

# Robust Entry Guidance with Atmospheric Adaptation

Kevin Tracy\*, Giusy Falcone †, and Zachary Manchester ‡  
*Carnegie Mellon University, Pittsburgh, PA, 15213*

**Robust atmospheric entry guidance for blunt-body entry vehicles with bank angle modulation is achieved by combining online atmospheric density estimation with an updated version of the Convex Predictor-corrector Entry Guidance (CPEG) algorithm. During atmospheric entry, a square-root Extended Kalman Filter is used to estimate a ratio between the density of the experienced atmosphere with that of an approximate model, which is spline-fit based on MarsGRAM perturbed data. The information from this filter is used to modify the approximate model used by the guidance algorithm. The proposed update to CPEG includes time as a decision variable, dramatically improving the robustness of the algorithm. CPEG predicts the trajectory at each control call with a nonlinear simulation followed by a single convex trajectory optimization problem that updates the commanded bank angle derivative. The robustness and performance of this estimator and controller guidance architecture are demonstrated on a wide range of realistic Martian atmospheres and is able to achieve state-of-the-art accuracy with respect to altitude-triggered parachute deployment.**

## Nomenclature

$A$	=	entry vehicle reference area, $\text{m}^2$
$a_D$	=	acceleration due to aerodynamic drag, $\text{m/s}^2$
$a_g$	=	acceleration due to gravity, $\text{m/s}^2$
$a_L$	=	acceleration due to aerodynamic lift, $\text{m/s}^2$
$D$	=	magnitude of drag acceleration, $\text{m/s}^2$
$\hat{e}$	=	aerodynamic basis vector
$F$	=	matrix square-root of the estimate covariance matrix
$k_\rho$	=	ratio of observed to approximate atmospheric densities
$L$	=	magnitude of lift acceleration, $\text{m/s}^2$
$m$	=	entry vehicle mass, $\text{kg}$
$r$	=	position of the entry vehicle in a Mars-fixed frame, $\text{m}$
$v$	=	velocity of the entry vehicle in a Mars-fixed frame, $\text{m/s}$
$V$	=	sensor noise covariance
$w$	=	wind vector in a Mars-fixed frame, $\text{m/s}$
$W$	=	process noise covariance
$\mu$	=	state estimate
$\omega$	=	angular velocity of Mars, $\text{rad/s}$
$\rho$	=	atmospheric density, $\text{kg/m}^3$
$\sigma$	=	entry vehicle bank angle, $\text{rad}$
$\Sigma$	=	estimate covariance matrix

## I. Introduction

Entry into the Martian atmosphere refers to the phase of Entry, Descent, and Landing (EDL) that occurs between the point at which the hypersonic entry vehicle first interfaces with the planet's sensible atmosphere and the point at which the supersonic parachute is deployed. It is during this portion of the EDL that the vehicle is subjected to the most demanding flight conditions, during which both peak acceleration and peak heating are experienced [1]. Large

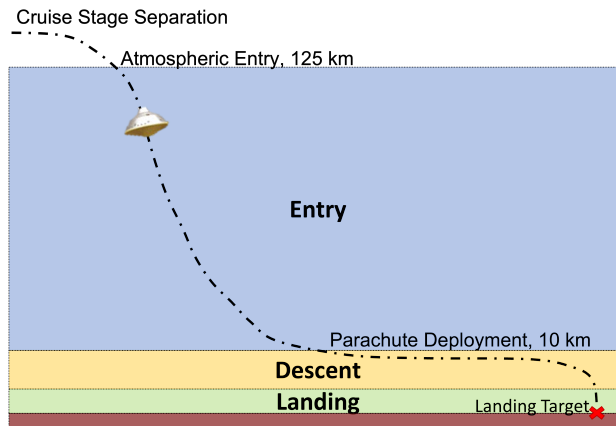
---

\*Graduate Student, The Robotics Institute, 5000 Forbes Ave, ktracy@cmu.edu

†Postdoctoral Fellow, The Robotics Institute, 5000 Forbes Ave, gfalcone@andrew.cmu.edu

‡Assistant Professor, The Robotics Institute, 5000 Forbes Ave, zacm@cmu.edu

uncertainties in models of the Martian atmospheric environment make entry guidance challenging, and significantly degrade landing accuracy, where the landing accuracy is the distance between the landing target site and the actual landing site. Actual atmospheric density can deviate from predictions by a factor of two or more [2], requiring guidance laws to combat uncertainty with frequent re-planning and reliance on conservative initial entry conditions. These approaches ultimately reduce performance and limit the potential landing sites that can be accessed on Mars [3]. To enable precision landing, which could potentially increase the number of reachable sites of high scientific interest, we propose a new approach to entry guidance that can strategically account for and reduce the atmospheric model uncertainty during guided entry. In the last few decades, predictor-corrector guidance algorithms have become the



**Fig. 1** Cartoon representation of entry, descent, and landing.

state-of-the-art; these have offered steadily improving landing accuracy with modest computational costs. These methods, like PredGuid [4, 5] and FNPEG [6], predict the trajectory of the entry vehicle with a nonlinear simulation, then use derivative or “sensitivity” information from the simulation to improve or “correct” the control plan [7, 8]. This predictor-corrector framework offers improved landing accuracy compared to Apollo-era entry guidance methods [9] while being relatively simple to implement with low computational complexity. However, a significant drawback of traditional predictor-corrector guidance methods is their inability to generate complex control policies; many such methods are restricted to solving for a static bank angle, which means that the initial bank angle control profile predicted for the whole entry phase is considered constant; however, only during the prediction and correction phase, the guidance uses independent downrange and crossrange control to determine when the sign of the bank angle should flip. This simplicity severely limits the sophistication of the guidance method, and the resulting performance can chatter with multiple aggressive bank angle switches. Another limitation of existing predictor-corrector guidance strategies is their inability to incorporate real-time information about the Martian atmosphere, which is the largest source of uncertainty in the vehicle’s dynamics, for prediction and correction purposes.

Recently, there has been a growing interest in trajectory optimization methods for open-loop entry guidance for both Earth and Mars. This approach, in which the guidance problem is formulated as a numerical optimization problem, enables algorithms that can reason about the dynamics of the vehicle and constraints like actuator limits, state bounds, and heating limits. These optimization problems can be solved by a variety of numerical methods. As an example, sequential convex programming, in which a non-convex optimization problem is iteratively convexified and solved until convergence, has recently become popular [10–15]. Alternatively, the trajectory optimization problem can be converted into a nonlinear program [16] and solved by a variety of off-the-shelf solvers, like SNOPT [17] or Ipopt [18], or by a more specialized trajectory optimization solver like ALTRO [19, 20]. While these trajectory optimization guidance methods offer more sophisticated and performant control plans than classical predictor-corrector methods, they can be prohibitively complex to implement on resource-constrained flight computers, and are still unable to account for real-time information about the atmosphere.

A middle ground between simple predictor-corrector guidance schemes and offline trajectory optimization is the Convex Predictor-corrector Entry Guidance algorithm (CPEG) [21]. This method uses the same predictor-corrector framework as existing methods, but instead of solving for a simple static bank angle, it forms a convex trajectory optimization problem that solves for the correction to the nominal control plan. This approach enables the direct inclusion of any actuator or safety constraints and the ability to reason about complex control policies and the dynamics

of the vehicle over the whole trajectory. By forming the convex trajectory optimization problem with a dynamics model linearized about the prediction rollout, the problem is guaranteed to always be feasible, and the convexity of the problem guarantees that an optimal solution can be computed in polynomial time [22].

