Adaptive Atmospheric Entry Guidance via Neural Networks

Felipe Giraldo-Grueso*
The University of Texas at Austin, Austin, TX 78712

Andrey A. Popov†
The University of Hawaiʻi at Mānoa, Honolulu, HI 96822

Jeremy R. Rea‡
NASA Johnson Space Center, Houston, TX 77058

Renato Zanetti§
The University of Texas at Austin, Austin, TX 78712

I. Introduction

One of the main sources of uncertainty during atmospheric entry is the limited knowledge of the true density [1]. Discrepancies between onboard atmospheric density models and the true density can lead to inaccurate state propagation, increasing the difficulty of targeting a final state through navigation and guidance algorithms. Since future robotic and human exploration missions require improvements in landing accuracy due to the need for increased landed mass [2], the development of precise entry navigation and guidance is crucial for their success. Addressing the uncertainty in atmospheric density can, therefore, bring improvements to future missions.

Numerical predictor-corrector (NPC) algorithms are used to solve onboard guidance problems. In these algorithms, the current state is numerically integrated to a final state (prediction step), and a control input is determined by minimizing the error between the final state and a target state (correction step). Different NPC approaches have been applied to atmospheric entry applications, such as PredGuid [3–5] and the Fully Numerical Predictor-Corrector Entry Guidance (FNPEG) algorithm [6–8]. FNPEG is a guidance algorithm that targets a desired location by adjusting the bank angle of entry vehicles. Since the vertical lift-to-drag ratio is a function of the bank angle, adjusting it allows the vehicle to modify its range by flying through denser or less dense atmospheric conditions [9].

These NPC algorithms applied to atmospheric entry applications can achieve high targeting accuracy since they use the full nonlinear dynamics to predict the trajectory of the vehicle [10]. Therefore, they are not limited by linear approximations [11]. It is important to note that other control strategies have been proposed, such as augmented convex-concave decompositions [12], and drag-energy formulations in convex form [13]. However, this work focuses only on NPC algorithms [14], such as FNPEG.*

*Ph.D. Candidate, Department of Aerospace Engineering and Engineering Mechanics; fgiraldo@utexas.edu (corresponding author)
†Assistant Professor, Department of Information and Computer Sciences; apopov@hawaii.edu
‡Orion Entry GN&C MODE Team Guidance Sub-System Manager and Guidance Technical Discipline Lead, Flight Mechanics & Trajectory Design Branch; jeremy.r.rea@nasa.gov
§Associate Professor, Department of Aerospace Engineering and Engineering Mechanics, AIAA Associate Fellow; renato@utexas.edu
*This paper extends the work presented at the Astrodynamics Specialist Conference, Broomfield, CO, Aug. 11-15, 2024. No. AAS 24-391 [14].

Although FNPEG has been shown to be suitable for entry applications [6–8, 11], this type of framework has certain limitations. An important limitation is that the prediction step assumes that the parameters in the dynamic models, such as atmospheric density, are known. This assumption can lead to inaccuracies when flying through uncertain environments, resulting in higher targeting errors. Therefore, the use of NPCs for atmospheric entry requires consideration of potential modeling errors in atmospheric density. This issue is typically addressed by incorporating first-order fading-memory filters to estimate atmospheric quantities such as lift or drag accelerations [11]. Recently, a constant-bias estimator, determined via a first-order fading-memory filter, has been successfully used to handle density adaptation in the skip entry of Orion [3, 5]. As such, this approach is considered state-of-the-art for this work.

Constant-bias estimation has limitations as it applies a multiplier to the nominal density profile [15], restricting how the model can adapt. A different way to address uncertain environments is to use adaptive estimation [16, 17]. If the navigation scheme incorporates adaptive estimation, the prediction step in NPCs can take advantage of the adapted models instead of relying on nominal profiles. In previous work, Kalman filters [18] and recursive neural networks [19, 20] have been used to estimate atmospheric density, demonstrating that incorporating corrected models into the prediction step of NPCs can increase targeting accuracy in atmospheric entry applications. Density interpolators and ensemble correlation filters have also been proposed for the case of aerocapture [15]. Furthermore, high-order sliding-mode theory has been used to develop an adaptive-control scheme robust to atmospheric disturbances [21].

An adaptive filtering technique for atmospheric entry was recently developed, by using a neural network to estimate density onboard [22]. In this approach, a neural network is trained offline on a low-fidelity atmospheric density model, and its parameters are adapted online to account for discrepancies between the low-fidelity model and the true atmosphere. The adaptation step is set up as a maximum likelihood (ML) problem, minimizing the difference between predicted and true measurements with respect to the network parameters. This work builds on this estimation technique by developing guidance commands using the adapted atmospheric model and testing the consistency of the filter when control inputs are included.

The purpose of this work, therefore, is twofold [14]. First, we use the navigation solution presented in [22] to incorporate the adaptive density model into the prediction step of FNPEG. Second, since the navigation solution was developed in an open-loop framework with a constant bank angle, this work validates the consistency of the estimation technique when applied to scenarios with bank control. By closing the loop, i.e., using the navigation output to update the bank angle during flight, it is shown that the filter remains consistent as the bank angle varies.

To show this, we use two different atmospheric entry scenarios. The first, a Mars entry mission, highlights the advantages of using the adapted model over nominal models determined offline. The second scenario, an Earth re-entry mission, shows how the proposed adapted model can be combined with state-of-the-art adaptation techniques used in real entry scenarios. In both cases, the targeting and estimation errors are analyzed to evaluate the performance of the guidance algorithm and the consistency of the filter.

The remainder of this paper is organized as follows. Section II provides an introduction to atmospheric entry including the different atmospheric density models used in this work. A summary of the navigation solution is presented in Section III. The new guidance solution, incorporating the adaptive model into the prediction step of FNPEG, is presented in Section IV. The proposed technique is evaluated in a Mars entry scenario in Section V.A, followed by an Earth re-entry problem in Section V.B. Finally, Section VI provides conclusions and future work.

II. Atmospheric 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 [23]. This section summarizes the equations of motion that govern hypersonic flight and presents some of the sensors available for use during the entry phase, as used in [22].

