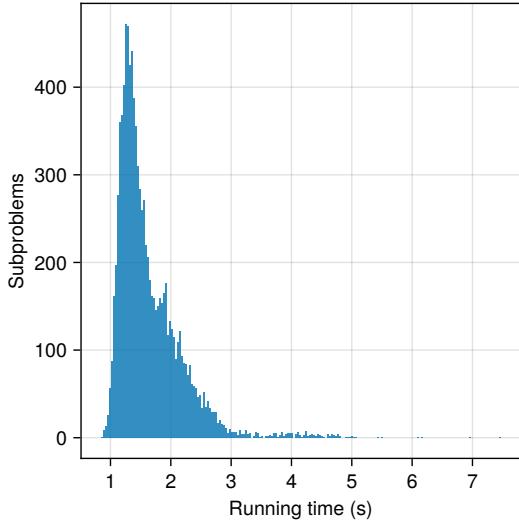
**Reachability** We then initialize the reachability analysis from the min-fuel ignition point and expand the hull for 10,000 iterations. Figure 11 shows the convergence rate as a function of polytope volume through the reachability problem by iteration and by wall clock time. Figure 12 shows the cumulative convergence of each subproblem by SCP iteration. The curve does not reach 100 % since we reject iterations whose solutions were not dynamically feasible; we rejected 4.3 % of generated solutions for dynamic infeasibility and 0.02 % due to numerical instabilities encountered during the solve. Total elapsed time was 4 hours, 20 minutes running on a AMD Ryzen AI 9 HX 370 with 32 threads, for an average time per defect hull trajectory optimization subproblem of 1.63 seconds; the histogram of running time is shown in Figure 14.

The reachable polytope computed for our example problem is shown in Figure 13. We visualize only the extremal (farthest in each cardinal direction) trajectories in each plane for the orthogonal views. We visualize the route of sampled points taken to construct the extremal points inn the north-up plane in Figure 15, starting from the ignition point of the initialization trajectory in red and expanding outwards to the surface of the final polytope.

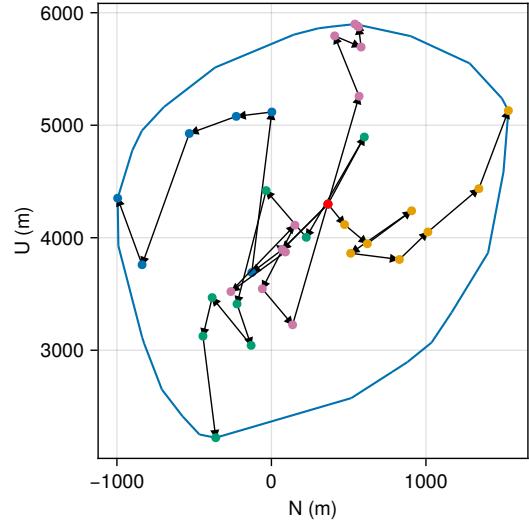
The progress of the polytope expanding shown in Figure 15 illustrates the nonconvexity of the subproblem. The subproblems make small steps inside of the reachable volume instead of immediately expanding to the reachable frontier. This slow progress is caused by the individual defect hull tasks falling into local minima based on their initialization and solution trajectory. The reachability algorithm is then able to expand further by re-initializing from a previously identified extreme point and escape the local minima only to find another one.



**Fig. 13** Reachable ignition polytope with extremal trajectories



**Fig. 14 Subproblem running time distribution.**



**Fig. 15 Expansion trajectories for extremal points on the polytope. Intermediate points are on the surface of the polytope at the time they were chosen, but may not have been the extremal points on the associated axis.**

## V. Conclusion

We described an approach for approximating the projection of a high-dimensional constrained reachability problem into a low dimensional space via nonlinear optimization. Our problem is not tractable for traditional reachability methods: not only does it have 11 state dimensions, it also has state and control constraints that set propagation based methods cannot easily accommodate. We avoid the curse of dimensionality by focusing only on the relevant low-dimensional subset of the greater reachable volume, thereby dramatically reducing the dimensionality of the space we are examining, then explore said volume using nonlinear optimization.

We show that the algorithm works to approximate the feasible ignition points for a 5 degree of freedom model of an aerodynamically controlled reusable launch vehicle's axisymmetric first stage while subject to many practical constraints. We see substantial opportunity for future work, as our approach described here was rather naive. While our algorithm was able to approximate the reachable volume, it took several hours to do so due to implementation limitations and sampling inefficiency. We think that several fruitful directions may consist of:

- Customized sparse implementation. Around 30% of our wall clock time was wasted due to convex parsing overheads; in particular, constructing the sparse constraint matrices and copying them into the solver's working memory. Using pre-allocated and customized sparse data structures and a solver optimized for such would provide an integer speedup. For example, Kamath et al. [23] was able to calculate solutions to a similar nonconvex trajectory optimization problem in 10 s of ms, though with a considerably smaller state dimension.
- Improved SCP algorithm. New work, such as Auto-SCvx [24, 25], is likely to substantially reduce the iteration count for subproblem convergence; their problem, which has similar constraints to ours, solved in less than 7 iterations on average.
- GPU implementation. As [8] suggests, the defect hull algorithm is intrinsically very parallelizable, as the selection of initial points and expansion directions is stochastic and largely independent. Furthermore, the multiple shooting sensitivity analysis is an embarrassingly parallel process as each of the constituent trajectories can be simulated independently. Chari et al. [26] used a pure-GPU implementation of SCP and achieved up to 5x uplift compared to a CPU implementation. CPU-GPU pipelining of convex optimization and sensitivity analysis may yield further performance uplift.
- Our version of Chan's defect hull algorithm is very naive in its sampling strategy as it does not use the dual-derived tangent information that Chan's proposed algorithm does. We believe that our slow convergence rate in terms of number of iterations is substantially due to this limitation. Improving reachability sampling efficiency would then reduce the number of trajectories needed to achieve sufficient polytope volume convergence.

We believe that nonlinear trajectory optimization based reachability analysis can allow the approximation of projections of highly nonlinear reachability problems. In this paper, we demonstrated that the approach can work on a practical multiphase trajectory for a aerodynamically and propulsively controlled reusable launch vehicle by approximating the spatial volume in which its ignition point can be placed. Furthermore, we believe that there are opportunities for substantial further algorithmic and implementation optimizations above our algorithm that can yield overall computation times in the minutes or seconds, instead of hours.

Our approach may also be generalized onto other classes of problem that are tractable for trajectory optimization. One concept is to extremize a metric like conditional value-at-risk, which has already been explored extensively using SCP [27, 28] which would then allow for reachability analysis over best and worst case outcomes for a stochastic system. Another would be to extend our two-phase analysis into a multiphase one derived from temporal and logical specifications [29] which would allow automatic identification of the state spaces in which transitions could occur.

## References

- [1] Blackmore, L., “Autonomous precision landing of space rockets,” *The Bridge on Frontiers of Engineering*, Vol. 4, No. 46, 2016, pp. 15–20.
- [2] Szmuk, M., Reynolds, T. P., and Açıkmese, B., “Successive convexification for real-time six-degree-of-freedom powered descent guidance with state-triggered constraints,” *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 43, No. 8, 2020, pp. 1399–1413. <https://doi.org/10.2514/1.G004549>.
- [3] Sagliano, M., Heidecker, A., Hernández, J. M., Fari, S., Schlotterer, M., Woicke, S., Seelbinder, D., and Dumont, E., “Onboard Guidance for Reusable Rockets: Aerodynamic Descent and Powered Landing,” *AIAA Scitech Forum*, 2021. <https://doi.org/10.2514/6.2021-0862>.
- [4] Bansal, S., Chen, M., Herbert, S., and Tomlin, C. J., “Hamilton-jacobi reachability: A brief overview and recent advances,” *Proc. IEEE Conf. on Decision and Control*, 2017. <https://doi.org/10.1109/CDC.2017.8263977>.
- [5] Althoff, M., Frehse, G., and Girard, A., “Set propagation techniques for reachability analysis,” *Annual Review of Control, Robotics, and Autonomous Systems*, Vol. 4, 2021, pp. 369–395. <https://doi.org/10.1146/annurev-control-071420-081941>.
- [6] Lew, T., and Pavone, M., “Sampling-based reachability analysis: A random set theory approach with adversarial sampling,” *Conference on Robot Learning*, 2020. <https://doi.org/10.48550/arXiv.2008.10180>.
- [7] Dueri, D., Açıkmese, B., Baldwin, M., and Erwin, R. S., “Finite-horizon controllability and reachability for deterministic and stochastic linear control systems with convex constraints,” *American Control Conference*, 2014. <https://doi.org/10.1109/ACC.2014.6859302>.
- [8] Chan, K. W., “Optimization-Based Reachability Analysis for Landing Scenarios,” Ph.D. thesis, Universität Bremen, 2022. <https://doi.org/10.26092/elib/1922>.
- [9] ISO 2533:1975, “Standard Atmosphere,” Standard, International Organization for Standardization, Geneva, CH, Mar. 1975.
- [10] Dukeman, G., and Calise, A., “Enhancements to an atmospheric ascent guidance algorithm,” *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2003. <https://doi.org/10.2514/6.2003-5638>.
- [11] Möller, T., and Hughes, J. F., “Efficiently Building a Matrix to Rotate One Vector to Another,” *Journal of Graphics Tools*, Vol. 4, No. 4, 1999, pp. 1–4. <https://doi.org/10.1080/10867651.1999.10487509>.
- [12] Eren, U., Dueri, D., and Açıkmese, B., “Constrained reachability and controllability sets for planetary precision landing via convex optimization,” *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 38, No. 11, 2015, pp. 2067–2083. <https://doi.org/10.2514/1.G000882>.
- [13] Ma, Y., Gowda, S., Anantharaman, R., Laughman, C., Shah, V., and Rackauckas, C., “Modeling Toolkit: A Composable Graph Transformation System For Equation-Based Modeling,” , 2021. Available at <https://arxiv.org/abs/2103.05244>.
- [14] Elango, P., Luo, D., Kamath, A. G., Uzun, S., Kim, T., and Açıkmese, B., “Successive Convexification for Trajectory Optimization with Continuous-Time Constraint Satisfaction,” , 2024. Available at <https://arxiv.org/abs/2404.16826>.
- [15] Bertsekas, D. P., “Necessary and sufficient conditions for a penalty method to be exact,” *Mathematical Programming*, Vol. 9, No. 1, 1975, pp. 87–99. <https://doi.org/10.1007/BF01681332>.