This work proposes an improvement to the CPEG algorithm that includes time as an optimization variable, allowing for the generation of trajectories with a free final time. This modification allows for significantly more flexibility when handling scenarios with multiple potentially successful control policies, since many trajectories can all be expressed using the same number of optimization variables. The absence of this feature in the original CPEG algorithm resulted in a lack of robustness when run online with significant model uncertainty. Additionally, this work directly addresses the main source of the uncertainty in the vehicle dynamics – the atmospheric density – by estimating information about the atmosphere online and including it in the approximate model used by CPEG. Our specific contributions include:

- 1) A free-final-time extension of the Convex Predictor-corrector Entry Guidance algorithm
- 2) An estimation architecture for atmospheric density and an atmospheric approximate model
- 3) Validation of the combined controller and estimator on realistic density and wind dispersions from MarsGRAM [23]

This paper proceeds by first describing the dynamics of the entry vehicle in Cartesian coordinates in Section II, followed by the free-final-time version of CPEG in Section III. The atmospheric density estimation approach is detailed in Section IV. Numerical experiments using Monte-Carlo sampling of atmospheric parameters from MarsGRAM are presented in Section V. Finally, Section VI summarizes our conclusion.

## II. Entry Vehicle Dynamics

This work considers a blunt-body entry vehicle in the Martian atmosphere with the ability to modulate its bank angle  $\sigma \in \mathbb{R}$ . These entry vehicles are traditionally described and simulated in spherical coordinates using the “Vinh” model [6, 11, 24–28]. While this model is ubiquitous and well-studied, it suffers from numerical scaling issues due to the state vector including angles as well as distance and velocity, and is highly nonlinear in both the state and the control input. These shortcomings make optimization challenging and numerically unreliable. Alternatively, the state and dynamics of the entry vehicle can be equivalently described in a Mars-fixed Cartesian reference frame. This parametrization is more common in the powered-descent guidance literature [29–31], and has much better numerical scaling, as well as linear kinematics, both of which make it a better fit for optimization-based control methods as detailed in [32]. The dynamics of the entry vehicle with this state representation are the following:

$$\dot{r} = v, \quad (1)$$

$$\dot{v} = a_L + a_D + a_g - 2(\omega \times v) - \omega \times (\omega \times r). \quad (2)$$

The aerodynamic accelerations from lift and drag are dependent on the vehicle velocity relative to the atmosphere, including wind. The wind vector  $w$  is expressed in the Mars-fixed frame, and the relative velocity is simply  $v_{rel} = v + w$ . The magnitude of the lift and drag accelerations can then be computed as follows:

$$L = \frac{1}{2m} \rho A C_L \|v_{rel}\|^2, \quad (3)$$

$$D = \frac{1}{2m} \rho A C_D \|v_{rel}\|^2. \quad (4)$$

The hypersonic coefficients,  $C_L$  and  $C_D$  have been evaluated through the Newtonian flow theory for a blunted body, specifically:

$$C_A = (1 - \sin^4 \delta) \frac{r_{nose}^2}{r_{base}^2} + (2 \sin^2 \delta \cos^2 \alpha + \cos^2 \delta \sin^2 \alpha) \left( 1 - \frac{r_{nose}^2}{r_{base}^2} \cos^2 \delta \right), \quad (5)$$

$$C_N = \left( 1 - \frac{r_{nose}^2}{r_{base}^2} \cos^2 \delta \right) \cos \delta^2 \sin 2\alpha, \quad (6)$$

$$C_L = C_N \cos \alpha - C_A \sin \alpha, \quad (7)$$

$$C_D = C_A \cos \alpha + C_N \sin \alpha, \quad (8)$$

where  $C_A$  is the axial force coefficient,  $C_N$  is the normal force coefficient,  $\delta$  is the cone angle of the blunted-cone vehicle,  $\alpha$  is the angle of attack,  $r_{base}$  is the base radius,  $r_{nose}$  is the nose radius of the hypersonic vehicle.

The direction of the drag acceleration is in the opposite direction of the relative velocity, expressed as:

$$a_D = -D \frac{v_{rel}}{\|v_{rel}\|}. \quad (9)$$

The lift acceleration is directed using the bank angle  $\sigma$ , and is modeled and represented with two basis vectors  $\hat{e}_1 \in \mathbb{R}^3$  and  $\hat{e}_2 \in \mathbb{R}^3$ , where together they span the plane orthogonal to the relative velocity of the vehicle in the atmosphere. These aerodynamic basis vectors are constructed as the following:

$$\hat{e}_1 = \frac{r \times v_{rel}}{\|r \times v_{rel}\|}, \quad (10)$$

$$\hat{e}_2 = \frac{v_{rel} \times \hat{e}_1}{\|v_{rel} \times \hat{e}_1\|}. \quad (11)$$

Using these basis vectors, the lift acceleration can be computed as the following:

$$a_L = L(\sin(\sigma)\hat{e}_1 + \cos(\sigma)\hat{e}_2). \quad (12)$$

The acceleration due to gravity can be calculated using any number of established methods ranging in fidelity from simple two-body gravity [33], to a more complex spherical harmonic expansion of the geopotential [34, 35].

The largest amount of uncertainty in the dynamics of the entry vehicle comes from the atmosphere. Both the atmospheric density  $\rho$ , as well as the wind vector  $w$  are highly uncertain. To visualize this uncertainty, Mars GRAM [2] was used to generate dispersions for these two parameters, which are shown in Fig. 2 and Fig. 3. A summary of the parameters used to run Mars GRAM is presented in Table 1. For the generation of atmospheric data points in highly perturbed environments, Mars GRAM not only uses random Monte Carlo samples that re-initialize the random number seed (NR1) for each sample but also has been set for different zoffsets, which are constant height offsets that modify the atmospheric density. Specifically, positive offsets increase density, while negative offsets decrease density. In addition, rpscale, the random perturbation scale was also set to the maximum, with increasing this factor intensifying the magnitude of the perturbation. The data have also been evaluated in the context of a global dust storm, through INTENS, which changes the dust storm intensity.

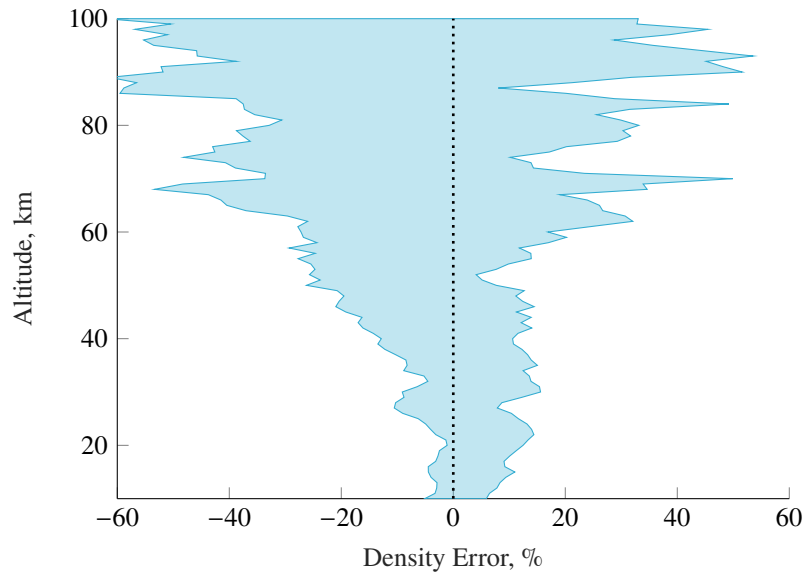
**Table 1 Mars GRAM nominal and dispersion parameters**