A. Dynamics

During the entry phase, the vehicle can be modeled using a three degrees of freedom (3-DOF) approach [24–26]. This model describes the position of the vehicle using the planet-centric radius (r), latitude (ϕ), and longitude (θ) with respect to the planet-centered, planet-fixed frame [7, 23, 27],

$$\dot{r} = v \sin(\gamma), \quad \dot{\phi} = \frac{\cos(\gamma)\cos(\psi)}{r} v, \quad \dot{\theta} = \frac{\cos(\gamma)\sin(\psi)}{r\cos(\phi)} v. \tag{1}$$

The velocity components of the vehicle are described by the magnitude of its velocity vector (v), flight path angle (γ), and heading azimuth (ψ), relative to the geographic frame [27]. The flight path angle is negative-down, and the heading azimuth is defined by 0° pointing north and 90° pointing east. Planetary rotation and higher-order gravity terms may be included in entry dynamics, but they are not considered here.[†] Since the analysis focuses on the effects of atmospheric density mismatch, these terms are omitted and a first-order gravity model is adopted. Under these assumptions, the velocity equations reduce to [27]

$$\dot{v} = -D - \frac{\mu}{r^2}\sin(\gamma), \quad \dot{\gamma} = \frac{1}{v}\left[L\cos(\sigma) - \frac{\mu}{r^2}\cos(\gamma) + \frac{v^2}{r}\cos(\gamma)\right], \quad \dot{\psi} = \frac{1}{v}\left[L\frac{\sin(\sigma)}{\cos(\gamma)} + \frac{v^2}{r}\cos(\gamma)\sin(\psi)\tan(\phi)\right], \tag{2}$$

where σ represents the positive-right bank angle, and the drag (D) and lift (L) accelerations are defined as [27]

$$D = \frac{1}{2}\rho v_\infty^2 \frac{C_D S}{m}, \quad L = \frac{1}{2}\rho v_\infty^2 \frac{C_L S}{m}. \tag{3}$$

Here, $v_\infty = v - W$, where W is the wind velocity, C_D is the drag coefficient, C_L is the lift coefficient, S represents the area of the vehicle in direct contact with the atmosphere, and m is the mass of the vehicle. The term ρ represents

[†]Note that in the case of short-duration flights, planetary rotation effects may be considered negligible [23].

atmospheric density. Considering the high speeds involved in the entry phase, wind velocities are significantly lower than the velocity of the vehicle ($W/v \ll 1$), making $v_\infty \approx v$ a valid assumption for most of the flight [19]. Figure 1 shows the position components with respect to the planet-centered, planet-fixed frame, and the velocity components with respect to the geographic frame.

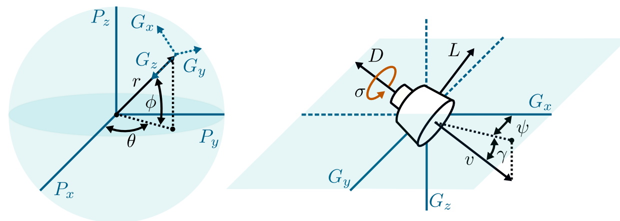

Fig. 1 Position components with respect to the planet-centered planet-fixed frame (left), and velocity components with respect to the geographic frame (right). Figure adapted from [22].

B. Onboard Sensors

For this work, sensors onboard the spacecraft are used, including an inertial measurement unit (IMU), a cluster of pressure sensors, and a suite of thermocouples. In addition to these, global positioning system (GPS) measurements are used for the Earth re-entry scenario. These sensor selections are based on the configurations used in previous EDL missions [4, 23, 28–30].

The IMU measures non-gravitational accelerations relative to the body frame,

$$\tilde{\boldsymbol{a}}^b = T_v^b \boldsymbol{a}^v + \boldsymbol{\eta_a}, \quad \boldsymbol{a}^v = \begin{bmatrix} -D & L\sin(\sigma) & L\cos(\sigma) \end{bmatrix}^T, \tag{4}$$

where $T_v^b \in \mathbb{R}^{3\times3}$ is the velocity frame to body frame transformation matrix, $\boldsymbol{a}^v \in \mathbb{R}^3$ are velocity frame accelerations [23], and $\boldsymbol{\eta_a} \in \mathbb{R}^3$ is zero-mean, white, and Gaussian measurement noise.

The pressure sensors are assumed to directly measure the dynamic pressure, defined as [23]

$$\tilde{q} = \frac{1}{2}\rho v^2 + \eta_q, \tag{5}$$

where η_q is zero-mean, white, and Gaussian measurement noise. Note that this is a simplification, assuming that the dynamic pressure can be recovered from a stagnation-point surface pressure measurement. Higher fidelity formulations can be found in [20].

The thermocouples are assumed to recover the convective heating rate, which also represents a simplification for simulation purposes. Note however, that recent EDL missions have also used dedicated heat flux sensors for direct heating-rate measurements [31, 32]. For this setup, the convective heating rate, assumed to be measured directly, is modeled using the Sutton-Graves relation [33, 34]

$$\tilde{\dot{Q}}_s = k \left(\frac{\rho}{R_n} \right)^{\frac{1}{2}} v^3 + \eta_{\dot{Q}_s}, \tag{6}$$

where $k = 1.9027 \times 10^{-4}$ kg$^{1/2}$m^{-1} for Mars [34], $k = 1.7030 \times 10^{-4}$ kg$^{1/2}$m^{-1} for Earth [33], $R_n = 1.125$ m denotes the vehicle nose radius [35], and $\eta_{\dot{Q}_s}$ represents zero-mean, white, and Gaussian measurement noise. Although the availability of the pressure and heating rate measurements may vary with flight regime, this work assumes available measurements throughout the entire entry phase.

In the case of Earth re-entry, GPS measurements may be available for altitudes $h \geq 80$ km. Here, the GPS is assumed to measure the position of the vehicle in the Earth-centered, Earth-fixed (ECEF) frame, such that [36]

$$\tilde{\boldsymbol{r}}_{\text{GPS}} = \begin{bmatrix} r\cos(\phi)\cos(\theta) & r\cos(\phi)\sin(\theta) & r\sin(\phi) \end{bmatrix}^T + \boldsymbol{\eta}_{\boldsymbol{r}}, \tag{7}$$

where $\boldsymbol{\eta}_{\boldsymbol{r}} \in \mathbb{R}^3$ represents the associated zero-mean, white, and Gaussian measurement noise.

C. Atmospheric Density Models

Different approaches can be used to model atmospheric density. In practice, the true atmospheric density is unknown, which means that the onboard computer does not have access to this exact information. In this work, an exponential model is used as the nominal profile to represent the information available pre-flight, while the NASA Global Reference Atmospheric Models (GRAMs) are used to simulate the true atmospheric density. It is important to note that this setup is a simplification compared to real missions, where the nominal profile is more accurate than an exponential model, and the true atmosphere can behave differently than the GRAMs profiles. However, maintaining a mismatch between the onboard model and the true atmospheric conditions reflects the uncertainty that can exist in real missions.

Given an atmosphere in isothermal equilibrium, the atmospheric density can be described as [37]

$$\hat{\rho}_{\text{exp}} = \rho_0 \exp\left(-\frac{(r - r_p)}{h_s} \right), \tag{8}$$

where ρ_0 is the density at the reference radial position, r_p is the reference radial position, and h_s is the atmospheric scale height. The model defined by Eq. (8) is used in this work as the pre-flight nominal atmospheric density. Note that a single exponential fit is used for the entire altitude range.

GRAMs represent a significant improvement in atmospheric modeling compared to other approaches. These atmospheric models were initially developed to address the need for a reference atmosphere that offers geographical variation, coverage from surface to orbital altitudes, and seasonal variations of thermodynamic variables and wind components [38]. In addition, GRAMs have the ability to simulate spatial and temporal disturbances. Different GRAMs exist for various celestial bodies. Here, Mars-GRAM [39] is used to generate the true trajectories in the Mars entry scenario, and Earth-GRAM [38] is used to generate the true trajectories in the Earth re-entry scenario.

