CPEG: A Convex Predictor-corrector Entry Guidance Algorithm

Kevin Tracy
The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
ktracy@cmu.edu

Zachary Manchester
The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
zacm@cmu.edu

***Abstract*— As scientific and crewed payloads have more demanding goals, precise atmospheric entry guidance is playing an increasing role in mission success. State-of-the-art entry guidance algorithms are structured in a predictor-corrector framework, where a simulation is used to predict a trajectory, and corrections are then made to the control inputs. These guidance methods are simple and effective, but current algorithms assume low lift-to-drag entry vehicles, are limited to only bank-angle control, and have a limited ability to guarantee the safety of the vehicle. We propose a new predictor-corrector entry guidance method that formulates the correction step as a convex optimization problem. This allows for more flexibility in specifying the vehicle's dynamics and control inputs, and the ability to explicitly handle safety constraints such as heating, pressure, and acceleration limits. We test the new algorithm in Mars entry scenarios similar to the Mars Science Laboratory with both bank-angle control and bank-angle plus angle-of-attack control, demonstrating both its performance and ability to generalize to future vehicle capabilities.**

TABLE OF CONTENTS

1. INTRODUCTION

In 1971 the Soviet Union's Mars-2 spacecraft made history by entering the Martian atmosphere before impacting the surface. Nine days later, an identical Mars-3 spacecraft performed the first soft-landing on the Martian surface, ushering in a new era in planetary exploration. NASA followed with successful Mars landings in 1976 with Viking 1 and 2 and has since then landed and operated multiple robotic systems on the Martian surface [1].

Entry vehicle architectures can be divided into three broad categories [1]: 1) Ballistic entry is an uncontrolled descent with drag as the only force, 2) unguided ballistic-lifting entry has an uncontrolled non-zero lift force, and 3) guided ballistic-lifting entry has some control over the vehicle's lift vector. Controlled entry guidance allows for the prioritization of landing locations with scientific merit instead of just those that minimize risk to the vehicle.

978-1-6654-3760-8/22/$31.00 ©2022 IEEE

The Mars Science Laboratory (MSL) carrying the Curiosity rover touched down in 2012 as the first Mars entry vehicle with guided ballistic-lifting entry. MSL had control over the vehicle bank angle during entry, enabling control of the direction of the lift vector within the lifting plane. While MSL dramatically reduced the size of the landing ellipse from over 100 km to 10 km, its guidance is still too coarse for pinpoint landings. By developing more performant entry guidance capabilities, entry vehicles could effectively place robotic or crewed landers in desirable science collection areas, including high altitude sites.

Much of the work on guidance for low lift-to-drag entry vehicles originated with the Apollo terminal guidance methods. These algorithms, as described in [2], rely on control of the bank angle with simple switching manuevers to control the cross-range and down-range errors. Slightly modified versions have been developed for use with more recent Mars entry vehicles, such as in [3]. Current research investigates the use of predictor-corrector algorithms [4] to improve the landing accuracy. A popular predictor-corrector formulation that exhibits bank-angle switching behavior is the Fully Numerical Predictor-corrector Entry Guidance (FNPEG) algorithm [5]. In the baseline FNPEG algorithm, Newton's method is used to solve for a static bank-angle that satisfies a terminal downrange distance constraint, and the sign of the bank-angle is modulated to control crossrange errors [6]. Here, the prediction phase is used to generate gradients for the terminal constraint, and corrections are applied to the open-loop commanded bank-angle in an effort to satisfy these terminal constraints. This framework is simple and effective but requires significant added complexity for incorporation of safety constraints or changes to the vehicle control inputs.

Trajectory optimization for offline planning of entry vehicle trajectories has been explored in [7] and [8], where the nonconvex optimal control problem was solved by linearizing the nonlinear dynamics and constraints, solving a conic optimization problem with a trust region, and repeating until convergence. This successive-convexification method was used instead of standard NonLinear Programming (NLP) solvers, like SNOPT [9] or IPOPT [10], because it is able to directly handle second-order cone constraints instead of relying on local linear approximations. Optimal trajectories computed offline were then paired with an optimization-based tracking controller, as described in [7] and [8]. While these formulations are able to stabilize a trajectory, there are no guarantees that safety constraints can be satisfied online. Also, the computational complexity of the trajectory-optimization formulation makes these methods intractable for real-time control onboard an entry vehicle.

The Convex Predictor-corrector Entry Guidance (CPEG) algorithm proposed in this paper combines ideas from trajectory optimization with the predictor-corrector guidance

framework by solving a constrained optimization problem during the correction step. First, the dynamics of the entry vehicle with the current control plan are simulated to a target altitude for a predicted trajectory. Next, the vehicle dynamics are linearized about the predicted trajectory and a convex trajectory optimization problem is solved that minimizes landing error. By solving for a correction using convex optimization, CPEG is able to reason about the full state and control history to inform the correction instead of just the final state. This also allows for the vehicle's safety constraints, such as heating, pressure, and acceleration, to be explicitly included in the correction computation. Our specific contributions in this paper are:

1. A general quasi-linear formulation of entry vehicle dynamics that is well-suited to numerical optimization.

2. A predictor-corrector entry guidance algorithm with a highly generalizable correction step utilizing convex optimization.

3. Customized trust regions and objective functions for entry vehicles with multiple control modalities.

The paper proceeds as follows: In Section 2, the classic Vinh entry vehicle dynamics are compared with a more modern Cartesian approach. In Section 3, the details of the full nonconvex trajectory optimization problem are discussed. In Section 4, the CPEG algorithm is derived. In Section 5, CPEG is validated on entry vehicles with bank-angle control, as well as bank-angle and angle-of-attack control. Finally, Section 6 outlines our conclusions and potential future research directions.

2. ENTRY VEHICLE DYNAMICS

Despite much of the recent powered-descent guidance literature using Cartesian state representations, entry vehicles are still most often represented in spherical coordinates. In this section, the traditional entry vehicle dynamics denoted below as the "Vinh" model will be discussed, as well as an alternative Cartesian formulation.

The Vinh Model

The classic Vinh model, presented in 1976 in [11] and again a few years Later in Vinh's textbook [12], has been the standard method for simulating entry vehicles for the past 45 years. Parameterizing the entry vehicle in spherical coordinates, the state in the Vinh model contains familiar terms like latitude, longitude, and flight-path angle. Despite being highly nonlinear and prone to scaling issues, it is the most common dynamics model in the literature [6], [11]–[16].