Parameter	Value	Parameter	Value
Initial Altitude, km	125	NR1	[1,5001]
Date	6 August 2012	zoffset, km	Uniform Dist., [-3.25,3.25]
Time	5:30:00	rpscale	2
Final Altitude, km	10	INTENS	2

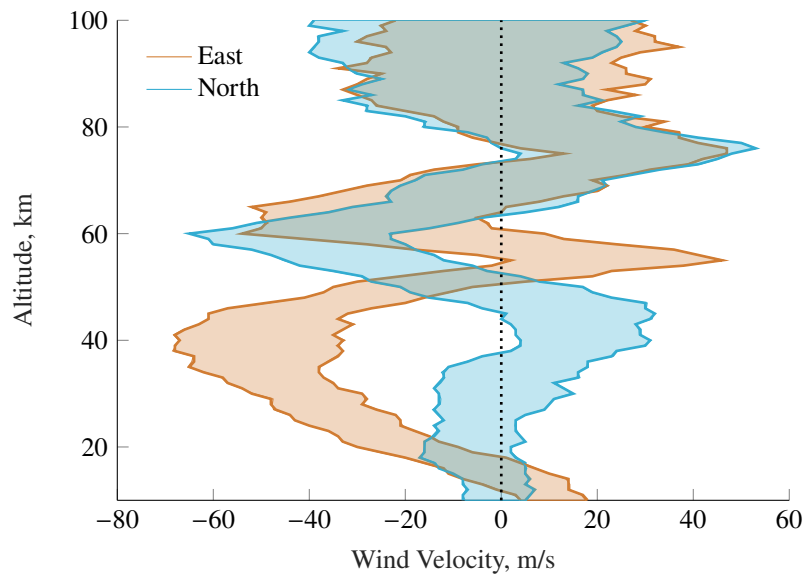
Since aerodynamic acceleration is the dominant term in the dynamics at the high speeds an entry vehicle experiences, the uncertainty in the atmospheric density results in a highly uncertain dynamics model. The uncertainty in the wind contributes to the model uncertainty, but less so than the density.

### III. CPEG

Predictor-corrector guidance algorithms are a simple and effective way to control underactuated systems like the entry vehicle. These methods all share a general framework in which a nominal control policy is used to simulate a predicted trajectory, then a correction is made to the nominal control policy based on this prediction. Predictor-corrector methods have been the state of the art for entry guidance of blunt-body vehicles, but they are not without their shortcomings. Existing methods like FNPEG are only able to reason about simple static bank angle policies, and rely on independent crossrange and downrange logic to modulate the bank angle [7]. Alternatively, offline trajectory optimization allows for more complex control policies and the direct inclusion of constraints, but is significantly more expensive than predictor-corrector methods. A trajectory optimization problem in its most general form is the following,



**Fig. 2** Dispersion of the atmospheric density as a function of altitude calculated with Mars GRAM. The dynamics of the entry vehicle are heavily influenced by this density, and the dramatic uncertainty of this value motivates the need for more robust guidance schemes.



**Fig. 3** Dispersions of the east and north wind velocities in the Martian atmosphere as a function of altitude calculated with Mars GRAM. This range of atmospheric wind profiles is used in the Monte Carlo testing of the proposed entry guidance algorithm.

$$\begin{aligned}
& \underset{x_{1:N}, u_{1:N-1}}{\text{minimize}} && \ell_N(x_N) + \sum_{k=1}^{N-1} \ell_k(x_k, u_k) \\
& \text{subject to} && x_{k+1} = f(x_k, u_k) \quad k = 1 \dots N-1, \\
& && g_k(x_k, u_k) \leq 0 \quad k = 1 \dots N-1, \\
& && u_{min} \leq u_k \leq u_{max} \quad k = 1 \dots N-1, \\
& && x_{min} \leq x_k \leq x_{max} \quad k = 1 \dots N
\end{aligned} \tag{13}$$

where  $x$  is the state,  $u$  is the control,  $\ell$  is the cost function,  $f(x, u)$  is the discrete dynamics function, and  $g(x, u)$  is a general constraint function. Problem (13) can be solved with a multitude of available trajectory optimization solvers, but they are prohibitively complex to run on board the entry vehicle.

### A. Baseline CPEG

The Convex Predictor-corrector Entry Guidance (CPEG) algorithm, introduced in [32], seeks to find a middle ground between simple predictor-corrector guidance methods and offline trajectory optimization. This is achieved by adopting the predictor-corrector framework, but solving a convex trajectory optimization problem for the correction instead of a simple update to a static bank angle like in FNPEG. This approach allows CPEG to reason about sophisticated control trajectories as well as the dynamics of the vehicle throughout the entirety of the trajectory. The convex optimization problem that makes up the correction portion of the algorithm is guaranteed to be feasible, and its optimum can be solved for at real-time rates [36].

In order for problem (13) to be convex, the cost and constraint functions must be convex. This means that problems with nonlinear dynamics, like the entry vehicle, must be approximated. We linearize the discrete dynamics function  $f(x, u)$  about the predicted trajectory  $(\bar{x}, \bar{u})$ :

$$\bar{x}_{k+1} + \delta x_{k+1} \approx f(\bar{x}_k, \bar{u}_k) + A_k \delta x_k + B_k \delta u_k, \tag{14}$$

where  $A_k$  and  $B_k$  are the following Jacobians,

$$A_k = \left. \frac{\partial f(x_k, u_k)}{\partial x_k} \right|_{\bar{x}_k, \bar{u}_k}, \tag{15}$$

$$B_k = \left. \frac{\partial f(x_k, u_k)}{\partial u_k} \right|_{\bar{x}_k, \bar{u}_k}. \tag{16}$$

Since the prediction was dynamically feasible, and  $\bar{x}_{k+1} = f(\bar{x}_k, \bar{u}_k)$ , equation (14) is reduced to

$$\delta x_{k+1} \approx A_k \delta x_k + B_k \delta u_k. \tag{17}$$

From here, the convex correction problem is posed as,

$$\begin{aligned}
& \underset{\delta x_{1:N}, \delta u_{1:N-1}}{\text{minimize}} && \ell_N(\bar{x}_N + \delta x_N) + \sum_{k=1}^{N-1} \ell_k(\bar{x}_k + \delta x_k, \bar{u}_k + \delta u_k) \\
& \text{subject to} && \delta x_{k+1} = A_k \delta x_k + B_k \delta u_k \quad k = 1 \dots N-1, \\
& && g_k(\bar{x}_k + \delta x_k, \bar{u}_k + \delta u_k) \leq 0 \quad k = 1 \dots N-1, \\
& && u_{min} \leq u_k \leq u_{max} \quad k = 1 \dots N-1, \\
& && x_{min} \leq x_k \leq x_{max} \quad k = 1 \dots N, \\
& && \delta u_{min} \leq \delta u_k \leq \delta u_{max} \quad k = 1 \dots N-1, \\
& && \delta x_{min} \leq \delta x_k \leq \delta x_{max} \quad k = 1 \dots N,
\end{aligned} \tag{18}$$

where the bounds on  $\delta x$  and  $\delta u$  serve as a customizable trust region that ensures the step direction stays within a region of the state space where the linearization is accurate [37]. Since the linearization is evaluated about the dynamically

feasible predicted trajectory, this optimization problem is guaranteed to have a feasible solution of all zeros for  $\delta x$  and  $\delta u$ , eliminating one of the largest vulnerabilities in real-time optimization-based control.

The full baseline CPEG algorithm is shown in Algorithm 1, where an initial condition and nominal control policy are used to simulate the entry vehicle until parachute deployment (the prediction), a linearized model is evaluated about this predicted trajectory, and a convex trajectory optimization problem (18) is solved for the correction.

