ADAPTIVE MARS ENTRY GUIDANCE WITH ATMOSPHERIC DENSITY ESTIMATION

Felipe Giraldo-Grueso,[*] Andrey A. Popov[†] and Renato Zanetti[‡]

Mars entry requires precise guidance algorithms to steer the entry vehicle to a specific target location. Uncertainty in atmospheric density can degrade the prediction step in numerical predictor-corrector guidance methods, which aim to find the optimal bank angle for the vehicle, thus decreasing targeting accuracy. This work addresses this challenge by introducing a new adaptive guidance algorithm based on a recently proposed navigation solution. The navigation solution, originally designed in an open-loop fashion, estimates atmospheric density using an online-adapting neural network to correct for discrepancies between the onboard nominal and true atmospheric densities. The guidance algorithm uses this neural network, coupled with a simple exponential model, to propagate the current state to a final target location in the prediction step of the Fully Numerical Predictor-Corrector Entry Guidance (FNPEG) framework. By incorporating this adaptive model, the accuracy of the prediction step is enhanced, allowing for more precise guidance of the vehicle toward its target. The proposed method, when compared to using only an exponential model for prediction, shows improved targeting accuracy. Furthermore, the navigation scheme is shown to maintain consistency in the presence of control inputs.

INTRODUCTION

The primary source of uncertainty during Mars entry is the limited knowledge of the true atmospheric density.[1] Discrepancies between the onboard models and the true density can result in inaccurate navigation estimates, increasing the difficulty of targeting a final state through guidance algorithms. Since future robotic and human Mars exploration missions require substantial improvements in landing accuracy due to increased landed mass needs,[2] the development of precise entry navigation and guidance is crucial for their success. Therefore, addressing the uncertainty in atmospheric density can bring substantial improvements to future missions.

Numerical predictor-corrector (NPC) methods are commonly used to address onboard guidance problems. In these types of algorithms, the current state is numerically integrated to a final state (prediction step), and a control input is determined to minimize the error between the final state and a target state (correction step). Different approaches have been applied to entry applications, such as PredGuid[3,4] and the Fully Numerical Predictor-Corrector Entry Guidance (FNPEG).[5,6] FNPEG is a bank control guidance algorithm that decouples the longitudinal and lateral channels to target a desired location by adjusting the bank angle of entry vehicles ranging from low to high lift. Since vertical lift is a function of the bank angle, adjusting it enables the vehicle to modify its range by flying through denser or less dense atmospheric conditions.[7]

[*] Ph.D. Student, Department of Aerospace Engineering and Engineering Mechanics, The University of Texas at Austin, Austin, TX 78712.

[†] Postdoctoral Fellow, Oden Institute for Computational Engineering and Sciences, The University of Texas at Austin, Austin, TX 78712.

[‡] Associate Professor, Department of Aerospace Engineering and Engineering Mechanics, The University of Texas at Austin, Austin, TX 78712.

While NPCs have been shown to be suitable for entry applications,[5,6] this type of framework has limitations. A significant limitation is that the prediction step assumes the parameters in the dynamic models are known. This assumption can make the prediction step inaccurate in uncertain environments, thus resulting in high targeting errors. Therefore, the application of NPCs for Mars entry requires careful consideration of potential modeling errors. A way to address uncertain environments is to use adaptive estimation. For example, if the navigation scheme incorporates adaptive estimation techniques, the prediction step in the NPCs can leverage the adapted models instead of relying solely on nominal profiles. In previous works, Kalman filters[8] and recursive neural networks[9] have been used for the estimation of atmospheric density, demonstrating that incorporating corrected nominal models in the prediction step can increase overall targeting accuracy.

A new method for adaptive filtering for Mars entry was recently developed, using a neural network to estimate atmospheric density and a "consider" analysis to account for its uncertainty.[10] In this approach, a neural network is trained offline on an exponential atmospheric density model, and its weights and biases are adapted online to account for discrepancies between the true atmosphere and the onboard model. The adaptation step is posed as a maximum likelihood problem, aiming to minimize measurement innovations by solving for optimal network parameters at each time step. Although this approach was shown to successfully quantify uncertainty in both state estimates and atmospheric density, its derivation was conducted in an open-loop manner, disregarding control inputs. This work builds upon this new estimation technique by developing guidance commands using the adapted atmospheric estimate and testing the consistency of the filter to control inputs.

The purpose of this work is two-fold. Firstly, we leverage the success of the navigation solution presented in previous work,[10] by incorporating the adaptive model into the prediction step of FNPEG. This integration enables FNPEG to improve targeting accuracy by using the adaptive model rather than relying on a nominal profile. Secondly, since the navigation solution was developed in an open-loop fashion, this work validates the consistency of the estimation technique with control inputs. By closing the loop and incorporating bank angle control into the simulation, we demonstrate that the filter maintains consistency and the guidance algorithm achieves increased targeting accuracy.

MARS ENTRY

The entry phase constitutes the longest segment of the entry, descent, and landing (EDL) sequence, thus playing a crucial role in determining landing precision. In Mars EDL, the entry phase begins approximately 125 km above the Martian surface, when the vehicle first enters the atmosphere, and concludes with parachute deployment at roughly 10 km above the surface.[11] This section summarizes the equations of motion governing hypersonic flight and presents the available sensors used during the entry phase in the recently developed navigation solution.[10]

Dynamics

During the entry phase, the behavior of entry vehicles is commonly described using a three degrees-of-freedom (3-DOF) model.[12–14] Therefore, this work adopts a point mass approximation, disregarding attitude dynamics. The position is described using the planet-centric radius (r), latitude (ϕ), and longitude (θ),[6,11,15]

$$\dot{r} = v\sin(\gamma), \tag{1}$$

$$\dot{\phi} = \frac{\cos(\gamma)\cos(\psi)}{r}v, \tag{2}$$

$$\dot{\theta} = \frac{\cos(\gamma)\sin(\psi)}{r\cos(\phi)}v. \tag{3}$$

The magnitude of the velocity vector (v), flight path angle (γ), and heading azimuth (ψ) are defined relative to the planet surface, based on the vehicle-carried local horizontal frame. The flight path angle is negative-down, and the heading azimuth is defined in the horizontal plane with 0° pointing north and 90° pointing east. In the case of short-duration flights such as the entry phase in EDL, planetary rotation effects may be considered negligible.[11] Therefore, by omitting planetary rotation terms and assuming a first-order gravity model, the velocity equations simplify to:[15]

$$\dot{v} = -\frac{1}{2}\rho\left(\cdot\right)v^2\frac{C_dS}{m} - \frac{\mu}{r^2}\sin(\gamma), \tag{4}$$

$$\dot{\gamma} = \frac{1}{v}\left[\frac{1}{2}\rho\left(\cdot\right)v^2\frac{C_lS}{m}\cos(\sigma) - \frac{\mu}{r^2}\cos(\gamma) + \frac{v^2}{r}\cos(\gamma)\right], \tag{5}$$