III. Navigation

This section provides a summary of the adaptive filtering strategy in [22]. This summary is included for this work to be self-contained. This adaptive filtering technique consists of two stages: offline training and online adaptation.

A. Offline Training

A neural network is first trained to approximate density with respect to planet-centric radius, such that

$$\hat{\rho} = \mathcal{NN}(r, \boldsymbol{\xi}), \tag{9}$$

where $\boldsymbol{\xi}$ are the parameters of the network. To train the network, training data is generated by using a least-squares exponential fit derived from Mars-GRAM or Earth-GRAM profiles, depending on the entry scenario. The least-squares exponential fit to the GRAM data is used to determine the parameters in Eq. (8). Using this exponential density model, a set number of entry trajectories are simulated and the planet-centric radius and atmospheric density are recorded. The network is then trained to match the density profile provided by the exponential model. It is important to mention that the GRAM profiles are treated as unknown, as they represent the true atmospheric density in this work. The only information assumed to be available pre-flight is the nominal model, which is described by the exponential fit. Although the nominal model is derived from the GRAM profiles, the network is never provided with the profiles themselves.

This training setup is designed to be independent of a specific neural network architecture and pre-flight nominal model. If the network were to be trained with a higher fidelity nominal model (for example, including latitude and longitude dependence), obtaining training data from propagated trajectories ensures that the dataset is representative of the nominal model within the approximate entry trajectory. However, since the nominal model in this work is only a function of planet-centric radius, generating training data using propagated trajectories is equivalent to querying the exponential model at different altitudes along the approximate entry trajectory.

1. Implementation Details

This section describes the specific implementation, used in this work, for offline training of the neural network. As previously mentioned, to generate the training data, a least-squares exponential fit was first derived from the GRAM profiles. Figure 2 shows the GRAM profiles used to obtain this fit for each scenario. The Mars-GRAM and Earth-GRAM profiles were generated by setting the density perturbation scale parameter to 1, representing nominal perturbations [38, 39]. Table 1 shows the exponential density model parameters obtained for each scenario.

Using the exponential fit, 1000 trajectories were simulated for each entry scenario by randomly sampling an initial altitude. The initial altitude was uniformly sampled between 120 km and 125 km, and the trajectories were propagated with a time step of 0.5 seconds until reaching a final altitude of approximately 10 km. For all trajectories, the bank angle was kept constant at $\sigma = 0$ (i.e., no guidance was performed). The planet-centric radius and atmospheric density were recorded at each time step.

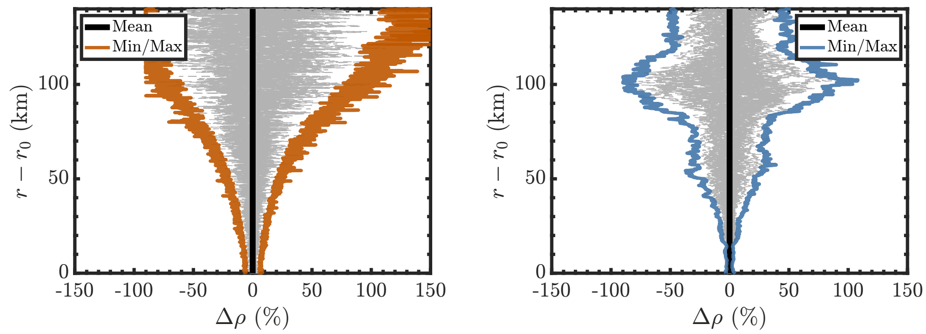

Fig. 2 Density dispersion plots for Mars-GRAM (left) and Earth-GRAM (right). The gray lines show the different profiles.

Table 1 Exponential density model parameters

Parameter	Mars	Earth
ρ_0 (kg/m^3)	0.014	1.221
r_p (km)	3396.190	6378.140
h_s (km)	10.248	8.795

Once the training data was obtained, both the input and output were preprocessed to facilitate training. The network input and output are defined as:

$$i_k = \frac{r_k - \bar{r}}{\sigma_r}, \quad o_k = \frac{\varrho_k - \bar{\varrho}}{\sigma_\varrho}, \tag{10}$$

where $\bar{r}$ is the mean of the training input data, σ_r is the standard deviation, $\bar{\varrho}$ is the mean of the training output data, and σ_ϱ is the corresponding standard deviation. The pseudo-density, ϱ_k, is defined as [19]:

$$\varrho_k = \sqrt{\mathcal{B} - \log_{10} \rho_k}, \tag{11}$$

where $\mathcal{B}$ is a shift ensuring the argument of the square root remains positive. For this work, $\mathcal{B} = 0$ since the atmospheric density is used in SI base units. The density can be recovered by solving for ρ_k in Eq. (11). Note that the scaling variables used to preprocess the training data were saved for use in the online adaptation portion of this work. Therefore, when referring to Eq. (9), this includes both the scaling of the input and output, as well as the recovery of the density.

For this work, a feed-forward network with one hidden layer containing 100 neurons and using tanh as the activation function was used. With the training data preprocessed and the network architecture defined, the network was trained using a 90/10 train/validation split. The Adam optimizer was used to minimize the mean squared error (MSE) between the predicted and nominal atmospheric densities during offline training. The network was trained for 1000 epochs, with a batch size of 10, using a variable learning rate for the Adam optimizer [22]. The accuracy of the trained network was evaluated using the mean absolute percentage error (MAPE) on the validation set. For the Mars scenario, a MAPE of 0.29% was obtained on the validation set, while for the Earth scenario a MAPE of 0.54% was obtained. As the network is adapted online, convergence of the offline training does not strictly rely on an accuracy threshold with respect to the nominal model. Nevertheless, a good fit to the nominal model is desired, and for the two scenarios evaluated in this work, the achieved final accuracy was sufficient.

B. Online Adaptation

Since the exponential fit does not fully capture the true atmospheric density, which in this work is represented by the GRAMs profiles, the network is adapted online to better approximate the true density. Given the dynamics in Section II.A, the filter state is defined as

$$\boldsymbol{x} = \begin{bmatrix} r & \phi & \theta & v & \gamma & \psi & B & L/D \end{bmatrix}^T, \tag{12}$$

where L/D is the lift-to-drag ratio and B is the inverse of the ballistic coefficient, both modeled as a random walk [23]. The random walks for both L/D and B are driven by zero-mean, white, and Gaussian process noise.

The measurement vector is defined as

$$\boldsymbol{y} = \left[(\tilde{\boldsymbol{a}}^b)^T \quad \tilde{q} \quad \tilde{\dot{Q}}_s \;\mid\; (\tilde{\boldsymbol{r}}_{\text{GPS}})^T \right]^T, \tag{13}$$

with the GPS measurement only available for the Earth re-entry problem when the altitude is greater than 80 km. For this section, let the measurement model be denoted as

$$\boldsymbol{y}_k = h\left(\boldsymbol{x}_k, \rho_k\right) + \boldsymbol{\eta}_{\boldsymbol{k}}, \tag{14}$$

where h refers to the model of the available sensors following Section II.B, and $\boldsymbol{\eta}_{\boldsymbol{k}}$ is zero-mean, white, and Gaussian measurement noise with covariance matrix R.