The dynamics in the Vinh model are calculated with the angle-of-attack, α, bank-angle, σ, flight-path angle, γ, longitude, θ, latitude, ϕ, and heading angle, ψ. The resulting equations of motion over a planet that's rotating with a constant angular velocity Ω are,

$$\dot{r} = V \sin\gamma, \tag{1}$$

$$\dot{\theta} = V \cos\gamma \sin\psi / (r \cos\phi), \tag{2}$$

$$\dot{\phi} = V \cos\gamma \cos\psi / r, \tag{3}$$

$$\begin{aligned}\dot{V} = &-D - \sin\gamma / r^2, \\ &+ \Omega^2 r \cos\phi \sin\gamma \cos\phi \\ &- \Omega^2 r \cos\phi \cos\gamma \sin\phi \cos\psi,\end{aligned} \tag{4}$$

$$\begin{aligned}\dot{\gamma} = &L \cos\sigma / V + \left(V^2 - 1/r\right) \cos\gamma / (Vr) \\ &+ 2\Omega \cos\phi \sin\psi + \Omega^2 r \cos\phi \cos\gamma \cos\phi / V \\ &+ \Omega^2 r \cos\phi \sin\gamma \sin\phi \cos\psi / V,\end{aligned} \tag{5}$$

$$\begin{aligned}\dot{\psi} = &L \sin\sigma / (V \cos\gamma) + V \cos\gamma \sin\psi \tan\phi / r \\ &- 2\Omega(\tan\gamma \cos\psi \cos\phi - \sin\phi) \\ &+ \Omega^2 r \sin\phi \cos\phi \sin\psi / (V \cos\gamma),\end{aligned} \tag{6}$$

where r is the normalized radial distance from the center of the planet, V is the normalized planet-relative velocity, and L and D are the magnitudes of the lift and drag accelerations.

This model is highly nonlinear in both the state and the control, even when the planetary motion is ignored. While the planet's angular velocity is assumed to be constant, its inclusion in the dynamics still contributes significant nonlinearities. Because of this, much of the literature ignores the planet's angular velocity [7]. There are also scaling issues present if these equations are naively implemented. Since r and V are not angles, they are usually of a much larger magnitude than the rest of the state. This can lead to poor accuracy in variable time-step integrators, as well as ill-conditioning in numerical trajectory optimization.

Cartesian Entry Dynamics

We have found that entry vehicle dynamics are both simpler to derive and numerically better-conditioned when represented in standard Cartesian coordinates instead of the spherical coordinates used in the Vinh formulation. This state representation is popular with the powered-descent guidance community, albeit without any aerodynamic forces in the dynamics [17]–[19].

We assume a planet-fixed frame P is aligned with an inertial frame N along the z axis. The planet spins with angular velocity $\omega \in \mathbb{R}^3$ in the positive z direction, making the velocity of the entry vehicle the following:

$$^P v = {}^N v - \omega \times r, \tag{7}$$

where ${}^N v \in \mathbb{R}^3$ is the inertial velocity, ${}^P v \in \mathbb{R}^3$ is the planet relative velocity, and $r \in \mathbb{R}^3$ is the position of the entry vehicle in the planet frame. This expression can be differentiated once more to provide the relationship between the inertial and planet-relative accelerations:

$$^P a = {}^N a - 2(\omega \times {}^P v) - \omega \times (\omega \times r). \tag{8}$$

The state of the entry vehicle can be parameterized with the planet-relative position vector r, and planet relative velocity ${}^P v$ denoted as just v, both expressed in the coordinates of the planet frame. The Cartesian dynamics can now be written in state space as,

$$\begin{bmatrix} v \\ a \end{bmatrix} = \begin{bmatrix} 0 & I \\ -[\omega\times]^2 & -2[\omega\times] \end{bmatrix} \begin{bmatrix} r \\ v \end{bmatrix} + \begin{bmatrix} 0 \\ a_g + a_D + a_L \end{bmatrix}, \tag{9}$$

where $[\omega\times]$ is the skew-symmetric cross product matrix,

$$[\omega\times] = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}. \tag{10}$$

One of the main benefits of the dynamics in equation (9) is the linear kinematics. This means that linear approximations

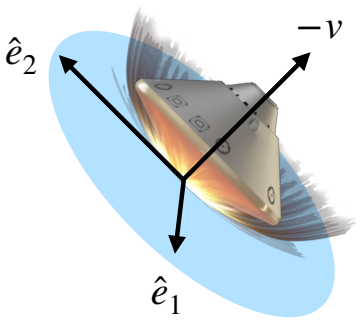

Figure 1. The E frame is fixed to the entry vehicle, with $\hat{e}_1$ in the direction of the specific angular momentum vector, and $\hat{e}_2 = \hat{v} \times \hat{e}_1$. When defined in this frame, the lift vector can be expressed using only $\hat{e}_1$ and $\hat{e}_2$.

of the relationship between position and velocity are exact, and the only nonlinearities present are in the accelerations. The gravitational acceleration in the direction of the planet's center is expressed assuming simple spherical gravity:

$$a_g = -\frac{\mu}{\|r\|^3} r, \tag{11}$$

where $\mu \in \mathbb{R}$ is the standard gravitational constant for the given planet. The acceleration caused by the drag force is in the direction opposing velocity, and is calculated as,

$$a_D = -\frac{1}{2m} \rho A C_d \|v\| v, \tag{12}$$

where $m \in \mathbb{R}$ is the mass of the entry vehicle, $\rho \in \mathbb{R}$ is the atmospheric density, $A \in \mathbb{R}$ is the aerodynamic reference area, and $C_d \in \mathbb{R}$ is the coefficient of drag. In this work, the atmospheric density $\rho \in \mathbb{R}$ will be represented by a piecewise exponential function [16].

For the description of the lift acceleration, a reference frame is defined that describes a plane about the entry vehicle that is orthogonal to the velocity vector. This two-dimensional frame, referred to as the E frame and depicted in Fig. 1, has two basis vectors described by the following:

$$\hat{e}_1 = \frac{r \times v}{\|r \times v\|}, \tag{13}$$

$$\hat{e}_2 = \frac{v \times \hat{e}_1}{\|v \times \hat{e}_1\|}. \tag{14}$$

The magnitude of the lift vector is calculated as,

$$\|L\| = \frac{1}{2m} C_L \rho(r) A \|v\|^2, \tag{15}$$

where $C_L \in \mathbb{R}$ is the coefficient of lift. In the case where the entry vehicle only has control over the bank-angle, the resulting lift acceleration can be described by the magnitude of the lift and the bank-angle:

$$a_L = \|L\|(\sin(\sigma)\hat{e}_1 + \cos(\sigma)\hat{e}_2). \tag{16}$$

In the case where the entry vehicle can control both the angle-of-attack as well as the bank-angle, the lift vector can be written as,

$$a_L = \|L\|(\ell_1 \hat{e}_1 + \ell_2 \hat{e}_2), \tag{17}$$

subject to the constraint $||\ell_1^2 + \ell_2^2|| \leq 1$. Here the lift acceleration is a linear function of the control inputs, which is a key feature when this model is linearized in an optimization problem. Both the Vinh model and the Cartesian model are nonlinear, but the Cartesian model behaves significantly better under linearization, making it a far better candidate for trajectory optimization.