$$\dot{\psi} = \frac{1}{v}\left[\frac{1}{2}\rho\left(\cdot\right)v^2\frac{C_lS}{m}\frac{\sin(\sigma)}{\cos(\gamma)} + \frac{v^2}{r}\cos(\gamma)\sin(\psi)\tan(\phi)\right], \tag{6}$$

where σ represents the bank angle. The lift (L) and drag (D) accelerations are defined as:[15]

$$D = \frac{1}{2}\rho\left(\cdot\right)v_\infty^2\frac{C_dS}{m}, \tag{7}$$

$$L = \frac{1}{2}\rho\left(\cdot\right)v_\infty^2\frac{C_lS}{m}. \tag{8}$$

Here, $v_\infty = v - W$, with W denoting wind velocity, C_d is the drag coefficient, C_l is the lift of the vehicle, S represents the area of the vehicle in direct contact with the atmosphere and m is the mass of the vehicle. The term $\rho(\cdot)$ represents atmospheric density, indicating its dependence on different variables such as altitude. Considering the high speeds involved in the entry phase, wind velocities ($v/W \ll 1$) are significantly lower than the velocity of the vehicle making $v_\infty \approx v$ a valid assumption.[16]

Onboard Sensors

For this work, only sensors onboard the spacecraft are used, including an inertial measurement unit (IMU), a cluster of pressure sensors, and a suite of thermocouples. These sensor selections are based on configurations used in previous Mars EDL missions.[11,17–19]

The IMU captures non-gravitational accelerations experienced by the entry vehicle, measured relative to the body frame:

$$\tilde{\boldsymbol{a}}^b = T_v^b\boldsymbol{a}^v + \boldsymbol{\eta}_{\boldsymbol{a}}, \tag{9}$$

where $\boldsymbol{\eta}_{\boldsymbol{a}} \in \mathbb{R}^3$ denotes measurement noise, T_v^b represents the transformation matrix from the velocity frame to the body frame, and $\boldsymbol{a}^v \in \mathbb{R}^3$ comprises non-gravitational accelerations in the velocity frame, defined as:[11]

$$\boldsymbol{a}^v = \begin{bmatrix}-D & L\sin\left(\sigma\right) & L\cos\left(\sigma\right)\end{bmatrix}^{\mathrm{T}}. \tag{10}$$

The dynamic pressure measurement, considering an aggregate measurement from all pressure sensors, is expressed as follows:[11]

$$\tilde{q} = \frac{1}{2}\rho\left(\cdot\right)v^2 + \eta_q, \tag{11}$$

where $\eta_q \in \mathbb{R}$ is measurement noise.

The thermocouples are expected to measure real-time heating rates. The measured convective heating rate follows the Sutton-Graves relation:[20,21]

$$\tilde{\dot{Q}}_s = k\left(\frac{\rho\left(\cdot\right)}{R_n}\right)^{\frac{1}{2}} v^3 + \eta_{\dot{Q}_s}, \tag{12}$$

where $k = 1.9027 \times 10^{-4}\ \mathrm{kg}^{1/2}\mathrm{m}^{-1}$, R_n denotes the vehicle nose radius, and $\eta_{\dot{Q}_s} \in \mathbb{R}$ represents measurement noise. While the precision and availability of the measurements may vary with flight regime, this work assumes available measurements throughout the entire entry phase.

NAVIGATION

This section provides a summary of the previous navigation work,[10] where a filter that uses a neural network within a maximum likelihood framework to adapt to fluctuations in atmospheric density was developed. This summary is included for this work to be self-contained, but readers are encouraged to refer to the original paper for a more detailed explanation. This adaptive filtering technique consists of two stages: offline training and online adaptation.

Offline Training

In the offline training portion, a neural network is trained to predict atmospheric density as a function of the planet-centric radius, such that,

$$\hat{\rho} = \mathcal{NN}(r, \boldsymbol{\xi}), \tag{13}$$

where the parameters of the network are given by $\boldsymbol{\xi}$. To train the network, synthetic data is generated by simulating Mars entry dynamics using a least-squares exponential fit (ρ_{exp}) obtained from different Mars-GRAM profiles. Using this density model, various entry trajectories are simulated, and planet-centric radius and atmospheric density are recorded at a predefined sampling rate. Since the least-squares exponential fit does not accurately represent the true atmospheric density (which, in this work, is represented by the Mars-GRAM profiles), the neural network is adapted online to better approximate the true density.

Online Adaptation

Once the neural network has been trained, an online adaptation scheme is used for the onboard estimation. For this work, the state to be estimated is defined as:

$$\boldsymbol{x} = \begin{bmatrix} r & \phi & \theta & v & \gamma & \psi & B & L/D \end{bmatrix}^T, \tag{14}$$

where the inverse of the ballistic coefficient (B) and the lift-to-drag ratio (L/D) are modeled as a random walk.[11] The available measurement vector is,

$$\boldsymbol{y} = \begin{bmatrix} \tilde{\boldsymbol{a}}^b & \tilde{q} & \tilde{\dot{Q}}_s \end{bmatrix}^T. \tag{15}$$

In this filtering scenario, the chosen estimator is the Unscented Schmidt-Kalman filter (USKF).[22,23] This selection removes the requirement for partial derivatives in both time propagation and measurement update. Furthermore, this filter uses a consider analysis to account for the uncertainty in the atmospheric density estimate, which is not part of the filter's state space.

Time Propagation The iterative process begins with the previous time posterior estimate of the state $\hat{\boldsymbol{x}}_{a,k-1}^{+}$ and covariance $P_{a,k-1}^{+}$, both of which have been augmented to incorporate the consider parameter. The augmented state and covariance are represented as:

$$\hat{\boldsymbol{x}}_{a,k-1}^{+} = \left[\hat{\boldsymbol{x}}_{k-1}^{+\mathrm{T}} \quad \hat{\boldsymbol{c}}_{k-1}^{+}\right]^{\mathrm{T}}, \tag{16}$$

$$P_{a,k-1}^{+} = \begin{bmatrix} P_{k-1}^{+} & C_{k-1}^{+} \\ \left(C_{k-1}^{+}\right)^{\mathrm{T}} & P_{c,k-1}^{+} \end{bmatrix}, \tag{17}$$

where $P_{c,0}^{+}$ is user-defined, with initial values of $\hat{\boldsymbol{c}}_{0}^{+} = 1$ and $C_{0}^{+} = 0_{n_x \times 1}$. Sigma points are then computed based on these augmented estimates. These sigma points are propagated to the next time step by integrating the dynamics and using the neural network to estimate the density. With $F_a(\boldsymbol{x}, \rho)$ representing the flow of the differential equations presented in the *Dynamics* section and the flow of the consider parameter modeled as an exponentially correlated random variable (ECRV)[24] centered at one,

$$\hat{\boldsymbol{\mathcal{X}}}_{k}^{-(i)} = F_a\left(\hat{\boldsymbol{\mathcal{X}}}_{k-1}^{+(i)}, \mathcal{P}_{k-1}^{(i)}\right), \tag{18}$$

$$\mathcal{P}_{k-1}^{(i)} = \hat{\boldsymbol{\mathcal{X}}}_{k-1}^{(i)+}(n_x + 1) \cdot \mathcal{NN}\left(\hat{\boldsymbol{\mathcal{X}}}_{k-1}^{+(i)}(1), \boldsymbol{\xi}_{k-1}\right), \tag{19}$$

where $\hat{\boldsymbol{\mathcal{X}}}_{k-1}^{+(i)}$ represents the i-th previous posterior sigma point, $\hat{\boldsymbol{\mathcal{X}}}_{k-1}^{+(i)}(n_x + 1)$ is the consider parameter at each sigma point, $\hat{\boldsymbol{\mathcal{X}}}_{k-1}^{+(i)}(1)$ is the estimated planet-centric radius at each sigma points and $\boldsymbol{\xi}_{k-1}$ are the current weights and biases of the neural network. After propagating, a prior estimate and covariance are calculated by a weighted sum of the propagated sigma points.[25]