The online adaptation consists of three steps, propagate, adapt, and update. For the first step, the previous posterior at time step $k-1$ is augmented with a consider parameter, such that

$$\hat{\boldsymbol{x}}^+_{a,k-1} = \begin{bmatrix} \hat{\boldsymbol{x}}^{+T}_{k-1} & c_{k-1} \end{bmatrix}^T, \quad P^+_{a,k-1} = \begin{bmatrix} P^+_{k-1} & C^+_{k-1} \\ C^{+T}_{k-1} & P^+_{c,k-1}, \end{bmatrix} \tag{15}$$

where $P^+_{c,0}$ is a tuning parameter, $c_0 = 1$, and $C^+_0 = 0_{n_x \times 1}$. The consider parameter [40, 41] is used to disperse the atmospheric estimate to account for its uncertainty, such that in the propagation and update

$$\hat{\rho}_{k-1} = c_{k-1} \cdot \mathcal{NN}\left(r_{k-1}, \boldsymbol{\xi}_{k-1}\right). \tag{16}$$

After this, the state and covariance are propagated to time step k, to obtain the prior estimate, that is, $\hat{\boldsymbol{x}}_{a,k}^{-}$ and $P_{a,k}^{-}$. This is done via an unscented Kalman filter [42] propagation, using the system dynamics in Section II.A, and modeling the consider parameter as an exponentially correlated random variable (ECRV) centered at one.

When a measurement is obtained, the log likelihood of the measurement is considered as a loss function for the parameters of the neural network, such that

$$\mathcal{L}\left(\xi_{k-1}\right) = \left[\boldsymbol{y}_k - h\left(\hat{\boldsymbol{x}}_k^{-}, \hat{\rho}_{k-1}\right)\right]^T R^{-1} \left[\boldsymbol{y}_k - h\left(\hat{\boldsymbol{x}}_k^{-}, \hat{\rho}_{k-1}\right)\right], \quad \hat{\rho}_{k-1} = \mathcal{NN}\left(\hat{\boldsymbol{x}}_k^{-}(1), \boldsymbol{\xi}_{k-1}\right), \tag{17}$$

where $\hat{\boldsymbol{x}}_k^{-}$ is the prior state estimate obtained from the augmented state, and $\hat{\boldsymbol{x}}_k^{-}(1)$ represents the first element of $\hat{\boldsymbol{x}}_k^{-}$, i.e., planet-centric radius. Therefore, the optimal network parameters are found by

$$\boldsymbol{\xi}_k = \arg\min_{\xi_{k-1}} \mathcal{L}\left(\boldsymbol{\xi}_{k-1}\right). \tag{18}$$

To minimize the loss function, the Adam optimizer [43] and back-propagation are used. It is important to note that the optimizer parameters, namely the gradient and the squared gradient, $\bar{\nabla}_k$ and $(\bar{\nabla}_k)^2$, are also adapted and saved for the next iteration. As it can be seen, the Adam optimization is performed at each navigation call, i.e., every time a new measurement is available. For this work, the optimization iterates until the loss reaches a predefined threshold, and a maximum patience is used to avoid redundant steps. The parameters for the optimization step used in this work are the same as those used in [22], with the only difference being that the loss threshold for the Earth re-entry scenario is increased from 1 to 10.

Once the adapted parameters of the network have been obtained, a Schmidt-Kalman update [40, 41, 44] is performed to obtain a posterior augmented state estimate, that is, $\hat{\boldsymbol{x}}_{a,k}^{+}$ and $P_{a,k}^{+}$, using the measurement models in Section II.B. Figure 3 shows a flowchart of the entire navigation scheme.

IV. Guidance

This section summarizes the version of FNPEG used in this work and introduces the new guidance solution incorporating the adaptive density model into the prediction step. In FNPEG, an NPC framework is used to calculate the bank angle by minimizing the range-to-go along the great-circle connecting the final location and the target location, at a set final energy. In this framework, the longitudinal and lateral channels can be decoupled, meaning that the magnitude of the bank angle can be calculated based on 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 [7].

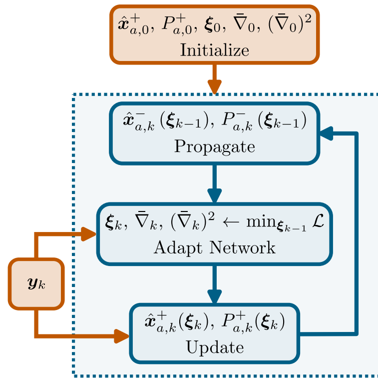

Fig. 3 Flowchart of the navigation algorithm. Figure adapted from [22].

A. 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 target energy e^*, predetermined by a reference trajectory,

$$e^* = \frac{1}{r_n^*} - \frac{v_n^*}{2}, \tag{19}$$

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 respect to energy [6]

$$\frac{dr_n}{de} = \frac{\sin(\gamma)}{D_n}, \quad \frac{d\gamma}{de} = \frac{1}{D_n v_n^2}\left[L_n \cos(\sigma) + \left(v_n^2 - \frac{1}{r_n}\right)\left(\frac{\cos(\gamma)}{r_n}\right)\right], \quad \frac{ds}{de} = -\frac{\cos(\gamma)}{r_n D_n}, \tag{20}$$

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

$$r_n = \frac{r}{r_p}, \quad v_n = \frac{v}{\sqrt{r_p g_p}}, \quad L_n = \frac{L}{g_p}, \quad D_n = \frac{D}{g_p}, \tag{21}$$

which follows from normalizing time by $\sqrt{r_p/g_p}$. For these equations, r_p refers to the mean radius of Mars or Earth, and g_p is the gravitational acceleration at r_p.

With this setup, the bank angle is found by minimizing the square of the integrated final range-to-go s_f^*,

$$\sigma^* = \arg\min_{\sigma} \frac{1}{2} s_f^*(e^*, \sigma)^2. \tag{22}$$

In addition, a linear parameterization of the bank angle profile, with respect to energy, is used in the prediction step to increase the robustness of the algorithm, following [6, 7]:

$$\sigma = \sigma^* + \frac{e - e_0}{e^* - e_0}\left(\sigma_f - \sigma^*\right), \tag{23}$$

where e_0 is the current energy, e is the energy along the trajectory, and σ_f is a user-defined parameter. Note that other parameterizations have also been studied, such as using exponential [45] or logistic [10] functions, or parameterizing the bank angle with respect to the range [11].