State and Control Definitions

In the case where only the bank-angle is controlled, the state is augmented with the bank-angle, and the sole control input is the derivative of this bank-angle with respect to time. This allows for cost functions that specify desired behavior for the derivative of the bank-angle, with the state and control as the following:

$$x = \begin{bmatrix} r^T & v^T & \sigma \end{bmatrix}, \tag{18}$$

$$u = \dot{\sigma}. \tag{19}$$

These dynamics are now in control-affine form with linear kinematics. For the case with actuation of both the bank angle and angle-of-attack, the state and control are the following:

$$x = \begin{bmatrix} r^T & v^T \end{bmatrix}^T, \tag{20}$$

$$u = \begin{bmatrix} \ell_1 & \ell_2 \end{bmatrix}^T, \tag{21}$$

where ℓ_1 and ℓ_2 were defined in (17).

3. Trajectory Optimization

Feedback control laws for entry vehicles suffer in performance due to the severe underactuation of the vehicle. This is, in part, due to the fact that an entry vehicle has very limited ability to speed up or slow down in the along-track direction. To deal with this, it makes more sense to solve the guidance problem with a holistic planning approach, one that can reason about this limited control authority and plan for it. Therefore, we pose this problem as a trajectory optimization problem, where a locally optimal state trajectory and control plan can be solved for numerically.

Safety Constraints

Three key vehicle safety constraints — heating, pressure, and acceleration — are most dependent on the atmospheric density. Unfortunately, this is also the part of the environment in which there is the largest amount of uncertainty. The atmospheric density is often only known to roughly within a factor of two, with even less known about the wind conditions [16].

The heating constraint has to do with the max allowable heat rate that the ablative heat shield can withstand [20]. This is measured in power per square centimeter, and it is expressed as the following:

$$\dot{Q} = k_q \sqrt{\rho} V^{3.15} \leq \dot{Q}_{max}. \tag{22}$$

This function is nonlinear but can be locally approximated with linear functions during the correction step. The next safety constraint is the maximum dynamic pressure on the entry vehicle, which is expressed as the following:

$$q = .5 \rho V^2 \leq q_{max}. \tag{23}$$

The last safety constraint is the maximum allowable normal load, which is the total aerodynamic force on the entry

vehicle. This is expressed as a norm of the lift and drag forces:

$$a = \sqrt{\|L\|^2 + \|D\|^2} \leq a_{max}. \tag{24}$$

Full Nonconvex Formulation

In order to formulate a convex correction problem, we first consider the full nonlinear non-convex problem. First, the dynamics described in equation (9) are discretized with an explicit integrator like the classic fourth-order Runge-Kutta method [21], giving a discrete-time dynamics model of the form,

$$x_{k+1} = f(x_k, u_k, \Delta t_k). \tag{25}$$

No assumptions have been made about the control configuration in this dynamics model: it can account for either bank-angle-only or bank-angle plus angle-of-attack control. The full nonlinear trajectory optimization problem has the form,

$$\begin{aligned} \underset{x,\,u,\,\Delta t}{\text{minimize}} \quad & \ell_N(x_N, u_N) + \sum_{k=1}^{N-1} \ell_k(x_k, u_k) \\ \text{subject to} \quad & x_{k+1} = f(x_k, u_k, \Delta t_k) \forall k, \\ & g_k(x_k, u_k) \leq 0 \qquad\qquad \forall k, \\ & \Delta t_{min} \leq \Delta t_k \leq \Delta t_{max} \; \forall k, \\ & x_N = x_{goal}, \end{aligned} \tag{26}$$

where safety constraints (22)—(24) are included in the inequality constraint function $g_k(x_k, u_k)$. Note that this is a free-final-time problem in which the Δt_k are decision variables in addition to the states and controls. This is necessary due to the inability of the entry vehicle to reach its goal state at an arbitrarily specified time. Problem (26) is nonconvex due to both the nonlinear dynamics, as well as the variable time between knot points. It is worth noting that, even with linear continuous-time dynamics, the discrete-time dynamics constraints (25) become nonlinear when the time step is made to be a decision variable.

Trajectory optimization problems like (26) can be solved with a variety of methods. One standard approach is to use an off-the-shelf NLP solver like IPOPT [10] or SNOPT [9]. Alternatively, more specialized trajectory optimizers like ALTRO can be used [22], [23]. While computationally tractable using one of the described methods, the nonconvexity of the problem means there are no available guarantees for the quality of the solution or convergence of the solver. As a result, running nonconvex trajectory optimization onboard safety-critical aerospace systems is unpopular, explaining the prevalence of simpler heritage methods for entry guidance.

4. CONVEX PREDICTOR-CORRECTOR

CPEG combines ideas from numerical trajectory optimization with the classic predictor-corrector guidance framework: It uses a prediction step, in which the vehicle dynamics are simulated until a target altitude is reached, combined with a corrector step that is based on solving a local convex approximation of a nonlinear trajectory optimization problem to steer the vehicle to the desired target. These steps are then repeated until convergence is achieved. This section provides a detailed derivation of the CPEG algorithm.

Prediction and Dynamics Linearization

In the first stage of CPEG, the dynamics of the entry vehicle are simulated with a standard Runge-Kutta method using the current nominal control trajectory, $\bar{U}$, until a target altitude is reached. We denote this predicted trajectory by $\bar{X}$. After the prediction step, the discrete-time nonlinear dynamics are approximated using a first-order Taylor series,

$$\bar{x}_{k+1} + \delta x_{k+1} \approx f(\bar{x}_k, \bar{u}_k) + A_k \delta x_k + B_k \delta u_k, \tag{27}$$

where A_k and B_k are the following Jacobians,

$$A_k = \left.\frac{\partial f(x_k, u_k, \Delta t_k)}{\partial x_k}\right|_{\bar{x}_k, \bar{u}_k}, \tag{28}$$

$$B_k = \left.\frac{\partial f(x_k, u_k, \Delta t_k)}{\partial u_k}\right|_{\bar{x}_k, \bar{u}_k}. \tag{29}$$

Subtracting the dynamics of the reference trajectory from both sides, the local linear dynamics of trajectory corrections can be written as:

$$\delta x_{k+1} = A_k \delta x_k + B_k \delta u_k. \tag{30}$$

A crucial distinction between CPEG and sequential convexification methods [7], [24], [25], is that trajectory iterates are always dynamically feasible, thanks to the prediction step. This eliminates the possibility of inconsistent linearizations of the dynamics constraints [26], in which no feasible correction trajectory exists. Specifically, there is always a trivial solution to (30) of all zeros for δx and δu.