---

**Algorithm 1** CPEG Algorithm

---

- |    |   |                               |
|----|---|-------------------------------|
| 1: | <b>function</b> CEPG( $x_0, U$ )                          | ▷ nominal control plan        |
| 2: | $\bar{X}, \bar{U} = \text{simulate}(x_0, U)$              | ▷ predict trajectory          |
| 3: | $A, B = \text{linearize}(\bar{X}, \bar{U})$               | ▷ linearize about prediction  |
| 4: | $\delta X, \delta U = \text{cvx}(\bar{X}, \bar{U}, A, B)$ | ▷ solve for correction        |
| 5: | $U += \delta U$   | ▷ correct control plan        |
| 6: | <b>return</b> $U$   | ▷ return updated control plan |
- 

## B. Variable-Time CPEG

We now extend CPEG to handle free-final-time problems where time is a decision variable. In the baseline algorithm, the prediction is done with a fixed time step and the simulation simply terminates when a parachute deployment trigger is satisfied. This means that the corrected control policy can result in a subsequent prediction with a different number of time steps. This characteristic of the baseline algorithm results in decreased robustness when there are multiple cost-equivalent trajectories in the cost landscape. In this case, CPEG can chatter between two or more control policies that are of different lengths but have equal cost.

To address this issue, an updated version of CPEG is proposed in this work that treats time as an explicit optimization variable, allowing for the time of parachute deployment to effectively be decided by the optimization solver, instead of solely by the prediction step. This updated version of CPEG is significantly more robust to chattering. The free-final-time modification can be incorporated into the existing CPEG framework by simply including the time step for each knot point as a control input. The resulting state variable  $x \in \mathbb{R}^7$  and control variable  $u \in \mathbb{R}^2$  are the following:

$$x = [r^T, v^T, \sigma]^T, \tag{19}$$

$$u = [\dot{\sigma}, \Delta t]^T, \tag{20}$$

where  $\Delta t \in \mathbb{R}$  is the size of the time step. We use the following cost function:

$$J(x, u) = \gamma \|r_N - r_{\text{target}}\|^2 + \sum_{k=1}^{N-1} \beta \dot{\sigma}_k^2 + (\Delta t_k - \Delta t_{\text{target}})^2, \tag{21}$$

where  $\gamma \in \mathbb{R}_+$  and  $\beta \in \mathbb{R}_+$  are positive cost weights,  $r_{\text{target}} \in \mathbb{R}^3$  is the target position, and  $\Delta t_{\text{target}} \in \mathbb{R}_+$  is target time-step size. These weights should be tuned such that the largest penalty is applied to the terminal position accuracy, and a sufficient penalty exists on the time-step size to keep the time steps positive and reasonable.

Each control correction requires the solution of a convex Quadratic Program (QP), of which there are many fast and robust solution methods available. This work utilized a custom implementation of the primal-dual interior point solver detailed in [36] and [38]. At the start of the QP solver, the inequality constraints are temporarily omitted and the closed-form solution to the equality-constrained QP is checked for feasibility. In many cases, the inequality constraints are inactive at the optimum, and this approach is able to solve the QP at the cost of solving a single linear system.

## IV. Atmospheric Estimation

During atmospheric entry guidance, the largest source of uncertainty in the dynamics comes from the atmosphere. The density of the atmosphere is highly variable based on the time of the year, weather, dust conditions, temperature, and space weather. Mars GRAM [2] is the highest fidelity tool available for the simulation of the Martian atmosphere, and it was used to sample 5,000 potential atmospheric density profiles. 1,000 of these profiles are shown in Fig. 2, where there is significant uncertainty at higher altitudes, with better certainty at lower altitudes, though still varying by 10 – 20%. Because of this uncertainty, open-loop control policies perform poorly.

To manage this uncertainty, we propose a simple and effective method for estimating critical information about the atmospheric density online during the entry process. The density of the Martian atmosphere roughly follows an exponential law, such that the difference in density between the surface and 125 km altitude, which is considered in this study as the entry interface, spans several orders of magnitude. Because of this large variation in magnitude, estimating the density directly in a Kalman Filter is numerically challenging. Another possible approach to tackle atmospheric uncertainties would be to directly model the atmosphere as an exponential and estimate the surface density and scale height [39], but this approach suffers from severe ill-conditioning because the exponential dramatically amplifies errors at higher altitudes [40, 41].

In place of estimating some parametrized version of the atmospheric density profile from scratch, we use an approach similar to [42] where a Kalman filter for estimating the ratio between the observed and nominal atmospheres:

$$k_\rho = \frac{\rho_{\text{observed}}}{\rho_{\text{nominal}}}, \quad (22)$$

where  $\rho_{\text{nominal}}$  is computed offline and stored. Estimating  $k_\rho$  directly is better behaved numerically because the estimator is simply looking to modify an existing approximate density profile, and the estimated value will always be near 1, regardless of altitude. The state and control for this filter are the following:

$$x_{\text{kf}} = [r^T, v^T, \sigma, k_\rho]^T, \quad (23)$$

$$u_{\text{kf}} = \dot{\sigma}, \quad (24)$$

where the dynamics for this state representation are the same as in section II, with the exception that  $\rho = k_\rho \cdot \rho_{\text{approximate}}$ . The measurement model is assumed to come directly from the navigation system, with additive Gaussian white noise. To estimate  $k_\rho$  effectively in the presence of highly variable dynamic pressure, a Square-Root Extended Kalman Filter (SQEKF) is used that allows for twice the dynamic range of the standard Extended Kalman Filter (EKF) [43]. This increased range and numerical robustness help to manage the potentially ill-conditioned covariance matrices present during entry guidance. A simple version of an SQEKF is demonstrated in [44] that leverages a highly mature QR decomposition routine to handle a majority of the computation [45].

The SQEKF takes a system of the following form:

$$x_{k+1} = f(x_k, u_k) + w_k, \quad (25)$$

$$y_k = g(x_k) + v_k, \quad (26)$$

where  $f(x, u)$  is the discrete time dynamics function,  $g(x)$  is the measurement function, and  $w \sim \mathcal{N}(0, W)$  and  $v \sim \mathcal{N}(0, V)$  are the process and sensor noise terms, respectively. The SQEKF is able to achieve better precision than the standard EKF by representing covariance matrices by their upper-triangular matrix square roots (Cholesky factors) [46–48]. Using this notation, the SQEKF algorithm from [44] is shown in Algorithm , where  $\Gamma_W$  and  $\Gamma_V$  are the upper triangular Cholesky factorizations of  $W$  and  $V$ , and the function  $\text{qr}_\tau$  returns the upper triangular  $R$  factor from a QR decomposition [19, 45]. The mean of the estimate is  $\mu_{t|t}$ , and the covariance  $\Sigma$  is represented with its matrix square root  $F$  such that  $F^T F = \Sigma$ . The performance of this filter on a realistic atmosphere sample from Mars GRAM is shown in Fig. 4.

## V. Numerical Experiments

To validate the combined controller and estimator architecture, 1,000 Monte Carlo runs were used to test the robustness and performance of the algorithms. Each Monte Carlo run consists of an atmospheric density and wind sampled from Mars GRAM, as shown in Fig. 2 and Fig. 3. An initial position dispersion with a standard deviation of 1 km and initial bank angle dispersion of 5 degrees was used to further disambiguate between runs. The initial conditions coincide with the Mars Sample Return (MSL) initial conditions; specifically, the initial position had a mean altitude of 125 km, with a flight path angle of -15.5 degrees and entry velocity of 5.85 km/s [49, 50]. The target was 632 km down range and 7.9 km cross range, with a parachute deployment triggered at 10 km altitude. Furthermore, the hypersonic vehicle considered has the same characteristics of the MSL vehicle; specifically, the vehicle is a 70 degrees sphere-cone, nose radius equal to 1.125 m, a base radius equal to 2.25 m, and an entry mass of 2800 kg [49].