To find σ^*, the Newton-Raphson algorithm is used, where

$$\sigma_{i+1} = \sigma_i - \lambda_i \frac{s_f^* \left(e^*, \sigma_i\right)}{\partial s_f^* \left(e^*, \sigma_i\right) / \partial \sigma}, \tag{24}$$

with $\partial s_f^* \left(e^*, \sigma_i\right) / \partial \sigma$ calculated using numerical differentiation and λ_i chosen to satisfy the Armijo condition [46]. The algorithm stops when $|s_f^* \left(e^*, \sigma_i\right)|$ reaches a specified threshold or when a set number of iterations have been completed.

B. Lateral Channel

The longitudinal channel only determines the magnitude of the bank angle. To determine the correct sign, the lateral logic described by Smith [47] is adopted. Once the magnitude of the bank angle is determined, the current state is integrated over time using the full 3-DOF equations, considering both signs of the bank angle, until the final target energy is reached. 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 current bank angle sign, and let χ_f^- represent the final crossrange error for the solution integrated with the opposite bank angle sign. Then, if the ratio of the crossrange errors is greater than a threshold K, such that

$$\left| \frac{\chi_f^+}{\chi_f^-} \right| > K, \tag{25}$$

a bank reversal is commanded [47]. 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.

C. Spherical Geometric Definitions

As FNPEG requires the calculation of crossrange error, range-to-go, and, for validation purposes, downrange error, it is important to address how these quantities are calculated in this work. Figure 4 shows the geometric definitions of these errors with respect to the planetary surface [8, 48].

Using the spherical law of sines, it can be seen that the final crossrange error can be calculated as

$$\chi_f = \arcsin\left(\sin\left(i\right)\sin\left(d\right)\right), \tag{26}$$

where d is the great-circle distance between the initial state and the final state and i is the angle between d and the arc

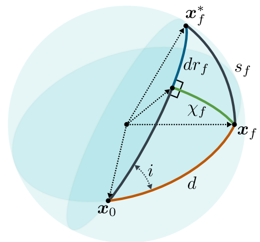

Fig. 4 Final crossrange error, downrange error, and range-to-go definitions. Figure adapted from [48].

connecting the initial state and the target state, both defined as

$$d = \arccos\left(\sin(\phi_0)\sin(\phi_f) + \cos(\phi_0)\cos(\phi_f)\cos(\theta_f - \theta_0)\right), \quad i = \arccos\left(\left[\frac{\boldsymbol{x}_0 \times \boldsymbol{x}_f}{||\boldsymbol{x}_0 \times \boldsymbol{x}_f||}\right] \cdot \left[\frac{\boldsymbol{x}_0 \times \boldsymbol{x}_f^*}{||\boldsymbol{x}_0 \times \boldsymbol{x}_f^*||}\right]\right). \quad (27)$$

The final downrange error is calculated by using the spherical law of cosines, such that

$$dr_f = s_0 - \arccos\left(\frac{\cos(d)}{\cos(\chi_f)}\right), \quad s_0 = \arccos\left(\sin(\phi_0)\sin\left(\phi_f^*\right) + \cos(\phi_0)\cos\left(\phi_f^*\right)\cos\left(\theta_f^* - \theta_0\right)\right). \quad (28)$$

where s_0 is the initial range-to-go. And the final range-to-go is calculated as

$$s_f = \arccos\left(\sin(\phi_f)\sin\left(\phi_f^*\right) + \cos(\phi_f)\cos\left(\phi_f^*\right)\cos\left(\theta_f^* - \theta_f\right)\right). \quad (29)$$

Note that the calculated final range-to-go will only yield positive values. Therefore, to assess trajectory overshooting, the downrange error serves as a more informative metric. Based on Eq. (28), a positive final downrange error indicates undershooting, while a negative downrange error indicates overshooting (assuming the crossrange error is small). It is also important to distinguish the difference between s_f and s_f^*. While s_f is determined using the geometric definitions presented, s_f^* is obtained by integrating longitudinal dynamics.

D. Propagating the Trajectory

The guidance solution described involves propagating the current state estimate to a final state defined by the target energy. Therefore, the novelty of this work relies on the atmospheric density model used to propagate the guidance trajectories. Since the neural network is being adapted to locally fit the current density, there is no guarantee that the adapted network will be a good fit at other altitudes. However, the exponential fit, due to its construction, is a good fit at lower altitudes. Therefore, the two can be paired to account for possible degradation in the fit outside the local adaptation region. Thus, for the prediction step (and lateral channel propagation), this work uses a weighted linear combination of the neural network output and the exponential model, formulated as follows

$$\rho(r, e) = \alpha(e)\,\mathcal{NN}(r, \boldsymbol{\xi}) + [1 - \alpha(e)]\,\rho_{\exp}(r), \quad \frac{d\alpha}{de} = -\kappa\frac{\alpha}{v_n}, \quad (30)$$

where α is an exponentially decaying factor with respect to energy, with κ controlling the exponential decay rate.

For every propagation $\alpha(e_k) = 1$. This implies that a high value of κ will result in a very rapid exponential decay of α, suggesting that the guidance algorithm will rely more on the exponential atmospheric model than on the neural network. In contrast, lower values of κ will make the guidance algorithm trust the neural network for a longer duration along the propagation of the trajectory.

Two aspects of the linear combination are worth noting. First, varying the parameter κ allows the neural network to be gradually introduced into the FNPEG prediction step. This enables a controlled study of how increasing reliance on the neural network affects targeting accuracy. The choice of κ, however, remains heuristic and application-dependent, as discussed in later sections (the tuning of κ could be informed by prior knowledge of the accuracy of the nominal model). Second, the linear combination adds modularity. The exponential model can easily be replaced with, for example, a corrected model used in real missions like Orion's skip re-entry [5]. This allows for the testing of different model combinations to identify those that work best together.

V. Numerical Examples

This section presents two numerical examples used to test the adaptive density model in different scenarios. The first is a Mars entry case, representing a direct entry. Three guidance algorithms are compared: FNPEG with access to the true state and density, FNPEG with the estimated state and exponential model, and FNPEG with the estimated state and linear combination. Each serves a different purpose. The use of the true state and density sets a baseline for the best possible performance. The use of the exponential model shows the issues that can arise when relying on an unadapted model. The linear combination is used to validate the integration of the network in the guidance loop.

After validating the neural network in the Mars entry scenario, a second case is considered: an Earth skip re-entry. This example is meant to compare the neural network with existing state-of-the-art adaptation techniques. Three guidance algorithms are also tested. FNPEG with access to the true state and density is used to establish a performance baseline. The second uses the adaptation method applied for Orion, where a first-order fading-memory filter is used to estimate a bias to the nominal model. The third combines the neural network with Orion's adaptation through the linear combination. The goal for this example is to show how the network can improve and complement existing techniques.

Before presenting the numerical examples, it is important to review the use of the different atmospheric density models. For the truth, the density model used are the GRAM profiles. For navigation, the density model used is the adaptive neural network. And for the new guidance algorithms, the density model used is a linear combination of the nominal model and the adaptive neural network. All other parameters are kept consistent across the truth, navigation, and guidance. This is done to isolate the effect of the atmospheric density model, since it is the main focus of the paper.