Cost Function

The cost function used in CPEG is comprised of a term that penalizes the miss distance from the target and a term that penalizes specified control behaviors. For the penalty on miss distance, putting a naive quadratic cost on the error between the final position and the desired position is inappropriate since it also penalizes altitude errors. Instead, only the position error projected onto the landing plane is penalized, effectively ignoring altitude error. Since the altitude target is implicitly satisfied during the prediction step, this allows for the correction to only apply changes to the control plan that minimize the projected miss distance. The cost function for this projected miss distance is the following:

$$\ell_{miss}(\delta X, \delta U) = \|W(r_N + \delta r_N - r_{goal})\|_2^2, \tag{31}$$

where $r_N \in \mathbb{R}^3$ is the final position in the reference trajectory, $\delta r_N \in \mathbb{R}^3$ is the correction computed for this position, and $r_{goal} \in \mathbb{R}^3$ is the desired final position for parachute deployment. To project this error onto the landing plane, we define following projection matrix,

$$W = I - pp^T, \tag{32}$$

where p is the unit vector normal to the planetary surface at the target position:

$$p = \frac{r_{goal}}{\|r_{goal}\|}. \tag{33}$$

The second part of the cost function seeks to shape the control behavior. In the case of bank-angle control, we consider two different control cost functions that produce qualitatively different behavior:

$$\ell_{\sigma,L1}(\delta U) = \lambda \|\dot{\sigma_k}\|_1, \tag{34}$$

and

$$\ell_{\sigma,quad}(\delta U) = \lambda \dot{\sigma_k}^2, \tag{35}$$

where λ is a scalar tuning parameter. The first cost function (34) penalizes the L1 norm of the derivative of the bank-angle, resulting in bank-angle trajectories with a minimum number of discrete switches. The second cost function (35) penalizes the square of the bank-angle derivative, resulting in smooth bank-angle trajectories. For the bank-angle plus angle-of-attack case, as described in (17), we apply a simple quadratic cost to the norm of the controlled lift vector, effectively penalizing high angles of attack:

$$\ell_{\sigma\alpha}(\delta U) = \lambda \|u_k\|_2^2. \tag{36}$$

Constraints

Of the three nonlinear safety constraints, two can be linearized, and the third can be converted to a conservative convex relaxation. For the heating and dynamic pressure constraints (22)–(23), a Taylor expansion of each is formed, approximating the constraint to first-order. From here, a linearized inequality constraint can be directly included in the convex correction problem. For these constraints, the linearized versions are:

$$[\nabla \dot{Q}(\bar{x}_k)]^T \delta x_k \leq \dot{Q}_{max} - \dot{Q}(\bar{x}_k), \tag{37}$$

$$[\nabla q(\bar{x}_k)]^T \delta x_k \leq q_{max} - q(\bar{x}_k). \tag{38}$$

The acceleration loading constraint (24) is nonlinear, but a conservative convex relaxation can be derived in the form of a second-order cone constraint. First, the kinematics for the velocity can be conservatively approximated as the following:

$$v_{k+1} = v_k + a_k \Delta t, \tag{39}$$

$$a_k = \frac{v_{k+1} - v_k}{\Delta t}, \tag{40}$$

$$a_k = \frac{\bar{v}_{k+1} + \delta v_{k+1} - \bar{v}_k - \delta v_k}{\Delta t}. \tag{41}$$

The maximum loading constraint can then be re-written as,

$$\|\bar{v}_{k+1} + \delta v_{k+1} - \bar{v}_k - \delta v_k\| \leq \Delta t \cdot a_{max}, \tag{42}$$

which is in the form of a convex second-order cone, and can be directly incorporated into the correction problem.

The three safety constraints from equations (37), (38), and (42), are stacked into a generic safety constraint function,

$$g_{safety}(\delta x_k, \delta u_k) \leq 0. \tag{43}$$

Trust Region

To ensure that corrections are sufficiently small that the dynamics linearizations and constraint approximations remain accurate, a trust-region constraint is added to the convex correction problem. While standard trust-region methods apply norm constraints to δX and δU [26], insight into the entry guidance problem enables a more tailored approach.

The quality of the linearization presented in (30) is highly accurate for approximating the vehicle kinematics, gravity, and atmospheric drag, but is much less accurate when applied to the bank-angle in the bank-angle-only control case. Therefore, we design a trust region that restricts corrections to the bank-angle, $\delta\sigma_k$, given the known accuracy of small-angle approximations but allows large corrections to the other states. This approach also allows us to avoid the need to adapt trust regions inside the solver, enabling faster and more reliable convergence. We apply the following trust-region constraints to each corrector problem:

$$\|\delta u_k\|_2 \leq \delta u_{max} \tag{44}$$

$$|\delta \sigma_k| \leq \delta \sigma_{max} \tag{45}$$

Convex Corrector Problem

For the case where the entry vehicle has control of only the bank-angle as described in (16), the convex correction problem can be formulated as,

$$\begin{aligned} \underset{\delta X,\, \delta U}{\text{minimize}} \quad & \ell_{miss}(\delta X, \delta U) + \ell_{\sigma}(\delta U) \\ \text{subject to} \quad & A_k \delta x_k + B_k \delta u_k = \delta x_{k+1}, \\ & g_{safety}(\delta x_k, \delta u_k) \leq 0, \\ & \|\delta u_k\|_2 \leq \delta u_{max}, \\ & |\delta \sigma_k| \leq \delta \sigma_{max}, \end{aligned} \tag{46}$$

where the miss cost function is described in (31), and the bank-angle cost function can be either (34) or (35).

For the case where the entry vehicle has control over both bank-angle and angle-of-attack as described in (17), the convex correction problem can be posed as:

$$\begin{aligned} \underset{\delta X,\, \delta U}{\text{minimize}} \quad & \ell_{miss}(\delta X, \delta U) + \ell_{\sigma\alpha}(\delta U) \\ \text{subject to} \quad & A_k \delta x_k + B_k \delta u_k = \delta x_{k+1}, \\ & g_{safety}(\delta x_k, \delta u_k) \leq 0, \\ & \|u_k + \delta u_k\|_2 \leq 1. \end{aligned} \tag{47}$$

These problems can be solved quickly and reliably by standard conic solvers such as Mosek [27], COSMO [28], and ECOS [29].