[16] Mao, Y., Szmuk, M., Xu, X., and Açıkmese, B., “Successive convexification: A superlinearly convergent algorithm for non-convex optimal control problems,” , 2018. Available at <https://arxiv.org/abs/1804.06539>.

[17] Malyuta, D., Reynolds, T. P., Szmuk, M., Lew, T., Bonalli, R., Pavone, M., and Açıkmese, B., “Convex optimization for trajectory generation: A tutorial on generating dynamically feasible trajectories reliably and efficiently,” *IEEE Control Systems Magazine*, Vol. 42, No. 5, 2022, pp. 40–113. <https://doi.org/10.1109/MCS.2022.3187542>.

[18] Rackauckas, C., Ma, Y., Martensen, J., Warner, C., Zubov, K., Supekar, R., Skinner, D., and Ramadhan, A., “Universal differential equations for scientific machine learning,” , 2020. Available at <https://arxiv.org/abs/2001.04385>.

[19] Hill, A., and Dalle, G., “SparseConnectivityTracer.jl,” , Oct. 2024. <https://doi.org/10.5281/zenodo.13961066>, URL <https://doi.org/10.5281/zenodo.13961066>.

[20] Goulart, P. J., and Chen, Y., “Clarabel: An interior-point solver for conic programs with quadratic objectives,” , 2024. Available at <https://arxiv.org/abs/2405.12762>.

[21] Lubin, M., Dowson, O., Dias Garcia, J., Huchette, J., Legat, B., and Vielma, J. P., “JuMP 1.0: Recent improvements to a modeling language for mathematical optimization,” *Mathematical Programming Computation*, Vol. 15, No. 3, 2023. <https://doi.org/10.1007/s12532-023-00239-3>.

[22] Ecker, T., Karl, S., Dumont, E., Stappert, S., and Krause, D., “Numerical Study on the Thermal Loads During a Supersonic Rocket Retropropulsion Maneuver,” *AIAA Journal of Spacecraft and Rockets*, Vol. 57, No. 1, 2020, pp. 131–146. <https://doi.org/10.2514/1.A34486>.

[23] Kamath, A. G., Elango, P., Yu, Y., Mceowen, S., Chari, G. M., Carson III, J. M., and Açıkmese, B., “Real-Time Sequential Conic Optimization for Multi-Phase Rocket Landing Guidance,” *IFAC World Congress*, 2023. <https://doi.org/https://doi.org/10.1016/j.ifacol.2023.10.1444>.

[24] Mceowen, S., Calderone, D. J., Tiwary, A., Zhou, J. S. K., Kim, T., Elango, P., and Açıkmese, B., “Auto-tuned Primal-dual Successive Convexification for Hypersonic Reentry Guidance,” , 2024. Under review for the AIAA Jurnal of Guidance, Control, and Dynamics. Available at <https://arxiv.org/abs/2411.08361>.

[25] Mceowen, S., Calderone, D. J., Tiwary, A., Zhou, J. S. K., Kim, T., Elango, P., and Açıkmese, B., “Auto-tuned Primal-dual Successive Convexification for Hypersonic Reentry Guidance,” *AIAA Scitech Forum*, 2025.

[26] Chari, G. M., Kamath, A. G., Elango, P., and Açıkmese, B., “Fast Monte Carlo Analysis For 6-DoF Powered-Descent Guidance via GPU-Accelerated Sequential Convex Programming,” *AIAA Scitech Forum*, 2025. <https://doi.org/10.2514/6.2024-1762>.

[27] Echigo, K., Sheridan, O., Buckner, S., and Açıkmese, B., “Dispersion Sensitive Optimal Control: A Conditional Value-at-Risk Based Tail Flattening Approach via Sequential Convex Programming,” *IEEE Transactions on Control Systems Technology*, Vol. 32, No. 6, 2024, pp. 2468–2475. <https://doi.org/10.1109/TCST.2024.3427910>.

[28] Echigo, K., “From Theory towards Flight: Convex Optimization based Approaches for Non-convex, Stochastic, and Realistic Aerospace Missions,” Ph.D. thesis, University of Washington, Seattle, WA, 2024.

[29] Uzun, S., Elango, P., Garoche, P.-L., and Açıkmese, B., “Optimization with Temporal and Logical Specifications via Generalized Mean-based Smooth Robustness Measures,” *arXiv preprint arXiv:2405.10996*, 2024. URL <https://doi.org/10.48550/arXiv.2405.10996>.