A. Mars Entry

This section presents a Monte Carlo (MC) test for a Mars entry mission. The test involves simulating 1000 different trajectories, where the state is estimated using the presented navigation scheme, and the guidance commands are calculated using FNPEG and the new adaptive density model. Each true trajectory is simulated using a different Mars-GRAM density profile. For each MC run, a full density profile sampled every 0.1 km is generated by querying Mars-GRAM offline. Cubic spline interpolation is then applied to this profile to estimate density values at arbitrary altitudes during the simulation for the truth. This approach reduces computational cost by avoiding repeated calls to Mars-GRAM at each time step and for each MC run. It is important to note that the least-squares exponential model, used to train the neural network offline and to propagate the guidance trajectories, is fitted to different Mars-GRAM profiles that were not used in the MC testing. However, in both cases, the Mars-GRAM profiles were generated by setting the density perturbation scale parameter to 1.

1. Monte Carlo Settings

This work uses the same MC settings as detailed in the previous work [22], with the only difference being that the vehicle is given slightly more lift to improve controllability. The initial conditions, obtained from [23, 49, 50] are described in Table 2.

Table 2 Entry conditions for the Mars entry scenario

Parameter	**Value**	**Standard Deviation** (3σ)
r_0 (m) [49]	3.5222×10^6	3.2066×10^1
ϕ_0 (deg) [49]	-0.3919×10^1	7.8100×10^{-4}
θ_0 (deg) [49]	1.2672×10^2	3.6700×10^{-4}
v_0 (m/s) [49]	6.0833×10^3	2.6059×10^{-2}
γ_0 (deg) [49]	-1.5489×10^1	4.0000×10^{-4}
ψ_0 (deg) [49]	9.3206×10^1	2.6800×10^{-4}
B_0 (m^2/kg) [23, 50]	7.1000×10^{-3}	4.8000×10^{-3}
L_0/D_0 (n.d.)	2.7000×10^{-1}	1.5178×10^{-1}

Each trajectory and measurement is corrupted by additive, zero-mean, white Gaussian noise. Tables 3 and 4 show the values used for the process and measurement noise (obtained from [28, 51]), respectively. For this scenario, the sensors are sampled at a frequency of 4 Hz. The consider parameter is modeled as an ECRV with $\tau = 5$, $P_{ss} = 1 \times 10^{-3}$, and $P_{c,0} = 1 \times 10^{-10}$. Furthermore, for each simulated trajectory, the angle of attack is kept constant at $\alpha = -17°$.

The final target state is obtained by propagating a nominal trajectory using the full 3-DOF equations with a randomly selected Mars-GRAM profile from the training set. The nominal trajectory is integrated using a bank angle of $\sigma = -50°$ when $v(t) \geq 6000$ m/s, $\sigma = -40°$ when $v(t) \leq 3000$ m/s, and a linear decay for $3000\,\text{m/s} \leq v(t) \leq 6000$ m/s (with the sign reversed within this range). This trajectory is propagated until a final velocity of $v_f = 1100$ m/s is reached. This

Table 3 Process noise statistics for the Mars entry scenario

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

Table 4 Measurement noise statistics for the Mars entry scenario

Parameter	Standard Deviation (3σ)
$\tilde{\boldsymbol{a}}^b$ (μg) [51]	3×10^2
$\tilde{q}$ (% of reading) [28]	1
$\tilde{\dot{Q}}_s$ (% of reading)	1

velocity marks the transition from range control to heading alignment [52]. Table 5 shows the final target state and Fig. 5 shows the nominal altitude as a function of range-to-target, the nominal bank angle as a function of time, and the nominal flight trajectory that ends close to the Gale Crater. Note that this is a direct entry.

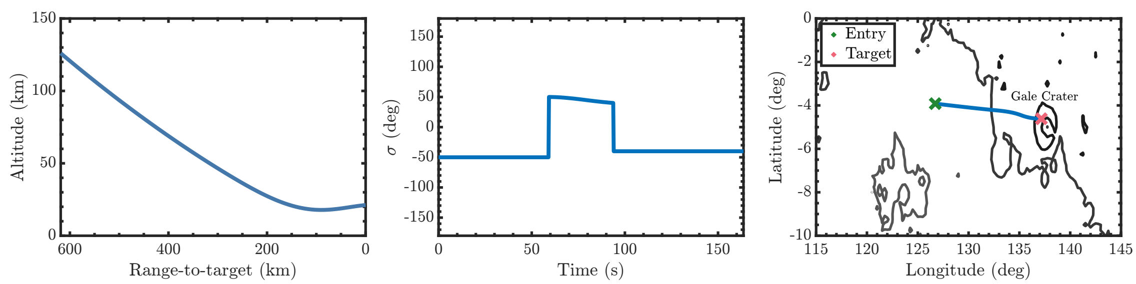

Fig. 5 Altitude as a function of range-to-target (left), bank angle as a function of time (center), and flight trajectory (right) for the Mars entry scenario.

2. Guidance Algorithms

As mentioned before, to assess the accuracy of the adaptive guidance algorithm, three different algorithms are tested and compared:

1) The first algorithm, referred to as true guidance (TG), has access to the true vehicle state and the true atmospheric density. This means that navigation errors are not considered. This algorithm serves as a baseline.

2) The second algorithm, referred to as guidance with exponential model (GEXP), uses the state estimates from the navigation filter but uses only the exponential density model in the prediction step of FNPEG (i.e. $\alpha = 0 \, \forall \, e$ in

Table 5 Final target state for the Mars entry scenario

Parameter	Value
r_f^* (m)	3.4173×10^6
ϕ_f^* (deg)	-0.4614×10^1
θ_f^* (deg)	1.3716×10^2
v_f^* (m/s)	1.1000×10^3

Eq. (30), or equivalently $\kappa \to \infty$).

3) The third algorithm, referred to as guidance with adaptive model (GADA), uses the state estimates of the filter and the adaptive density model as in Eq. (30).

For each guidance algorithm, 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) [52, 53]. Before range control starts, the vehicle flies at a constant bank angle of $\sigma = -50°$. For the longitudinal channel, the Newton-Raphson algorithm is used until one of the following conditions is met: $|s_f^*(e^*, \sigma_i)| \leq 0.1$ km or 10 iterations have been completed. For the lateral channel, $K = 3.5$, and bank reversals are performed following a minimum time solution. For the prediction step and lateral channel propagation, $\sigma_f = \sigma^*$, so that the bank angle follows a constant profile during the prediction step.

3. Monte Carlo Results

To compare the three solutions, the final crossrange error, the downrange error, and the range-to-go error are used [8]. Figure 6 shows (on the left) the final crossrange and downrange errors for TG, GEXP, and GADA using two different values of κ. From the figure, it can be seen that GEXP has the highest downrange error, landing mostly outside a 2.5 km radius from the target. As the prediction step begins to rely more on the neural network, that is, as the value of κ decreases, the landing profile moves closer to the TG results. This behavior shows that the online adaptation of the neural network can capture the current atmospheric density values more accurately than the pre-flight nominal model. In this entry scenario, relying solely on the neural network for the prediction step ($\kappa = 0$) yields the closest performance to the TG results. This outcome is expected because the exponential model does not capture the true atmospheric density at high altitudes, showing that relying on inaccurate nominal models during prediction can be problematic.