CPEG Algorithm

The full CPEG algorithm is detailed in algorithm 1. The inputs to CPEG are the current position and the current control plan. From here, the dynamics of the entry vehicle are simulated until parachute deployment with the current control plan. This predicted trajectory is then discretized and linearized, resulting in dynamics Jacobians A_k and B_k (equations (28)-(29)). From here, the convex correction problem is posed given the control configuration and cost strategy. This convex optimization problem is solved, and the correction δU is used to correct the control plan. The prediction-correction steps are repeated until the norm of the correction being made to the control plan is below a specified tolerance.

5. NUMERICAL EXPERIMENTS

Parameters roughly matching those of the Mars Science Laboratory (MSL) [3] were used to test the CPEG algorithm. All scenarios begin at an altitude of 125 km above the Martian surface with a a Mars-relative velocity of 5.845 km/second. CPEG was implemented in the Julia programming language

Algorithm 1 CPEG Algorithm

input x_0, U ▷ nominal control plan
while $\|\delta U\| >$ tolerance **do**
  $\bar{X}, \bar{U} = \text{simulate}(x_0, U)$ ▷ predict trajectory
  $A, B = \text{linearize}(\bar{X}, \bar{U})$ ▷ linearize about prediction
  $\delta X, \delta U = \text{cvx}(\bar{X}, \bar{U}, A, B)$ ▷ solve for correction
  $U \mathrel{+}= \delta U$ ▷ correct control plan
end while
return U ▷ return updated control plan

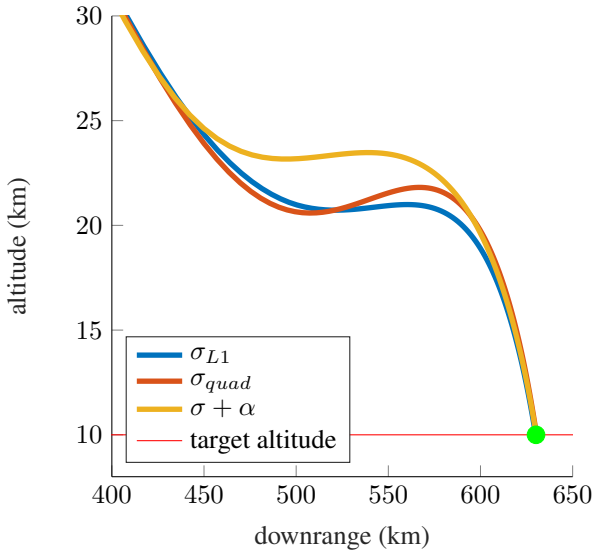

Figure 2. Altitude and downrange distance from the converged trajectories from CPEG on the three specified cases. The σ_{L1} case is with bank-angle control and an L1 penalty on bank-angle derivative, σ_{quad} is bank-angle only with a quadratic penalty on bank-angle derivative, and $\sigma + \alpha$ is control over both bank-angle and angle-of-attack. Due to the differences in control authority and cost function, all three converge on different trajectories that hit the target position at parachute deployment.

[30], using the Convex.jl optimization modeling library [31], and the Mosek [27] and OSQP [32] solvers. CPEG was validated on the following three cases:

- Bank-angle control with L1 cost penalty, denoted σ_{L1}.
- Bank-angle control with quadratic penalty, denoted σ_2.
- Bank-angle plus angle-of-attack control, denoted $\sigma + \alpha$.

For the bank-angle cases, CPEG was arbitrarily initialized with a constant bank-angle of zero, with noise added to the bank-angle derivative. For the bank-angle plus angle-of-attack case, a similar approach was used, but noise was added to the normalized lift vector. The final converged trajectories for the three cases are shown in figures 2 and 3. In all of the cases, CPEG was able to successfully guide the entry vehicle to the target point at the desired altitude.

Bank-Angle Control

For the case where the entry vehicle has only bank-angle control, the convergence of CPEG can be observed in figures 4 and 5 for the case with an L1 cost on the bank-angle derivative, and figures 6 and 7 with a quadratic cost. These

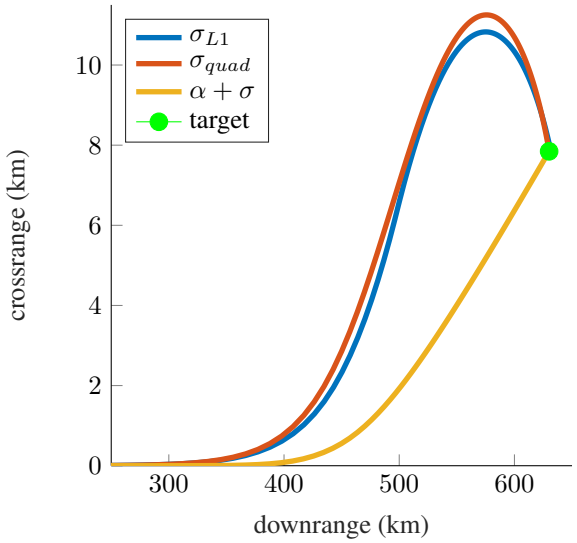

Figure 3. Crossrange and downrange trajectory data from the converged trajectories from CPEG on the three specified cases. The cases with only control over the bank-angle have to do a bank reversal to hit the target, whereas the case with control over bank-angle and angle-of-attack is able to leverage the full lift control to avoid the switching.

plots show the output of the prediction step of CPEG, where the color of the predictions is blue for the first iteration of the algorithm and turns purple, then pink for later iterations.

After convergence, the two bank-angle profiles that CPEG produced for the L1 and quadratic cost functions are shown in figure 8. The L1 cost on the derivative of the bank-angle encouraged sparsity in this derivative, resulting in a bank-angle profile that switches between constant bank-angles. For the case with a quadratic cost on the bank-angle derivative, the resulting bank-angle profile is smooth with no discrete switching behavior.

Bank-Angle Plus Angle-of-Attack Control

As described by the dynamics in equation (17), the control input for this case is the lift vector itself in the directions orthogonal to the velocity vector. This allows for manipulation of both the bank-angle and angle-of-attack and is guaranteed to be within the maximum allowable lift by the unit norm constraint in equation (47). In this control case, the control input Jacobian is constant and independent of the nominal control plan, making the linearization significantly more accurate than the bank-angle-only case. As a result, the convergence of CPEG with bank-angle plus angle-of-attack control is significantly faster than with the bank-angle alone. The evolution of the predicted trajectories is shown in figures 9 and 10, with the same coloring scheme as the bank-angle only section. After convergence, the control inputs were converted back into bank-angle and angle-of-attack and shown together in figure 11.