Maximum Likelihood Optimization Once a measurement ($\boldsymbol{y}_k$) is obtained, a loss function is constructed using the measurement log likelihood,

$$\mathcal{L}\left(\xi_{k-1}\right) = \left[\boldsymbol{y}_k - h\left(\hat{\boldsymbol{x}}_k^{-}, \hat{\rho}_{k-1}\right)\right]^{\mathrm{T}} R^{-1} \left[\boldsymbol{y}_k - h\left(\hat{\boldsymbol{x}}_k^{-}, \hat{\rho}_{k-1}\right)\right], \tag{20}$$

$$\hat{\rho}_{k-1} = \mathcal{NN}\left(\hat{\boldsymbol{x}}_k^{-}(1), \boldsymbol{\xi}_{k-1}\right), \tag{21}$$

where h represents the equations presented in the *Onboard Sensors* section. Therefore, to find the optimal network parameters, a minimization problem can be constructed as,

$$\boldsymbol{\xi}_k = \arg\min_{\xi_{k-1}} \mathcal{L}\left(\xi_{k-1}\right). \tag{22}$$

Since this problem deals with the optimization of the parameters of a neural network, backpropagation is used to compute the corresponding gradient. Furthermore, the Adam optimizer[26] is used to minimize the loss function.

Measurement Update Once the adapted parameters of the neural network have been obtained, sigma points are calculated around the prior estimate using the augmented and propagated state and covariance. These prior sigma points are then used to compute a set of expected measurements through the measurement model, incorporating the adapted parameters of the network. These predicted measurements are perturbed slightly by the sigma points of the consider parameter to address uncertainties in the atmospheric density,

$$\hat{\mathcal{Y}}_k^{(i)} = h\left(\hat{\boldsymbol{\mathcal{X}}}_k^{-(i)}, \mathcal{P}_k^{(i)}\right), \tag{23}$$

$$\mathcal{P}_k^{(i)} = \hat{\boldsymbol{\mathcal{X}}}_k^{-(i)}(n_x + 1) \cdot \mathcal{NN}\left(\hat{\boldsymbol{\mathcal{X}}}_k^{-(i)}(1), \boldsymbol{\xi}_k\right). \tag{24}$$

Afterward, the predicted measurement mean, covariance, and cross-covariance with the state are computed to perform a Schmidt-Kalman update, where only the state is updated, while the propagated consider parameter remains unchanged.[22–24] As presented, the filter is independent of bank control inputs, eliminating the need for retraining in scenarios such as closed-loop simulations or when adopting different bank angle profiles.

GUIDANCE

Achieving high targeting accuracy during entry is fundamental for the subsequent descent and landing phases. Bank control has been used as a guidance strategy for maneuvering the vehicle during entry. While various guidance algorithms have been proposed, one of the most adopted methods is the Fully Numerical Predictor-Corrector Entry Guidance (FNPEG), proposed by Lu.[6] This section presents a summary of FNPEG.

Fully Numerical Predictor-Corrector Entry Guidance

FNPEG is an entry guidance algorithm that determines the optimal bank angle of an entry vehicle given a final target state.[6] As previously mentioned, vertical lift is determined by the bank angle. Therefore, adjusting the bank angle allows the vehicle to modify the range covered, either by increasing or decreasing it to fly through denser or less dense atmospheric conditions.[7] FNPEG uses an NPC framework to calculate the magnitude of the bank angle, aiming to minimize the range-to-go along the great circle connecting the final location of the vehicle and the target location, at a set final energy. In FNPEG, the longitudinal and lateral channels are decoupled, meaning that the magnitude of the bank angle is calculated based only on the longitudinal equations. Once the magnitude of the bank angle is determined, the lateral channel determines the correct sign to bound the lateral error within a user-specified threshold.[6]

Longitudinal Channel Under the assumption that the lateral error is small, the magnitude of the bank angle is determined by minimizing the final range-to-go from a specified target latitude and longitude ϕ^*, θ^*. The final range-to-go is obtained by integrating the current state up to the final normalized energy e^* (prediction step), predetermined by a reference trajectory,

$$e^* = \frac{1}{r_n^*} - \frac{v_n^*}{2}, \tag{25}$$

where r_n^* and v_n^* are the normalized target planet-centric radius and velocity, respectively. In this work, to calculate the final range-to-go, the reduced and normalized longitudinal equations are integrated with energy as the independent variable, such that:[5]

$$\frac{dr_n}{de} = \frac{\sin(\gamma)}{D_n}, \tag{26}$$

$$\frac{d\gamma}{de} = \frac{1}{D_n v_n^2}\left[L_n \cos(\sigma) + \left(v_n^2 - \frac{1}{r_n}\right)\left(\frac{\cos(\gamma)}{r_n}\right)\right], \tag{27}$$

$$\frac{ds}{de} = -\frac{\cos(\gamma)}{r_n D_n}, \tag{28}$$

where $v_n = \sqrt{2\left(1/r_n - e\right)}$ and s is the range-to-go. The normalized variables are defined as:

$$r_n = \frac{r}{r_{♂}}, \tag{29}$$

$$v_n = \frac{v}{\sqrt{r_{♂} g_{♂}}}, \tag{30}$$

$$L_n = \frac{L}{g_{♂}}, \tag{31}$$

$$D_n = \frac{D}{g_{♂}}, \tag{32}$$

which follows from normalizing time by $\sqrt{r_{♂}/g_{♂}}$. For these equations, $r_{♂}$ refers to the mean radius of Mars and $g_{♂}$ is the gravitational acceleration at $r_{♂}$. Although using the full 3-DOF dynamics in the longitudinal channel could improve accuracy, in short-range missions, the difference between using the full 3-DOF equations and the longitudinal equations is small.[6] With this setup, the bank angle is found by minimizing the square of the final range-to-go s_f (correction step),

$$\sigma^* = \arg\min_{\sigma} \frac{1}{2} s_f\left(e^*, \sigma\right)^2. \tag{33}$$

For this work, the parametrization of the bank angle profile in the prediction step is kept constant ($\sigma_0 = \sigma_f$, following the notation by presented Lu[5,6]) to reduce the number of tuning parameters in the algorithm. The original FNPEG work minimizes (33) using the Gauss-Newton algorithm, where the first derivative is computed via numerical differentiation, with subsequent iterations using the secant method.[5,6] In this work, a more robust alternative, pattern search,[27] is used to minimize this function, as the focus of this work is on adaptivity rather than optimization techniques.