The variable-time CPEG algorithm was run with a nominal time step size of 2 seconds until the length of the trajectory was less than 50 knot points, at which point the nominal time step was lowered to 0.1 seconds. This algorithm was run at 5 Hz with one convex correction solution at each time step, 94% of which were solved in a single iteration of



---

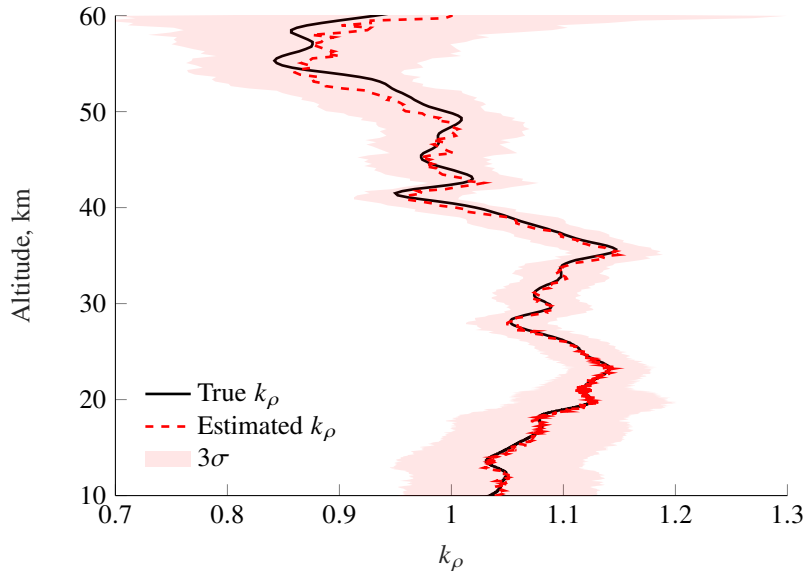
**Algorithm 2** Square Root Extended Kalman Filter
 

---

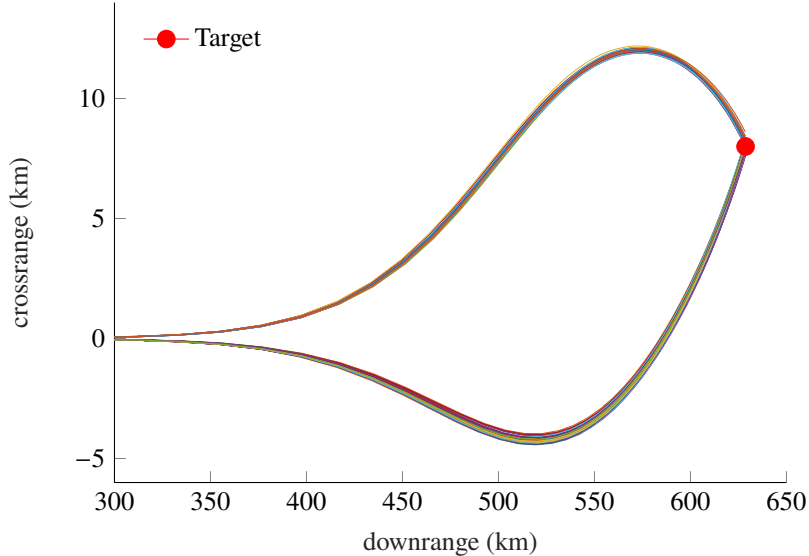
```

1: function sqkf( $\mu_{t|t}, F_{t|t}, u_t, y_{t+1}, \Gamma_W, \Gamma_V$ )
2:
3:    $A_t = \partial f / \partial x |_{\mu_t, u_t}$  ▷ dynamics Jacobian
4:    $\mu_{t+1|t} = f(\mu_{t|t}, u_t)$  ▷ state prediction
5:    $F_{t+1|t} = \text{qr}_r(F_{t|t} A_t^T, \Gamma_W)$  ▷ covariance prediction
6:
7:    $C_t = \partial g / \partial x |_{\mu_{t+1|t}}$  ▷ measurement Jacobian
8:    $z = y_{t+1} - g(\mu_{t+1|t})$  ▷ measurement innovation
9:    $G = \text{qr}_r(F_{t+1|t} C_t^T, \Gamma_V)$  ▷ innovation covariance
10:   $L = [G^{-1} (G^{-T} C_t) F_{t+1|t}^T F_{t+1|t}]^T$  ▷ Kalman gain
11:
12:   $\mu_{t+1|t+1} = \mu_{t+1|t} + Lz$  ▷ state update
13:   $F_{t+1|t+1} = \text{qr}_r(F_{t+1|t} (I - LC_t)^T, \Gamma_V L^T)$  ▷ covariance update
14:
15:  return( $\mu_{t+1|t+1}, F_{t+1|t+1}$ )
  
```

---



**Fig. 4** Results from the atmospheric density estimator during entry guidance where the ratio between the actual and nominal densities,  $k_\rho$ , is shown alongside the true value. There is large uncertainty in the higher altitudes where the atmosphere is very thin, and again at lower altitudes where the velocity of the vehicle is such that aerodynamic forces aren't as dominant in the dynamics.



**Fig. 5 Downrange and crossrange distances from the point of atmospheric interface for 1,000 Monte Carlo runs with variable-time CPEG and atmospheric estimation. Depending on the initial conditions and the atmosphere, not all the trajectories take the same route.**

the QP solver. A trust region was used to keep the updated trajectory in an area of the state space where the linearized dynamics model is still accurate, with  $|\delta\sigma| \leq 20^\circ$  and  $|\delta\Delta t| \leq 0.1$ . The SQEKF was initialized at an altitude of 60 km, and the estimated  $k_\rho$  value was used in the dynamics model in CPEG. The measurement model in the SQEKF assumed full state observations from the navigation system, with conservative measurement error standard deviations of 100 meters for the position, and 20 cm/s for the velocity.

All 1,000 Monte Carlo runs were successful in deploying the parachute within 1km of the target, with the trajectories shown in Fig. 5 and Fig. 6. Depending on the specific atmosphere and initial perturbation, there were two families of trajectories that the entry vehicle traversed. The bank-angle profiles for these trajectories are shown in Fig. 8. The terminal errors for these runs are shown in Fig. 7, with each individual run shown as well as a  $3\sigma$  ellipse.

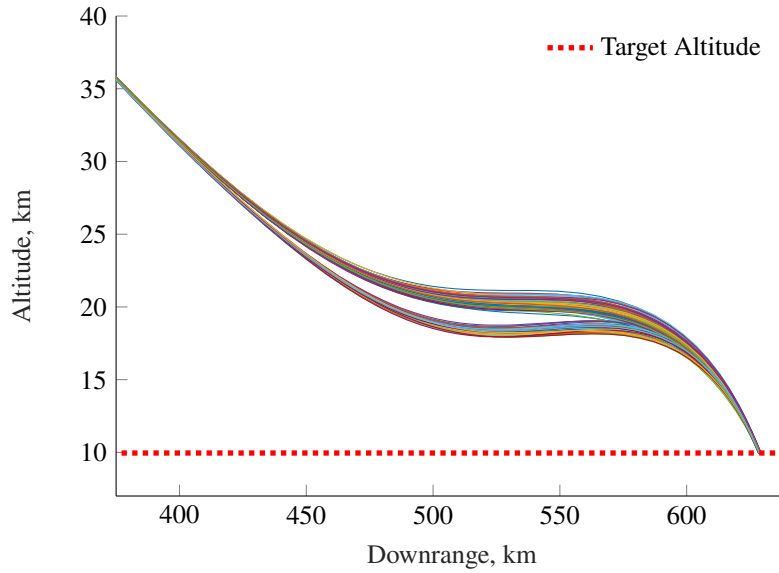
To demonstrate the effectiveness of the atmospheric adaptation, the same 1,000 Monte Carlo runs were executed with and without adaptation and the errors are shown in Fig. 10. The terminal errors for the runs with adaptation have significantly lower errors than the same runs without adaptation. The statistics from these terminal errors are shown in Table 2, showing a 37% decrease in the mean error, a 33% reduction in median error, and a 49% smaller maximum error with the atmospheric adaptation turned on.