It is important to note that the increased crossrange variance seen in both variants of GADA is an artifact of the lateral guidance logic. The version of FNPEG used in this work is designed to minimize the final range-to-go, assuming that the crossrange error remains within a specified tolerance. As a result, it does not explicitly minimize lateral dispersion, which can lead to increased variance in the crossrange direction. For this example, the lateral threshold, K, is held constant across all algorithms. However, there is likely an optimal K for each specific algorithm. In this case, the chosen value of K happens to favor GEXP, resulting in tighter crossrange dispersion. While tuning K individually for each algorithm could potentially reduce the crossrange variance, this would require additional optimization and is outside the scope of the current work.

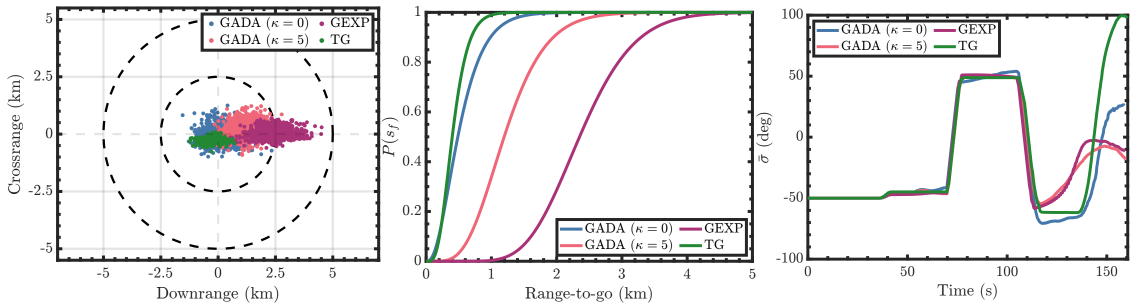

Fig. 6 Crossrange error vs. downrange error (left), final range-to-go (center), average bank angle profile (right) for the Mars entry scenario.

In the center of Fig. 6, the final range-to-go cumulative densities for the three guidance solutions is shown. From this figure, it can be seen that as the prediction step (and lateral channel propagation) increasingly uses the neural network throughout the propagation of the trajectory, the final range-to-go begins to align with the TG solution. Note that for this example, the exponential model is not being adapted and does not accurately represent the atmospheric density conditions in which the vehicle is flying.

The right panel of Fig. 6 shows the average bank angle as a function of time for the three guidance solutions. For GEXP, the trajectory adopts slightly higher bank angles for $t \in [70, 110]$ s, indicating that the nominal model underestimates the true density at this point. This results in faster energy depletion, which leads the vehicle to undershoot the target. For GADA, the bank profile is closer to TG, showing that the atmospheric estimate is closer to the true density. This allows the trajectory to align more closely with the TG solution, ending closer to the target. Note that some energy depletion could be mitigated by using $\sigma_f \neq \sigma^*$, but as shown in the next section, the linear bank angle profile cannot fully compensate for atmospheric mismatch.

As mentioned previously, the navigation technique was developed without considering control inputs [22]. This current work also serves as a sanity check to verify the filter's consistency when its state estimates are used for guidance. Figure 7 shows the estimation error as a function of time. While only 100 MC 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 MC runs.

Table 6 summarizes the average runtime of each algorithm stage for a single trajectory (MATLAB 2025b, M1 MacBook Pro, 16 GB RAM). The average runtime spent at each navigation call, including filter propagation, ML optimization, and filter update, is reported as *Nav. Runtime*. The average number of ML optimization steps per navigation call is reported as *ML Iter*. The average runtime spent at each guidance call is reported as *Gui. Runtime*, and

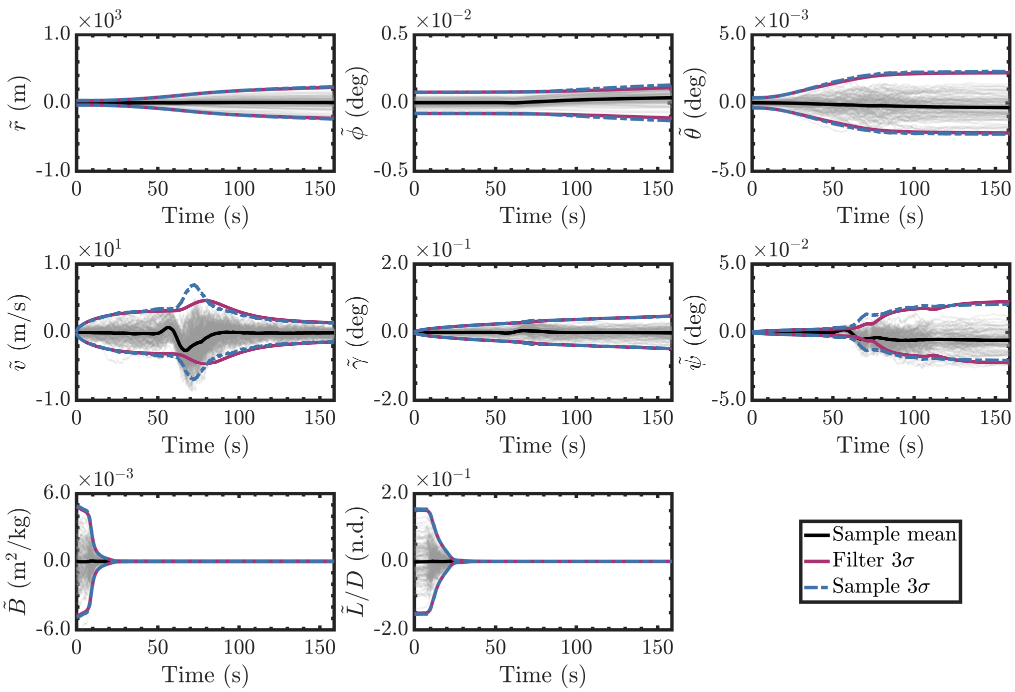

Fig. 7 Estimation error as a function of time for the Mars entry scenario. The gray lines show 100 MC trajectories out of the 1000 runs.

the average number of prediction and correction steps in FNPEG per guidance call is reported as *FNPEG Iter.* Note that no navigation runtime is reported for TG, as it uses the true state, and only the results for GADA ($\kappa = 0$) are presented, as this configuration performs closest to TG.

Comparing the guidance runtime for TG and GADA shows that, although TG has fewer average iterations, GADA achieves a faster guidance runtime. This indicates that using the neural network for propagating trajectories is faster than interpolating the GRAM profiles, as done in TG. For GADA, the navigation runtime is approximately one order of magnitude slower than the guidance runtime. This is consistent with the higher number of iterations in the ML optimization compared to FNPEG. A maximum number of iterations could be imposed to further bound the ML optimization steps. However, determining the optimal maximum number steps is left as future work.