Lateral Channel The longitudinal channel determines the magnitude of the bank angle since the longitudinal equations are only a function of the cosine of this angle. To determine the correct sign of the bank angle, the lateral logic described by Smith[28] is adopted. Once the magnitude of the bank angle is determined, the current state is integrated over time using the full 3-DOF equations until reaching the final target energy, considering both positive and negative solutions for the bank angle. Once the final state is obtained for both solutions, the final crossrange error is calculated for each. Let χ_f^+ represent the final crossrange error for the solution integrated with the positive bank angle and χ_f^- represent the final crossrange error for the solution integrated with the negative bank angle. Then, if the ratio of the crossrange errors is greater than a threshold K, such that,

$$\left|\frac{\chi_f^+}{\chi_f^-}\right| > K, \tag{34}$$

a bank reversal is commanded. The threshold is designed to be,

$$K = \left|\frac{\chi_f^+}{\chi_{\text{tol}}}\right|^{1/n_i}, \tag{35}$$

where χ_{tol} is the final crossrange tolerance and n_i is the total number of bank reversals left for the trajectory.

Once the magnitude and sign of the bank angle have been determined, the vehicle flies that bank angle until the next guidance call, where the same process is repeated. This continues until the vehicle reaches the final energy. Figure 1 illustrates the longitudinal and lateral channels in FNPEG, where the algorithm aims to minimize the final range-to-go while maintaining the final crossrange error within a specified tolerance.

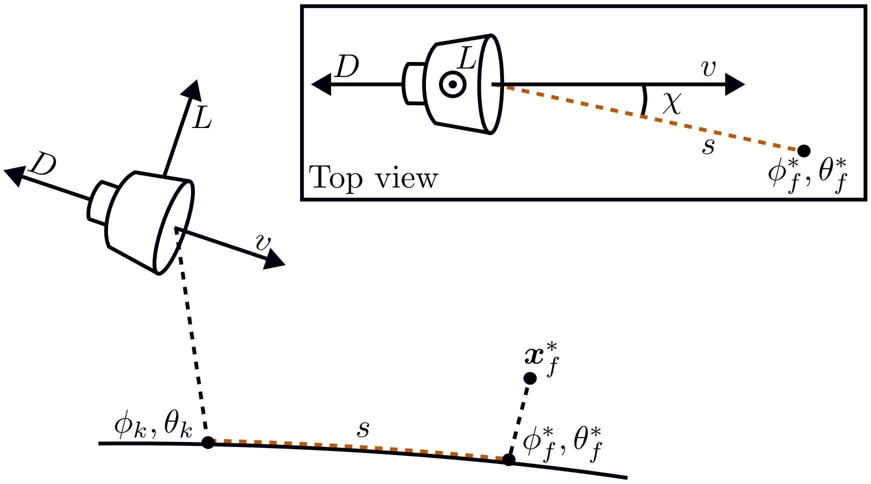

Figure 1. Longitudinal and lateral channels in FNPEG.

Propagating the Trajectory The guidance solution described involves propagating the current state estimate to a final state defined by the target energy. Therefore, it is important to explain how this process is conducted in this work. The neural network used in the navigation filter is consistently being adapted to appropriately fit the atmospheric density within a local region of the filter's current state. In contrast, the exponential fit, used to train the initial neural network, is relatively precise at lower altitudes, as the exponential fit inherently aims to minimize errors in this region. Therefore, the novelty of this work relies on the atmospheric density used for propagating the guidance trajectories. The prediction step (and lateral channel propagation) uses a linear combination of the neural network output and the exponential model, formulated as follows,

$$\rho\left(r,e\right)=\alpha\left(e\right)\mathcal{N}\mathcal{N}\left(r,\boldsymbol{\xi}\right)+\left[1-\alpha\left(e\right)\right]\rho_{\exp}\left(r\right), \tag{36}$$

where α is an exponentially decaying factor with respect to energy defined as,

$$\frac{d\alpha}{de}=-\kappa\frac{\alpha}{v_n}, \tag{37}$$

with κ controlling the exponential decay rate. For every propagation $\alpha(e_k) = 1$, indicating that α always starts equal to one at the current energy. This implies that a high value of κ will result in a very rapid exponential decay of α, indicating that the guidance algorithm will rely more on the exponential atmospheric model than on the neural network. Conversely, lower values of κ will make the guidance algorithm to trust the neural network for a longer duration along the propagation of the trajectory.

RESULTS AND DISCUSSION

To evaluate this new adaptive guidance scheme, a comprehensive Monte Carlo testing approach is used. This involves simulating 1000 different trajectories where the state is estimated via the presented navigation scheme, and guidance commands are calculated using FNPEG and the adaptive density model. Each true trajectory is simulated using an independent and distinct Mars-GRAM density profile. Since Mars-GRAM trajectories provide density at discrete altitudes, trajectory points are interpolated using a cubic spline to determine density values at the queried altitude.

It is important to note that, the least squares exponential model used to train the neural network offline and used to propagate the guidance trajectories, is a fit to different Mars-GRAM profiles that are not used in the Monte Carlo testing. In both cases, however, the Mars-GRAM profiles are generated by setting the density and wind random perturbations to 1, which represent a standard deviation ranging from 2% to 45% of the unperturbed mean.

Monte Carlo Settings

This work uses the same Monte Carlo settings as detailed in the previous work,[10] with the only difference being that the vehicle is given slightly more lift to improve controllability. The initial conditions, as outlined in Table 1, correspond to the entry conditions for the Mars Science Laboratory (MSL).

Table 1. Entry conditions for the true trajectories

Parameter	**Value**	**Standard Deviation** (3σ)
r_0 (m)[29]	3.5222×10^6	3.2066×10^1
ϕ_0 (deg)[29]	-0.3919×10^1	7.8100×10^{-4}
θ_0 (deg)[29]	1.2672×10^2	3.6700×10^{-4}
v_0 (m/s)[29]	6.0833×10^3	2.6059×10^{-2}
γ_0 (deg)[29]	-1.5489×10^1	4.0000×10^{-4}
ψ_0 (deg)[29]	9.3206×10^1	2.6800×10^{-4}
B_0 (m^2/kg)[11,30]	7.1000×10^{-3}	4.8000×10^{-3}
L_0/D_0 (n.d.)	2.7000×10^{-1}	1.5178×10^{-1}

The statistical properties of process and measurement noise are presented in Tables 2 and 3 respectively. The consider parameter is modeled as an ECRV centered at one with $\tau = 5$, $P_{ss} = 1 \times 10^{-3}$, and an initial covariance of $P_{c,0} = 1 \times 10^{-10}$. Additionally, for each simulated trajectory, the angle of attack remains constant at $\alpha = -17°$.