**Table 2 Comparison of the parachute deployment error for variable time CPEG with and without the atmospheric adaptation. With the adaptation on, the mean, median, and maximum errors are all reduced.**

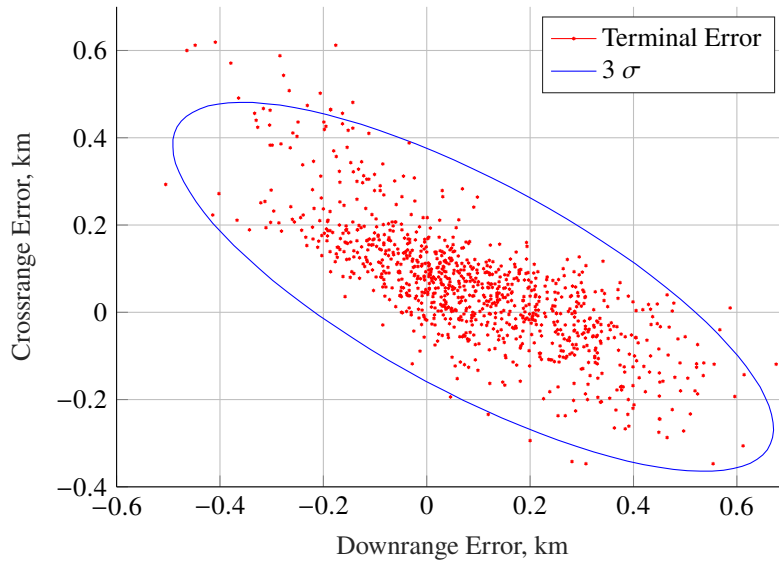
	Mean Error	Median Error	Maximum Error
Adaptation Off	0.353 km	0.287 km	1.485 km
Adaptation On	<b>0.223 km</b>	<b>0.193 km</b>	<b>0.758 km</b>

## VI. Conclusions

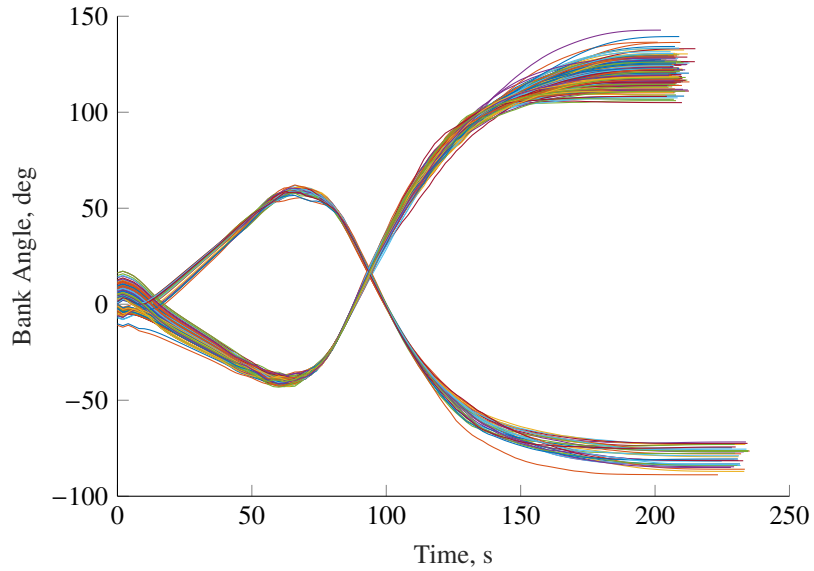
This work presents a combined estimator and controller architecture for atmospheric entry guidance in the Martian atmosphere. Atmospheric adaptation is accomplished with a square-root extended Kalman Filter that estimates the ratio of observed atmosphere to a nominal model computed offline, and uses this estimated ratio to modify the atmospheric model used in the controller. The guidance portion was accomplished using a variant of the convex predictor-corrector entry guidance algorithm (CPEG) that has been modified to include time as an optimization variable. This guidance algorithm is based on the predictor-corrector framework in which a nonlinear simulation is performed from the current



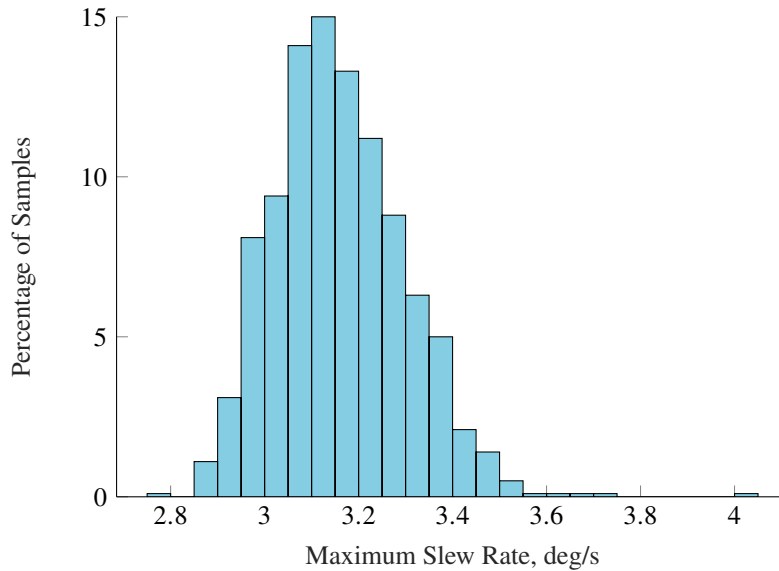
**Fig. 6** Altitude and downrange history for 1,000 Monte Carlo runs that terminate at an altitude of 10km. Based on the initial perturbation and atmosphere, there are two common paths that the entry vehicle takes to get to the target, one maintaining a higher altitude than the other.



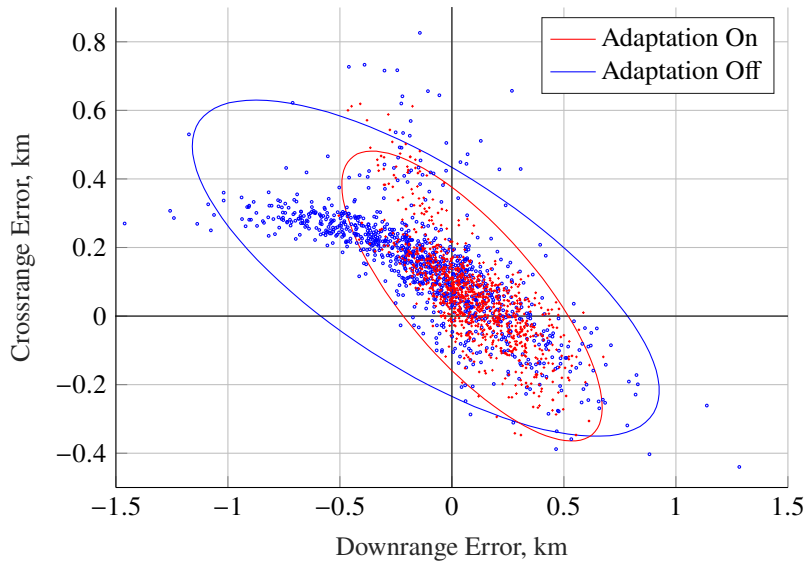
**Fig. 7** Terminal errors for parachute deployment for the 1,000 Monte Carlo runs as shown in downrange and crossrange errors. Each terminal error is shown, as well as an ellipse denoting the  $3\sigma$  bounds.