6. CONCLUSION

This paper proposes an improved version of the classic predictor-corrector entry guidance scheme in which the correction step is formulated as a convex optimization problem. Two control strategies were tested with CPEG: bank-angle control, and bank-angle plus angle-of-attack modulation. For the bank-angle-only case, cost functions that

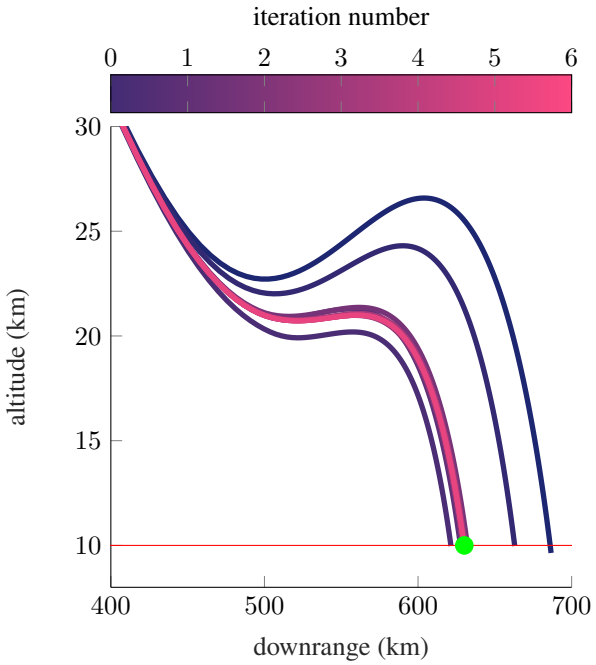

Figure 4. Predicted entry vehicle trajectories for the bank-angle only L1 penalty case, as seen by the altitude and downrange data. As the iterates continue, the entry vehicle converges on a trajectory that reaches the target at the 10km altitude mark.

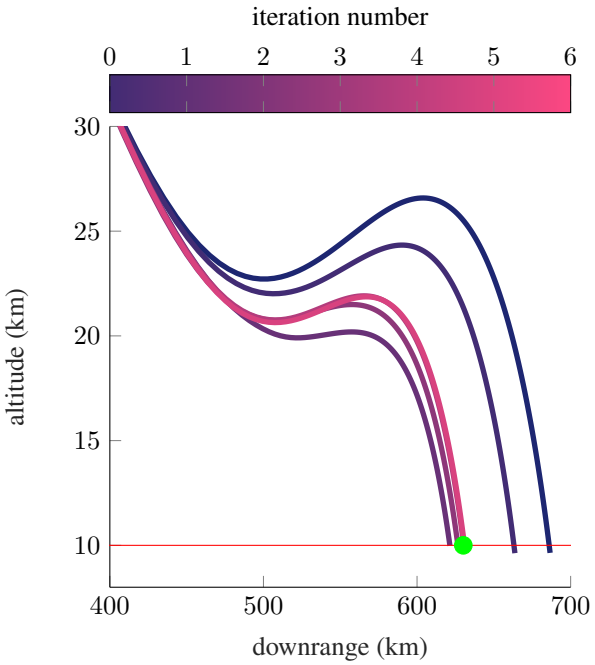

Figure 6. Predicted entry vehicle trajectories for the bank-angle only quadratic penalty case, as seen by the altitude and downrange data. The

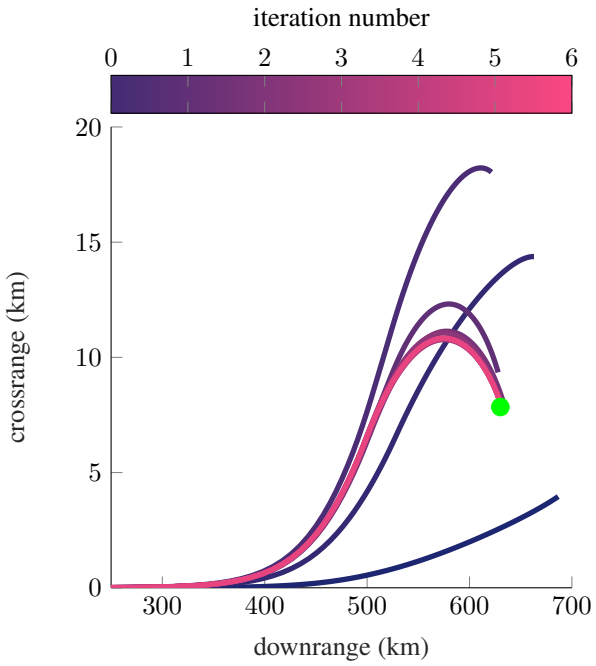

Figure 5. Predicted entry vehicle trajectories for the bank-angle only L1 penalty case, as seen by the crossrange and downrange data.

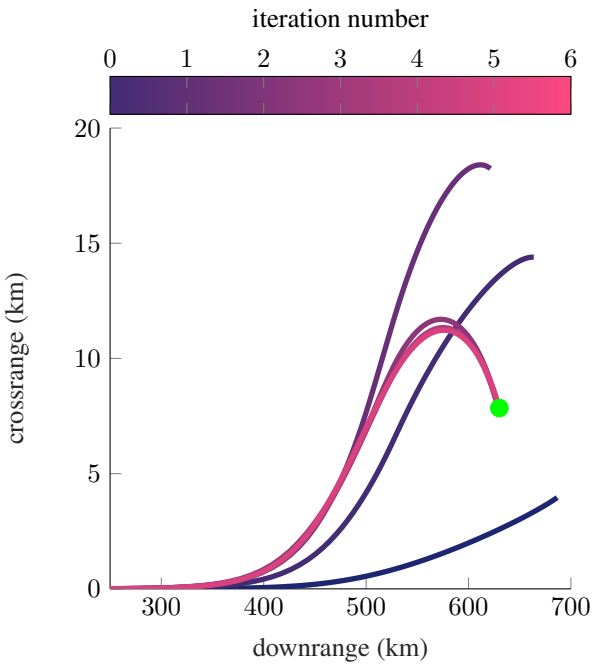

Figure 7. Predicted entry vehicle trajectories for the bank-angle only quadratic penalty case, as seen by the crossrange and downrange data.