Table 2. Process noise statistics

Parameter	**Standard Deviation** (3σ)
r (m)	-
ϕ (deg)	-
θ (deg)	-
v (m/s)	3×10^{-1}
γ (deg)	4×10^{-6}
ψ (deg)	4×10^{-7}
B (m^2/kg)	1×10^{-5}
L/D (n.d.)	3×10^{-5}

Table 3. Measurement noise statistics

Parameter	Standard Deviation (3σ)
$\tilde{\boldsymbol{a}}^b$ (μg)[14]	1×10^2
$\tilde{q}$ (% of reading)[17]	1
$\tilde{\dot{Q}}_s$ (% of reading)	1

The target final state is obtained by propagating a nominal trajectory using the full 3-DOF equations with a randomly selected Mars-GRAM profile. The nominal trajectory is integrated using a constant bank angle of $\sigma = 50°$ (reversing the sign when 2500 m/s $\leq v(t) \leq 5500$ m/s), until a final velocity of $v_f = 1100$ m/s is reached. This velocity marks the transition from range control to heading alignment.[31] Additionally, this targeting state is selected so that it is reachable by the guidance algorithm if no navigation errors are considered, and the true atmospheric density is known.

Table 4. Target final state

Parameter	Value
r_f^* (m)	3.4167×10^6
ϕ_f^* (deg)	-0.4385×10^1
θ_f^* (deg)	1.3726×10^2
v_f^* (m/s)	1.1000×10^3

Guidance Algorithms

To assess the effectiveness of the adaptive guidance algorithm, three different algorithms are tested and compared.

- The first algorithm, referred to as TG (True Guidance), has access to the true vehicle state and the true atmospheric density. This means that no navigation errors are considered. This algorithm serves as a baseline for the best achievable performance.
- The second algorithm, referred to as GEXP (Guidance with Exponential Model), uses the state estimates from the navigation filter but employs only the exponential density model in the prediction step of FNPEG and the crossrange calculations (i.e. $\alpha = 0 \,\forall\, e$ in (36)).
- The third algorithm, referred to as GADA (Guidance with Adaptive Model), uses the state estimates from the filter and the adaptive density model for the prediction and crossrange calculations as in (36).

In each guidance algorithm, for a more realistic scenario, the maximum rate of change of the bank angle is set to 15°/s. In addition, FNPEG is called at a frequency of 1 Hz once the sensed acceleration is greater than 0.2 times Earth's gravity, or roughly 0.53 times Mars gravity ($||\tilde{\boldsymbol{a}}^b|| >$ 1.96 m/s^2).[31,32] Before range control starts, the vehicle flies at a constant bank angle of $50°$. For crossrange calculations, the final crossrange tolerance is set to $\chi_{\text{tol}} = 0.0075°$, and the maximum desired number of bank reversals is $n = 3$.

Monte Carlo Results

To compare the three solutions, the final crossrange error (χ_f), downrange error (dr_f) and absolute range-to-go ($|s_f|$) are used. The final crossrange error is calculated as:

$$\begin{aligned} \sin(\chi_f) = &-\sin(\psi_f)\left[\sin(\phi^*)\cos(\phi_f)\cos(\Delta\theta) - \cos\phi^*\sin(\phi_f)\right] \\ &- \cos(\psi_f)\cos(\phi_f)\sin(\Delta\theta)\,, \end{aligned} \tag{38}$$

where $\Delta\theta = \theta^* - \theta_f$. To calculate the final downrange error, the following equations are used:

$$\begin{aligned}
\sin(\mu) &= \cos(\psi_f)\left[\sin(\phi^*)\cos(\phi_f)\cos(\Delta\theta) + \cos(\phi^*)\sin(\phi_f)\right] \\
&\quad + \sin(\psi_f)\cos(\phi_f)\sin(\Delta\theta), \\
\cos(\mu) &= \cos(\phi^*)\cos(\phi_f)\cos(\Delta\theta) + \sin(\phi^*)\sin(\phi_f), \\
dr_f &= \arctan 2\left(\sin(\mu), \cos(\mu)\right).
\end{aligned} \tag{39}$$

And the final absolute range-to-go is calculated as:

$$|s_f| = \arccos(\sin(\phi^*)\sin(\phi_f) + \cos(\phi^*)\cos(\phi_f)\cos(\Delta\theta)). \tag{40}$$

It is important to note that the final absolute range-to-go, as calculated, will only yield positive values. Therefore, to gauge trajectory overshooting, the downrange error serves as a more informative metric. Furthermore, since all these angles are expressed in radians, to quantify the distance dispersion, these quantities can be multiplied by the mean radius of Mars.

Figure 2 illustrates the final targeting plots (at the final target energy) for the three previously mentioned guidance algorithms. The crosses show the mean values, and the ellipses show the 3-σ error bounds. For this specific figure, $\kappa = 0.5$ for GADA. As expected, the true guidance solution achieves the best targeting results, concluding the trajectory with minimal crossrange and downrange errors. The results obtained with the adaptive guidance exhibit higher errors but remain lower than those obtained with the guidance algorithm using only the exponential density model. This is further evident as the targeting ellipse for GADA lies entirely within a 2.5 km radius of the target position, whereas the targeting ellipse for GEXP expands up to a 5 km radius. Table 5 shows the final mean downrange and crossrange errors along with their standard deviation.

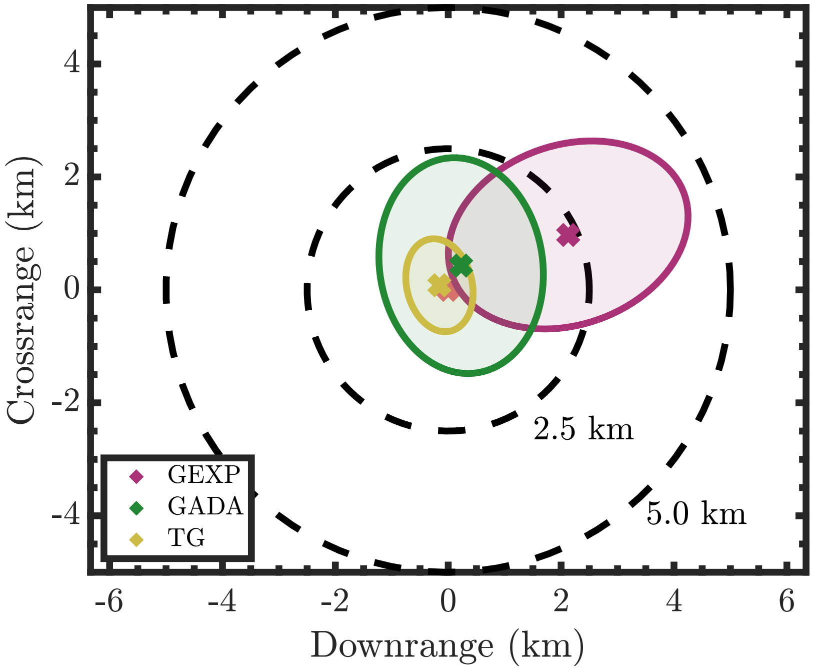

Figure 2. Crossrange error vs. downrange error for the three guidance algorithms. The crosses represent the mean, and the ellipses show the 3-σ error bounds.