**Fig. 8** Bank-angle profiles for the 1,000 Monte Carlo runs. The initial bank angle varies as it was part of the dispersion, and depending on this initial condition and the atmosphere sampled during the run, one of two families of control approaches was taken.



**Fig. 9** Maximum slew rate for each of the 1,000 Monte Carlo runs. These maximum slew rates are tightly coupled around 3.2 deg/s, well below the 20 deg/s constraint.



**Fig. 10 Comparison of the terminal position errors from 1,000 Monte Carlo runs where atmospheric adaptation was switched on and off. The spread of the terminal errors is significantly tighter and closer to the origin for the cases with adaptation on.**

initial condition and nominal control policy, and information from this prediction is used to correct the nominal control policy. By solving for this correction using convex trajectory optimization, CPEG is able to explicitly reason about the dynamics of the vehicle through the entirety of the trajectory and formulate a sophisticated control plan.

The estimator and controller were validated with a 1,000 Monte Carlo simulations in which atmospheric density and wind profiles were sampled from Mars GRAM and the initial conditions of the entry vehicle were varied. With realistic noise levels, the median miss distance in these 1,000 runs was 193 meters, compared with a median miss distance of 287 meters with the atmospheric adaptation turned off. In 94% of the control calls in these 1,000 runs, the convex optimization problem was able to be solved in closed form with the solution to just one linear system. We believe that this reflects a substantial accuracy improvement over prior entry guidance methods that do not adapt to unmodeled atmospheric uncertainties, with only a modest increase in computational cost.

### Acknowledgments

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

### References

- [1] Way, D. W., Powell, R. W., Chen, A., Steltzner, A. D., Martin, A. M. S., Burkhart, P. D., and Mendeck, G. F., “Mars Science Laboratory: Entry, Descent, and Landing System Performance,” *2007 IEEE Aerospace Conference*, 2007, pp. 1–19. <https://doi.org/10.1109/AERO.2007.352821>.
- [2] Justh, H. L., and Lee, K. L., “Mars-GRAM 2010: Additions and Resulting Improvements,” *International Planetary Probe Workshop*, 17-21 Jun. 2013, United States, 2013.
- [3] Li, S., and Jiang, X., “Review and Prospect of Guidance and Control for Mars Atmospheric Entry,” *Progress in Aerospace Sciences*, Vol. 69, 2014, pp. 40–57. <https://doi.org/10.1016/j.paerosci.2014.04.001>.
- [4] Putnam, Z. R., Neave, M. D., and Barton, G. H., “PredGuid Entry Guidance for Orion Return from Low Earth Orbit,” *2010 IEEE Aerospace Conference*, IEEE, Big Sky, MT, USA, 2010, pp. 1–13. <https://doi.org/10.1109/AERO.2010.5447010>.
- [5] Putnam, Z. R., and Braun, R. D., “Precision Landing at Mars Using Discrete-Event Drag Modulation,” *Journal of Spacecraft and Rockets*, Vol. 51, No. 1, 2014, pp. 128–138. <https://doi.org/10.2514/1.A32633>.

- [6] Lu, P., “Entry Guidance: A Unified Method,” *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 3, 2014, pp. 713–728. <https://doi.org/10.2514/1.62605>.
- [7] Lu, P., “Predictor-Corrector Entry Guidance for Low-Lifting Vehicles,” *Journal of Guidance, Control, and Dynamics*, Vol. 31, No. 4, 2008, pp. 1067–1075. <https://doi.org/10.2514/1.32055>.
- [8] Brunner, C. W., and Lu, P., “Skip Entry Trajectory Planning and Guidance,” *Journal of Guidance, Control, and Dynamics*, Vol. 31, No. 5, 2008, pp. 1210–1219. <https://doi.org/10.2514/1.35055>.
- [9] Brunner, C. W., and Lu, P., “Comparison of Fully Numerical Predictor-Corrector and Apollo Skip Entry Guidance Algorithms,” *The Journal of the Astronautical Sciences*, Vol. 59, No. 3, 2012, pp. 517–540. <https://doi.org/10.1007/s40295-014-0005-1>.
- [10] Wang, Z., and Grant, M. J., “Constrained Trajectory Optimization for Planetary Entry via Sequential Convex Programming,” *AIAA Atmospheric Flight Mechanics Conference*, American Institute of Aeronautics and Astronautics, Washington, D.C., 2016. <https://doi.org/10.2514/6.2016-3241>.
- [11] Wang, Z., and Grant, M. J., “Near-Optimal Entry Guidance for Reference Trajectory Tracking via Convex Optimization,” American Institute of Aeronautics and Astronautics, 2018. <https://doi.org/10.2514/6.2018-0013>.
- [12] Mao, Y., Szmuk, M., Xu, X., and Acikmese, B., “Successive Convexification: A Superlinearly Convergent Algorithm for Non-convex Optimal Control Problems,” *arXiv:1804.06539 [math]*, 2019. <https://doi.org/10.48550/arXiv.1804.06539>.
- [13] Szmuk, M., Reynolds, T. P., Acikmese, B., Mesbahi, M., and Carson III, J. M., “Successive Convexification for 6-DoF Powered Descent Guidance with Compound State-Triggered Constraints,” *arXiv:1901.02181 [math]*, 2019.
- [14] Szmuk, M., Reynolds, T. P., and Açıkmeşe, B., “Successive Convexification for Real-Time Six-Degree-of-Freedom Powered Descent Guidance with State-Triggered Constraints,” *Journal of Guidance, Control, and Dynamics*, Vol. 43, No. 8, 2020, pp. 1399–1413. <https://doi.org/10.2514/1.G004549>.
- [15] Malyuta, D., Reynolds, T. P., Szmuk, M., Lew, T., Bonalli, R., Pavone, M., and Acikmese, B., “Convex Optimization for Trajectory Generation,” *arXiv:2106.09125 [cs, eess, math]*, 2021. <https://doi.org/10.48550/arXiv.2106.09125>.
- [16] Betts, J. T., *Practical Methods for Optimal Control Using Nonlinear Programming*, Advances in Design and Control, Vol. 3, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2001.
- [17] Gill, P. E., Murray, W., and Saunders, M. A., “SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization,” *SIAM Review*, Vol. 47, No. 1, 2005, pp. 99–131. <https://doi.org/10.1137/S0036144504446096>.
- [18] Wächter, A., and Biegler, L. T., “On the Implementation of an Interior-Point Filter Line-Search Algorithm for Large-Scale Nonlinear Programming,” *Mathematical Programming*, Vol. 106, No. 1, 2006, pp. 25–57. <https://doi.org/10.1007/s10107-004-0559-y>.
- [19] Howell, T. A., Jackson, B. E., and Manchester, Z., “ALTRO: A Fast Solver for Constrained Trajectory Optimization,” *IEEE International Conference on Intelligent Robots and Systems*, 2019, pp. 7674–7679. <https://doi.org/10.1109/IROS40897.2019.8967788>.
- [20] Jackson, B. E., Punnoose, T., Neamati, D., Tracy, K., Jitosh, R., and Manchester, Z., “ALTRO-C: A Fast Solver for Conic Model-Predictive Control; ALTRO-C: A Fast Solver for Conic Model-Predictive Control,” *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021. <https://doi.org/10.1109/ICRA48506.2021.9561438>.
- [21] Tracy, K., and Manchester, Z., “CPEG: A Convex Predictor-corrector Entry Guidance Algorithm,” *IEEE Aerospace Conference*, Big Sky, MT, USA, 2022, p. 10.
- [22] Boyd, S., and Vandenberghe, L., *Convex Optimization*, Cambridge University Press, 2004.
- [23] Justh, H. L., “Mars Global Reference Atmospheric Model 2010 Version Users Guide,” Tech. Rep. No. NASA/TM-2014-217499, National Aeronautics and Space Administration, Marshall Space Flight Center, 2014.
- [24] Busemann, A., Vinh, N., and Culp, R., “Hypersonic Flight Mechanics,” Tech. Rep. NASA-CR-149170, Sep. 1976.
- [25] Vinh, N. X., Busemann, A., and Culp, R. D., “Hypersonic and Planetary Entry Flight Mechanics,” *NASA STI/Recon Technical Report A*, Vol. 81, 1980, p. 16245.
- [26] Vinh, N., Johnson, W., and Longuski, J., “Mars Aerocapture Using Bank Modulation,” *Astrodynamics Specialist Conference*, American Institute of Aeronautics and Astronautics, Denver, CO, U.S.A., 2000. <https://doi.org/10.2514/6.2000-4424>.