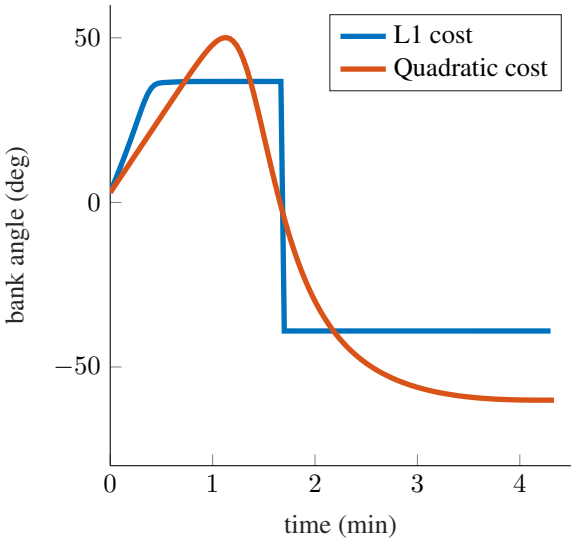

Figure 8. Bank-angle only control plans for both the L1 and quadratic cost cases. The L1 cost motivated a bang-bang switching style bank-angle profile. The quadratic cost resulted in a smooth and continuous bank-angle profile.

penalized the derivative with an L1 cost and a quadratic cost were both demonstrated, resulting in dramatically different optimal bank-angle profiles. For the case with both bank-angle and angle-of-attack control, the quality of the dynamics linearization was accurate enough that CPEG was able to converge on an optimal trajectory in just a few iterations. An implementation of CPEG running all of the examples in this paper is available at `https://github.com/RoboticExplorationLab/EntryGuidance.jl`.

Acknowledgments

This work was supported by an Early Career Faculty Award from NASA's Space Technology Research Grants Program (Grant Number 80NSSC21K1329).

References

[1] S. Li and X. Jiang, "Review and prospect of guidance and control for Mars atmospheric entry," *Progress in Aerospace Sciences*, vol. 69, pp. 40–57, Aug. 2014.

[2] C. Graves and A. Harpold, "Apollo experience report: Mission planning for Apollo entry," NASA Johnson Space Center, Houston, TX, Technical Report NASA-TN-D-6725, Mar. 1972.

[3] G. F. Mendeck and L. Craig McGrew, "Entry Guidance Design and Postflight Performance for 2011 Mars Science Laboratory Mission," *Journal of Spacecraft and Rockets*, vol. 51, no. 4, pp. 1094–1105, Jul. 2014.

[4] C. W. Brunner and P. Lu, "Comparison of Fully Numerical Predictor-Corrector and Apollo Skip Entry Guidance Algorithms," *The Journal of the Astronautical Sciences*, vol. 59, no. 3, pp. 517–540, Sep. 2012.

[5] P. Lu, "Predictor-Corrector Entry Guidance for Low-Lifting Vehicles," *Journal of Guidance, Control, and Dynamics*, vol. 31, no. 4, pp. 1067–1075, Jul. 2008.

[6] ——, "Entry Guidance: A Unified Method," *Journal of Guidance, Control, and Dynamics*, vol. 37, no. 3, pp. 713–728, May 2014.

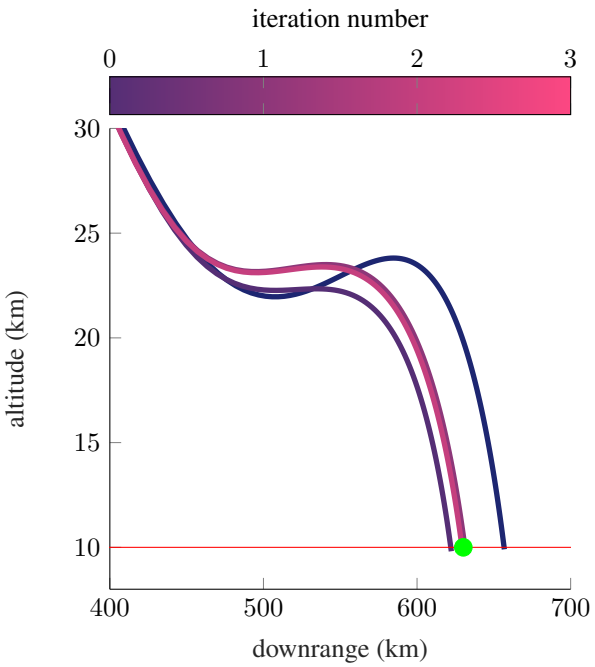

Figure 9. Predicted entry vehicle trajectories for the bank-angle and angle-of-attack case, as seen by the altitude and downrange data.

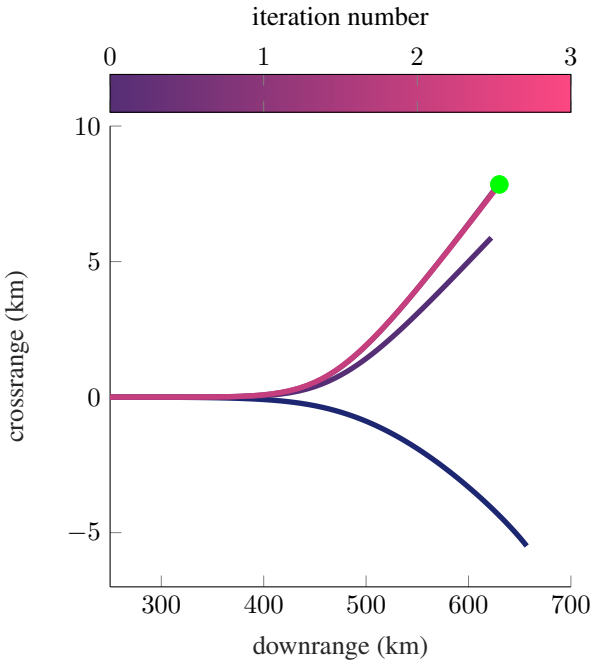

Figure 10. Predicted entry vehicle trajectories for the bank-angle and angle-of-attack case, as seen by the crossrange and downrange data.

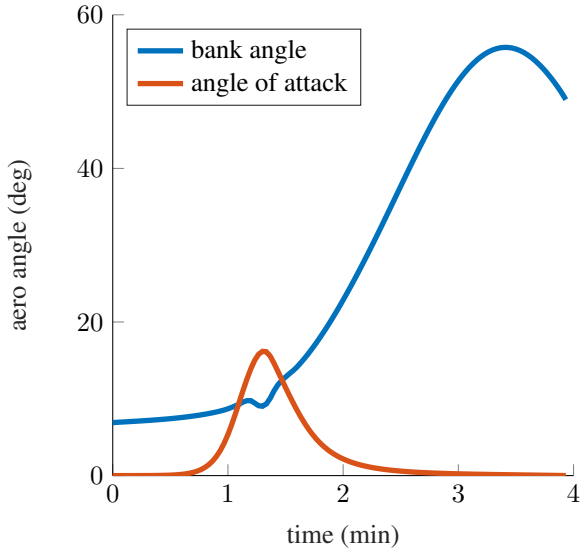

Figure 11. Bank-angle and angle-of-attack profiles for the case where both angles are being controlled. CPEG was able to converge on this control plan in just three iterations.