Table 5. Final downrange and crossrange errors

Algorithm	$\bar{dr}_f$ **(km)**	$\sigma\left(dr_f\right)$ **(km)**	$\bar{\chi}_f$ **(km)**	$\sigma\left(\chi_f\right)$ **(km)**
GEXP	0.2126×10^{1}	7.0784×10^{-1}	9.7219×10^{-1}	5.5486×10^{-1}
GADA	2.2858×10^{-1}	4.8578×10^{-1}	4.2985×10^{-1}	6.3682×10^{-1}
TG	-1.5576×10^{-1}	1.9885×10^{-1}	8.1747×10^{-2}	2.7434×10^{-1}

Figure 3 shows the final absolute range-to-go histograms for the three guidance solutions. As in Figure 2, for this figure, $\kappa = 0.5$ for GADA. The true guidance exhibits the best performance, with a final absolute range-to-go approaching zero. Similar to the crossrange and downrange plots, the final absolute range-to-go for the adaptive guidance is closer to zero than that obtained with the guidance algorithm using the exponential model. These results indicate that the adaptive density model provided to the guidance algorithm by the navigation filter better quantifies the true density. Consequently, the prediction step and crossrange calculations are more accurate, enabling the control inputs to guide the vehicle more precisely toward the target.

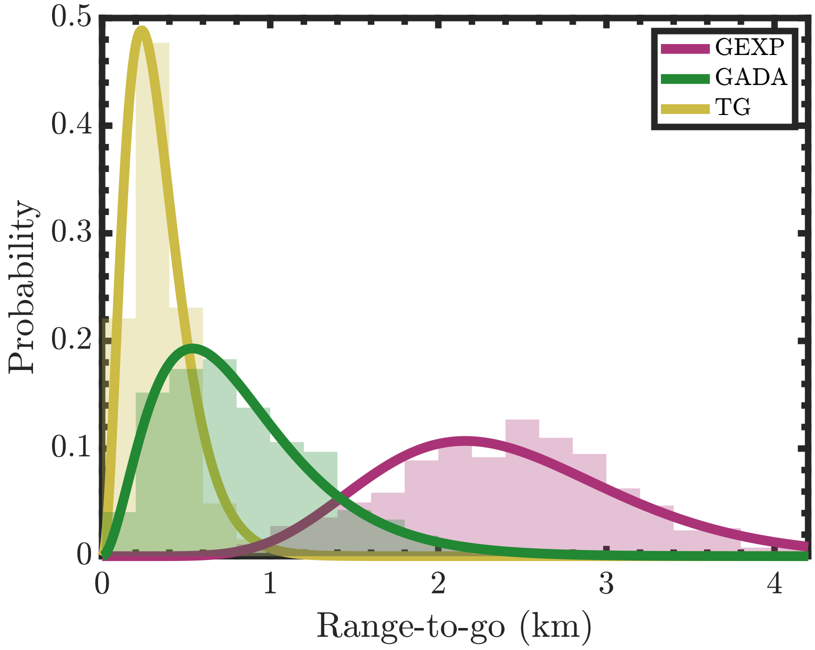

Figure 3. Final absolute range-to-go for the three guidance algorithms.

As the results obtained with GADA are a function of the exponential decay parameter in (36), it becomes important to study the impact that this parameter can have on the final errors. Figure 4 shows the final targeting plots for different values of κ, and Figure 5 shows the final absolute range-to-go for different values of κ. As previously mentioned, a high value of κ indicates a fast exponential decay of α, meaning that the guidance algorithm will trust the exponential atmospheric model more than the neural network. Lower values of κ make the guidance algorithm trust the neural network for longer in the trajectory. As it can be seen from the figure, $\kappa = 50$ yields very similar results to those obtained using the exponential model only. In comparison, by increasing κ, the error ellipses tend to move towards the results obtained with TG. These results show that the information given by the neural network to the guidance algorithm is valuable and enhances the targeting performance.

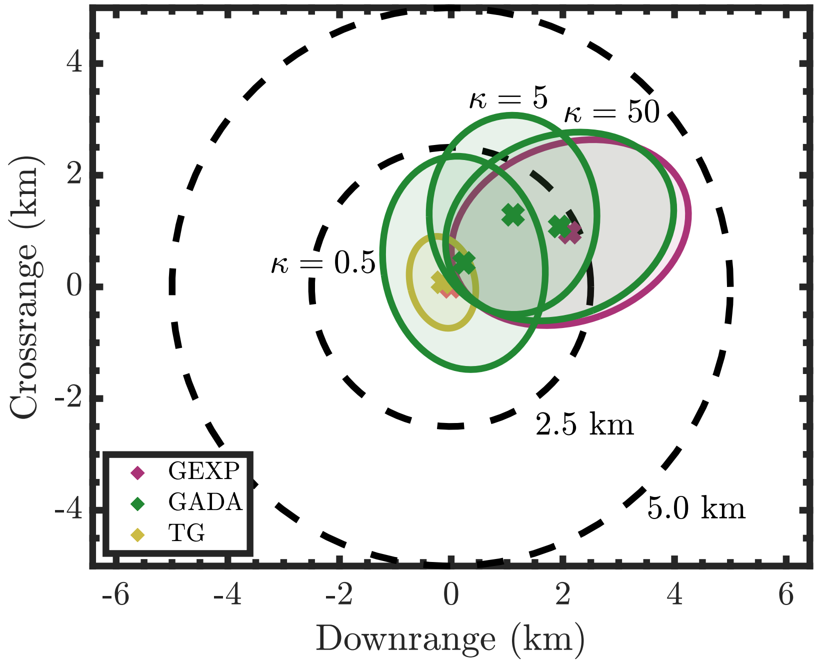

Figure 4. Crossrange error vs. downrange error for different values of κ.

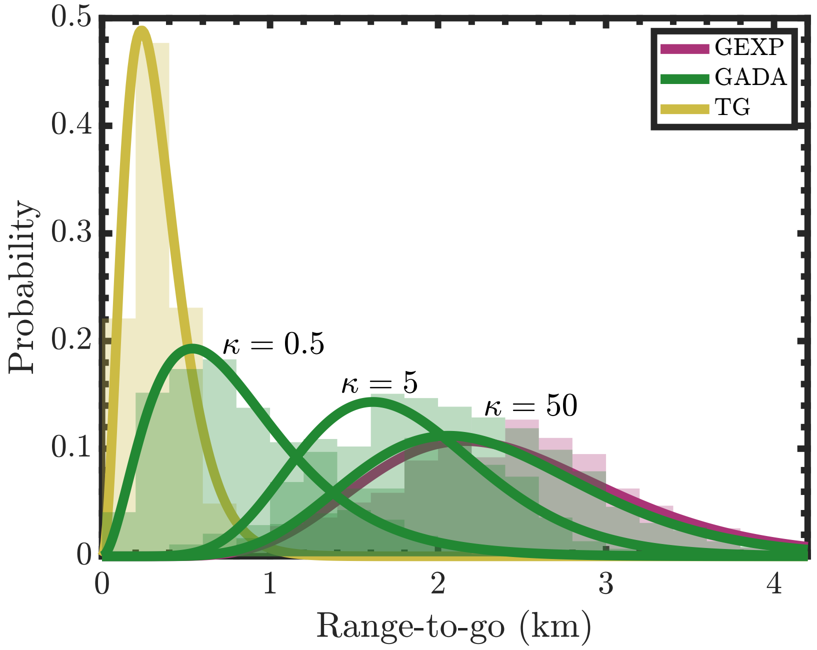

Figure 5. Final absolute range-to-go for different values of κ.