- [27] Wang, Z., and Grant, M. J., “Improved Sequential Convex Programming Algorithms for Entry Trajectory Optimization,” *AIAA Scitech 2019 Forum*, American Institute of Aeronautics and Astronautics, San Diego, California, 2019. <https://doi.org/10.2514/6.2019-0667>.
- [28] Gallais, P., *Atmospheric Re-Entry Vehicle Mechanics*, Springer, Berlin ; New York, 2007.
- [29] Blackmore, L., Açikmeşe, B., and Carson, J. M., “Lossless Convexification of Control Constraints for a Class of Nonlinear Optimal Control Problems,” *2012 American Control Conference (ACC)*, 2012, pp. 5519–5525. <https://doi.org/10.1109/ACC.2012.6314722>.
- [30] Acikmese, B., and Ploen, S. R., “Convex Programming Approach to Powered Descent Guidance for Mars Landing,” *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 5, 2007, pp. 1353–1366. <https://doi.org/10.2514/1.27553>.
- [31] Acikmese, B., Carson, J. M., and Blackmore, L., “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, 2013, pp. 2104–2113. <https://doi.org/10.1109/TCST.2012.2237346>.
- [32] Tracy, K., and Manchester, Z., “CPEG: A Convex Predictor-corrector Entry Guidance Algorithm,” *IEEE Aerospace Conference*, Big Sky, MT, USA, 2022, p. 10.
- [33] Montenbruck, O., Gill, E., and Lutze, F., “Satellite Orbits: Models, Methods, and Applications,” *Applied Mechanics Reviews*, Vol. 55, No. 2, 2002, p. B27. <https://doi.org/10.1115/1.1451162>.
- [34] Hirt, C., Claessens, S., Kuhn, M., and Featherstone, W., “Kilometer-Resolution Gravity Field of Mars: MGM2011,” *Planetary and Space Science*, Vol. 67, No. 1, 2012, pp. 147–154. <https://doi.org/10.1016/j.pss.2012.02.006>.
- [35] Genova, A., Goossens, S., Lemoine, F. G., Mazarico, E., Neumann, G. A., Smith, D. E., and Zuber, M. T., “Seasonal and Static Gravity Field of Mars from MGS, Mars Odyssey and MRO Radio Science,” *Icarus*, Vol. 272, 2016, pp. 228–245. <https://doi.org/10.1016/j.icarus.2016.02.050>.
- [36] Mattingley, J., and Boyd, S., “CVXGEN: A Code Generator for Embedded Convex Optimization,” *Optimization Engineering*, 2012, pp. 1–27.
- [37] Nocedal, J., and Wright, S. J., *Numerical Optimization*, 2<sup>nd</sup> ed., Springer, 2006.
- [38] Vandenberghe, L., “The CVXOPT Linear and Quadratic Cone Program Solvers,” ????, p. 30.
- [39] Markley, F. L., and Crassidis, J. L., *Fundamentals of Spacecraft Attitude Determination and Control*, Springer New York, New York, NY, 2014. <https://doi.org/10.1007/978-1-4939-0802-8>.
- [40] Heidrich, C. R., Holzinger, M. J., and Braun, R. D., “Optimal Information Filtering for Robust Aerocapture Trajectory Generation and Guidance,” *Journal of Spacecraft and Rockets*, 2021. <https://doi.org/10.2514/1.A35175>.
- [41] Dutta, S., Braun, R. D., and Karlgaard, C. D., “Uncertainty Quantification for Mars Entry, Descent, and Landing Reconstruction Using Adaptive Filtering,” *Journal of Spacecraft and Rockets*, Vol. 51, No. 3, 2014, pp. 967–977. <https://doi.org/10.2514/1.A32716>.
- [42] Roelke, E., Hattis, P. D., and Braun, R. D., “IMPROVED ATMOSPHERIC ESTIMATION FOR AEROCAPTURE GUIDANCE,” *Spaceflight Mechanics Meeting*, AAS/AIAA, Savannah, Georgia, 2009, p. 16.
- [43] Kaminski, P., Bryson, A., and Schmidt, S., “Discrete Square Root Filtering: A Survey of Current Techniques,” *IEEE Transactions on Automatic Control*, Vol. 16, No. 6, 1971, pp. 727–736. <https://doi.org/10.1109/TAC.1971.1099816>.
- [44] Tracy, K., “A Square-Root Kalman Filter Using Only QR Decompositions,” , Aug. 2022.
- [45] Strang, G., *Linear Algebra and Its Applications*, 4<sup>th</sup> ed., 1968.
- [46] Simon, D., *Optimal State Estimation: Kalman, H [Infinity] and Nonlinear Approaches*, Wiley-Interscience, Hoboken, N.J, 2006.
- [47] Van der Merwe, R., and Wan, E., “The Square-Root Unscented Kalman Filter for State and Parameter-Estimation,” *IEEE*, 2001, pp. 3461–3464. <https://doi.org/10.1109/ICASSP.2001.940586>.
- [48] Bar-Shalom, Y., Kirubarajan, T., and Li, X.-R., *Estimation with Applications to Tracking and Navigation*, John Wiley & Sons, Inc., USA, 2002.

- [49] Braun, R., Manning, R., Braun, R., and Manning, R., "Mars Exploration Entry, Descent and Landing Challenges," *2006 IEEE Aerospace Conference*, IEEE, Big Sky, MT, USA, 2006, pp. 1–18. <https://doi.org/10.1109/AERO.2006.1655790>.
- [50] Steltzner, A., Kipp, D., Chen, A., Burkhart, D., Guernsey, C., Mendeck, G., Mitcheltree, R., Powell, R., Rivellini, T., San Martin, M., and Way, D., "Mars Science Laboratory entry, descent, and landing system," *2006 IEEE Aerospace Conference*, 2006, pp. 15 pp.–. <https://doi.org/10.1109/AERO.2006.1655796>.