[7] Z. Wang and M. J. Grant, "Constrained Trajectory Optimization for Planetary Entry via Sequential Convex Programming," in *AIAA Atmospheric Flight Mechanics Conference*, Washington, D.C.: American Institute of Aeronautics and Astronautics, Jun. 2016.

[8] ——, "Near-Optimal Entry Guidance for Reference Trajectory Tracking via Convex Optimization," in *2018 AIAA Atmospheric Flight Mechanics Conference*, Kissimmee, Florida: American Institute of Aeronautics and Astronautics, Jan. 2018.

[9] P. E. Gill, W. Murray, and M. A. Saunders, "SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization," *SIAM Review*, vol. 47, no. 1, pp. 99–131, Jan. 2005.

[10] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, Mar. 2006.

[11] A. Busemann, N. Vinh, and R. Culp, "Hypersonic Flight Mechanics," Tech. Rep. NASA-CR-149170, Sep. 1976, p. 440.

[12] N. X. Vinh, A. Busemann, and R. D. Culp, "Hypersonic and planetary entry flight mechanics," *NASA STI/Recon Technical Report A*, vol. 81, p. 16 245, Jan. 1980.

[13] N. Vinh, W. Johnson, and J. Longuski, "Mars aerocapture using bank modulation," in *Astrodynamics Specialist Conference*, Denver,CO,U.S.A.: American Institute of Aeronautics and Astronautics, Aug. 2000.

[14] Z. Wang and M. J. Grant, "Near-Optimal Entry Guidance for Reference Trajectory Tracking via Convex Optimization," American Institute of Aeronautics and Astronautics, Jan. 2018.

[15] ——, "Improved Sequential Convex Programming Algorithms for Entry Trajectory Optimization," in *AIAA Scitech 2019 Forum*, San Diego, California: American Institute of Aeronautics and Astronautics, Jan. 2019.

[16] P. Gallais, *Atmospheric Re-Entry Vehicle Mechanics*. Berlin ; New York: Springer, 2007.

[17] L. Blackmore, B. Açikmeşe, and J. M. Carson, "Lossless convexification of control constraints for a class of nonlinear optimal control problems," in *2012 American Control Conference (ACC)*, Jun. 2012, pp. 5519–5525.

[18] B. Acikmese and S. R. Ploen, "Convex Programming Approach to Powered Descent Guidance for Mars Landing," *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 5, pp. 1353–1366, Sep. 2007.

[19] B. Acikmese, J. M. Carson, and L. Blackmore, "Lossless Convexification of Nonconvex Control Bound and Pointing Constraints of the Soft Landing Optimal Control Problem," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 6, pp. 2104–2113, Nov. 2013.

[20] K. T. Edquist, B. R. Hollis, A. A. Dyakonov, B. Laub, M. J. Wright, T. P. Rivellini, E. M. Slimko, and W. H. Willcockson, "Mars Science Laboratory Entry Capsule Aerothermodynamics and Thermal Protection System," in *2007 IEEE Aerospace Conference*, Big Sky, MT, USA: IEEE, 2007, pp. 1–13.

[21] O. Montenbruck, E. Gill, and F. Lutze, "Satellite Orbits: Models, Methods, and Applications," *Applied Mechanics Reviews*, vol. 55, no. 2, B27, 2002.

[22] T. A. Howell, B. E. Jackson, and Z. Manchester, "ALTRO: A Fast Solver for Constrained Trajectory Optimization," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Macau, China, Nov. 2019.

[23] B. E. Jackson, T. Punnoose, D. Neamati, K. Tracy, and R. Jitosho, "ALTRO-C: A Fast Solver for Conic Model-Predictive Control," in *International Conference on Robotics and Automation (ICRA)*, Xi'an, China, 2021, p. 8.

[24] D. Malyuta, T. P. Reynolds, M. Szmuk, T. Lew, R. Bonalli, M. Pavone, and B. Acikmese, "Convex Optimization for Trajectory Generation," *arXiv:2106.09125 [cs, eess, math]*, Jun. 2021.

[25] Y. Mao, M. Szmuk, X. Xu, and B. Acikmese, "Successive Convexification: A Superlinearly Convergent Algorithm for Non-convex Optimal Control Problems," *arXiv:1804.06539 [math]*, Feb. 2019.

[26] J. Nocedal and S. J. Wright, *Numerical Optimization*, Second. Springer, 2006.

[27] Mosek ApS, "The MOSEK optimization software," Tech. Rep., 2014.

[28] M. Garstka, M. Cannon, and P. Goulart, "COSMO: A conic operator splitting method for convex conic problems," *arXiv:1901.10887 [math]*, Sep. 2020.

[29] A. Domahidi, E. Chu, and S. Boyd, "ECOS: An SOCP solver for embedded systems," in *2013 European Control Conference (ECC)*, Zurich: IEEE, Jul. 2013, pp. 3071–3076.

[30] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah, "Julia: A Fresh Approach to Numerical Computing," *SIAM Review*, vol. 59, no. 1, pp. 65–98, Jan. 2017.

[31] M. Udell, K. Mohan, D. Zeng, J. Hong, S. Diamond, and S. Boyd, "Convex Optimization in Julia," in *2014 First Workshop for High Performance Technical Computing in Dynamic Languages*, LA, USA: IEEE, Nov. 2014, pp. 18–28.

[32] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd, "OSQP: An Operator Splitting Solver for Quadratic Programs," p. 40,

BIOGRAPHY

Kevin Tracy *is a graduate student with the Robotic Exploration Lab at Carnegie Mellon University. He received his BS in Mechanical Engineering from Rice University in 2018 and his MS in Mechanical Engineering from Stanford University in 2020. His research interests are optimization-based solutions for spacecraft guidance, navigation, and control.*

Zac Manchester *is an assistant professor in the Robotics Institute at Carnegie Mellon University and founder of the Robotic Exploration Lab. He received a PhD in aerospace engineering in 2015 and a BS in applied physics in 2009, both from Cornell University. His research interests include control and optimization with application to aerospace and robotic systems with challenging nonlinear dynamics.*