As mentioned earlier, the navigation architecture was previously developed without considering any control inputs to the system.[10] This current work serves as a sanity check to verify the filter's consistency when its state estimates are used for guidance. Figure 6 illustrates the estimation error as a function of time. While only 50 Monte Carlo runs are shown, the sample mean, filter covariance and sample covariance are calculated using every run. As it can be seen, the estimates are very close to zero mean, and the filter's predicted covariance accurately matches the sample covariance of the Monte Carlo runs. These error plots closely resemble those presented in the previous work,[10] although there is a higher estimation error with the heading azimuth, which stems from the fact that the bank angle is constantly changing. Regardless, this demonstrates that the estimation architecture presented in previous work remains consistent in a closed-loop simulation.

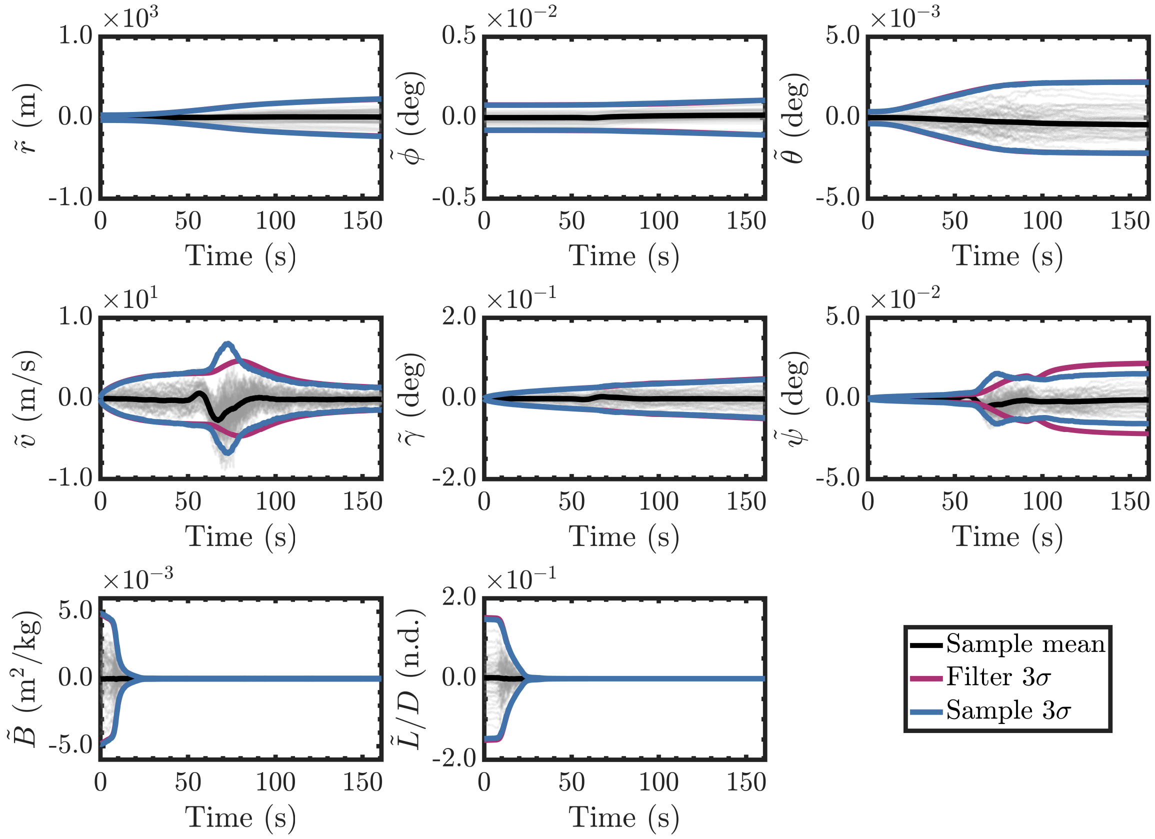

Figure 6. Estimation error as a function of time. The gray lines show 50 Monte Carlo trajectories out of the 1000 runs.

CONCLUSIONS

In this work, a new adaptive guidance algorithm is presented. This algorithm combines the adaptive neural network density model from a recently developed navigation solution[10] with a simple exponential model to provide a better atmospheric density approximation for the prediction step in FNPEG. The neural network is continually adapted to accurately represent the atmospheric density within the local region of the filter's current state, while the exponential model inherently has low errors at low altitudes. Therefore, the prediction step and lateral channel propagation in FNPEG use a weighted linear combination of the neural network output and the exponential model. In proximity to the current state, the density from the neural network is given more weight, whereas at lower altitudes, the density from the exponential model is given a higher weight.

The proposed algorithm was tested through a Monte Carlo analysis using different atmospheric profiles sampled from Mars-GRAM. The new adaptive guidance algorithm was compared with FNPEG using only the exponential model. The results showed that using the linear combination of the neural network and the exponential model increased targeting accuracy. Specifically, every trajectory using the linear combination ended within a 2.5 km radius of the target final location, while trajectories using only the exponential model had errors of up to a 5.0 km radius. The new algorithm closely approached the baseline performance given by running FNPEG with no navigation errors and using the true atmospheric density. Additionally, the new navigation solution was tested in a closed-loop scenario, demonstrating that it can be used with control inputs without significantly affecting its performance.

ACKNOWLEDGMENT

The authors would like to thank Jeremy R. Rea for the insightful discussion on entry guidance. This material is based on research sponsored by Air Force Office of Scientific Research (AFOSR) under award number: FA9550-22-1-0419.

REFERENCES

[1] R. Zanetti, *Advanced Navigation Algorithms for Precision Landing*. PhD thesis, The University of Texas at Austin, 2007.

[2] R. D. Braun and R. M. Manning, “Mars exploration entry, descent and landing challenges,” *2006 IEEE Aerospace Conference*, 2006, 10.1109/AERO.2006.1655790.

[3] J. L. DiCarlo, “Aerocapture guidance methods for high energy trajectories,” Master’s thesis, Massachusetts Institute of Technology, 2003.

[4] Z. R. Putnam, M. D. Neave, and G. H. Barton, “PredGuid entry guidance for Orion return from low Earth orbit,” *2010 IEEE Aerospace Conference*, 2010, pp. 1–13, 10.1109/AERO.2010.5447010.

[5] P. Lu, “Predictor-Corrector Entry Guidance for Low-Lifting Vehicles,” *Journal of Guidance, Control, and Dynamics*, Vol. 31, No. 4, 2008, pp. 1067–1075, 10.2514/1.32055.

[6] P. Lu, “Entry Guidance: A Unified Method,” *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 3, 2014, pp. 713–728, 10.2514/1.62605.

[7] J. Ridderhof, P. Tsiotras, and B. J. Johnson, “Stochastic Entry Guidance,” *Journal of Guidance, Control, and Dynamics*, Vol. 45, No. 2, 2022, pp. 320–334, 10.2514/1.G005964.

[8] K. Tracy, G. Falcone, and Z. Manchester, “Robust Entry Guidance with Atmospheric Adaptation,” *AIAA SciTech Forum*, 2023, 10.2514/6.2023-0301.

[9] J. A. Rataczak, D. Amato, and J. W. McMahon, “Density Estimation for Entry Guidance Problems using Deep Learning,” *AIAA SciTech Forum*, 2024, 10.2514/6.2024-0946.

[10] F. Giraldo-Grueso, A. A. Popov, and R. Zanetti, “Precision Mars Entry Navigation with Atmospheric Density Adaptation via Neural Networks,” *Journal of Aerospace Information Systems*, Vol. 0, No. 0, 2024, 10.2514/1.I011426.

[11] N. S. Zewge and H. Bang, “A Distributionally Robust Fusion Framework for Autonomous Multisensor Spacecraft Navigation during Entry Phase of Mars Entry, Descent, and Landing,” *Remote Sensing*, Vol. 15, No. 1139, 2023, 10.3390/rs15041139.

[12] N. Vinh and R. Busemann, A.and Culp, *Hypersonic and Planetary Entry Flight Mechanics*. The University of Michigan Press, 1980.

[13] J. Lévesque, *Advanced Navigation and Guidance for High-Precision Planetary Landing on Mars*. PhD thesis, University of Sherbrooke, 2006.

[14] J. Christian, A. Verges, and R. Braun, “Statistical Reconstruction of Mars Entry, Descent, and Landing Trajectories and Atmospheric Profiles,” *AIAA SPACE 2007 Conference & Exposition*, 2007, 10.2514/6.2007-6192.

[15] J. F. Lévesque and J. d. Lafontaine, “Innovative Navigation Schemes for State and Parameter Estimation During Mars Entry,” *Journal of Guidance, Control and Dynamics*, Vol. 30, No. 1, 2007, pp. 169–184, 10.2514/1.25107.

[16] D. Amato and J. W. McMahon, “Deep Learning Method for Martian Atmosphere Reconstruction,” *Journal of Aerospace Information Systems*, Vol. 18, 2021, pp. 728–738, 10.2514/1.I010922.

[17] M. J. Gazarik, M. J. Wright, A. Little, F. M. Cheatwood, J. A. Herath, M. M. Munk, F. J. Novak, and E. R. Martinez, “Overview of the MEDLI Project,” *2008 IEEE Aerospace Conference*, 2008, 10.1109/AERO.2008.4526285.

[18] H. Hwang, D. Bose, T. White, H. Wright, M. Schoenenberger, C. Kuhl, D. Trombetta, J. Santos, T. Oishi, C. Karlgaard, and e. al, “Mars 2020 Entry, Descent and Landing Instrumentation 2 (MEDLI2),” *46th AIAA Thermophysics Conference*, 2016.

[19] T. R. White, M. Mahzari, R. A. Miller, C. Y. Tang, C. D. Karlgaard, H. Alpert, H. S. Wright, and C. Kuhl, “Mars Entry Instrumentation Flight Data and Mars 2020 Entry Environments,” *AIAA SCITECH 2022 Forum*, San Diego, CA, 2022, 10.2514/6.2022-0011.

[20] K. Sutton and J. Randolph A. Graves, “A General Stagnation-Point Convective-Heating Equation for Arbitrary Gas Mixtures,” tech. rep., NASA Langley Research Center, 1971.

[21] J. Benito and K. D. Mease, “Reachable and Controllable Sets for Planetary Entry and Landing,” *Journal of Guidance, Control, and Dynamics*, Vol. 33, No. 3, 2010, pp. 641–654, 10.2514/1.47577.

[22] R. Zanetti and K. J. DeMars, “Joseph Formulation of Unscented and Quadrature Filters with Application to Consider States,” *Journal of Guidance, Control, and Dynamics*, Vol. 36, No. 6, 2013, pp. 1860–1864, 10.2514/1.59935.

[23] J. Stauch and M. Jah, “Unscented Schmidt–Kalman Filter Algorithm,” *Journal of Guidance, Control, and Dynamics*, Vol. 38, No. 1, 2015, pp. 117–123, 10.2514/1.G000467.

[24] R. Zanetti and C. D’Souza, “Recursive Implementations of the Consider Filter,” *Journal of the Astronautical Sciences*, Vol. 60, No. 3, 2013, p. 672–685, 10.1007/s40295-015-0068-7.

[25] E. Wan and R. Van Der Merwe, “The unscented Kalman filter for nonlinear estimation,” *IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium*, 2000, pp. 153–158, 10.1109/ASSPCC.2000.882463.

[26] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” *3rd International Conference for Learning Representations*, 2017, 10.48550/arXiv.1412.6980.

[27] C. Audet and J. E. Dennis, “Analysis of Generalized Pattern Searches,” *SIAM Journal on Optimization*, Vol. 13, No. 3, 2002, pp. 889–903, 10.1137/S1052623400378742.

[28] K. M. Smith, “Predictive Lateral Logic for Numerical Entry Guidance Algorithms,” *AAS/AIAA Space Flight Mechanics Meeting*, 2016.

[29] S. Dutta and R. D. Braun, “Statistical Entry, Descent, and Landing Performance Reconstruction of the Mars Science Laboratory,” *Journal of Spacecraft and Rockets*, Vol. 51, No. 4, 2014, pp. 1048–1061, 10.2514/1.A32937.

[30] D. W. Way, R. W. Powell, A. Chen, A. D. Steltzner, A. M. S. Martin, P. D. Burkhart, and G. F. Mendeck, “Mars Science Laboratory: Entry, Descent, and Landing System Performance,” *2007 IEEE Aerospace Conference*, 2007, 10.1109/AERO.2007.352821.

[31] G. Mendeck and L. Craig, “Entry Guidance for the 2011 Mars Science Laboratory Mission,” *AIAA Atmospheric Flight Mechanics Conference*, 2010, 10.2514/6.2011-6639.

[32] A. Nelessen, C. Sackier, I. Clark, P. Brugarolas, G. Villar, A. Chen, A. Stehura, R. Otero, E. Stilley, D. Way, K. Edquist, S. Mohan, C. Giovingo, and M. Lefland, “Mars 2020 Entry, Descent, and Landing System Overview,” *2019 IEEE Aerospace Conference*, 2019, pp. 1–20, 10.1109/AERO.2019.8742167.