Table 6 Timing results for the Mars entry scenario.

Algorithm	Nav. Runtime (s)	ML Iter.	Gui. Runtime (s)	FNPEG Iter.
TG	-	-	0.021	1.441
GADA ($\kappa = 0$)	0.097	23.327	0.008	2.357

B. Earth Re-Entry

The linear combination in Eq. (30) provides the added advantage of modularity, enabling the exponential density model to be replaced with a different model for improved accuracy. In this section, a MC test with 1000 runs, for an

Earth skip re-entry scenario is presented. Additionally, the exponential model is improved with the atmospheric density adaptation used in the Orion skip re-entry mission [3, 5]. The purpose of this entry scenario is to compare the proposed network filtering approach with state-of-the-art adaptation techniques, and show how they can complement each other. As in the Mars entry scenario, each true trajectory is simulated using a different Earth-GRAM density profile. The GRAM profiles, used for the truth, are also interpolated to approximate density values at the queried altitudes.

For this setup, the linear combination used to propagate the guidance trajectories is expressed as

$$\rho\left(r,e\right) = \alpha\left(e\right) \mathcal{NN}\left(r, \boldsymbol{\xi}_k\right) + \left[1-\alpha\left(e\right)\right] \hat{\mathcal{K}}_k \rho_{\text{exp}}\left(r\right), \tag{31}$$

where $\hat{\mathcal{K}}_k$ is a correction factor determined using a first-order fading-memory filter [3]

$$\rho_{\text{est},k} = 2\frac{D_k}{\hat{B}_k \hat{v}_k^2}, \quad D_k = \frac{||\tilde{\boldsymbol{a}}_k^{\boldsymbol{v}}||}{\sqrt{\left(\hat{L}/D_k\right)^2+1}}, \quad \hat{\mathcal{K}}_k = k_{\text{gain}}\frac{\rho_{\text{est},k}}{\rho_{\text{exp}}\left(\hat{r}_k\right)} + \left(1-k_{\text{gain}}\right)\hat{\mathcal{K}}_{k-1}, \tag{32}$$

where k_{gain} is a user-defined gain. The estimates for $\hat{B}_k$, $\hat{v}_k$, and $\hat{L}/D_k$ are obtained from the navigation filter, meaning these quantities do not need to be estimated using first-order fading-memory filters as in previous approaches [11]. As with the Mars entry example, the least-squares exponential model, used to train the neural network offline and to propagate the guidance trajectories, is fitted to different Earth-GRAM profiles that were not used in the MC testing. However, in both cases, the Earth-GRAM profiles are generated by setting the density perturbation scale parameter to 1.

1. Monte Carlo Settings

The initial conditions used for this scenario, as shown in Table 7, correspond to the entry conditions for the Orion entry capsule [4, 5, 54]. The entry trajectories are simulated following the same procedure as for the Mars entry scenario, meaning that the dynamics are propagated with additive process noise and the measurements are obtained from the true state, corrupted with additive process noise as well. The values used for the process and measurement noise (obtained from [28, 36, 51]) are reported in Table 8 and Table 9, respectively. For this scenario, the sensors are sampled at a frequency of 2 Hz. The consider parameter is also modeled as an ECRV with $\tau = 1$, $P_{ss} = 1 \times 10^{-3}$, and an initial covariance of $P_{c,0} = 1 \times 10^{-10}$. The angle of attack also remains constant at $\alpha = -17°$.

In this case, the nominal trajectory is integrated using a bank angle of $\sigma = -75°$ when $v(t) \geq 7500$ m/s, $\sigma = -30°$ when $v(t) \leq 4500$ m/s, and a linear decay for 4500 m/s $\leq v(t) \leq 7500$ m/s (with the sign reversed within this range). For this re-entry scenario, the trajectory is propagated until a final velocity of $v_f = 304.8$ m/s is reached, which marks the transition from range control to heading alignment [3]. Table 10 shows the final target state and Fig. 8 shows the nominal altitude as a function of range-to-target, the nominal bank angle as a function of time, and the nominal flight trajectory that ends close to the western coast of Baja California.

Table 7 Entry conditions for the Earth re-entry scenario

Parameter	Value	Standard Deviation (3σ)
r_0 (m) [4, 5]	6.5001×10^6	1.5240×10^2
ϕ_0 (deg) [4, 5]	-2.5829×10^1	2.0000×10^{-1}
θ_0 (deg) [4, 5]	-1.2008×10^2	2.0000×10^{-1}
v_0 (m/s) [4, 5]	1.0992×10^4	2.7432×10^1
γ_0 (deg) [4, 5]	-0.5664×10^1	1.0000×10^{-1}
ψ_0 (deg) [4, 5]	0.4654×10^1	0.5000×10^{-1}
B_0 (m^2/kg) [4, 54]	3.1000×10^{-3}	4.8000×10^{-3}
L_0/D_0 (n.d.) [4, 54]	3.2500×10^{-1}	1.5178×10^{-1}

Table 8 Process noise statistics for the Earth re-entry scenario

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

2. Guidance Algorithms

For this re-entry scenario, three algorithms are also used to assess the accuracy of the adaptive atmospheric density model. Instead of using the nominal exponential model, GEXP uses the adaptation presented at the beginning of the section and is therefore renamed guidance with Orion adaptation (GORI). Consequently, GADA now represents a linear combination of the neural network and the adapted exponential model. The guidance algorithms are summarized below.

1) The first algorithm, TG, represents the solution with access to the true state of the vehicle and the true density.
2) The second algorithm, GORI, uses the state estimates from the filter but uses only the adapted exponential density model in the prediction step of FNPEG (i.e. $\alpha = 0 \,\forall\, e$ in Eq. (31)). The adaptation is only performed when the sensed acceleration is greater than 0.05 times Earth's gravity ($||\tilde{\boldsymbol{a}}^b|| > 0.49$ m/s^2) [3], with $k_{\text{gain}} = 0.9$.
3) The third algorithm, GADA, uses the state estimates from the filter and the adaptive density model as in Eq. (31).

For each guidance algorithm, the maximum rate-of-change of the bank angle is set to 15°/s, and FNPEG is called at a frequency of 1 Hz once $||\tilde{\boldsymbol{a}}^b|| > 0.49$ m/s^2 [3]. Before range control starts, the vehicle flies at a constant bank angle of $\sigma = -75°$. For the longitudinal channel, the Newton-Raphson algorithm is used with the same convergence conditions as for the Mars entry scenario. For calculations of the lateral channel, $K = 3.5$, and bank reversals are also performed following a minimum time solution. For the prediction step and the propagation of the lateral channel, $\sigma_f = 55°$, so that

Table 9 Measurement noise statistics for the Earth re-entry scenario

Parameter	Standard Deviation (3σ)
$\tilde{\boldsymbol{a}}^b$ (μg) [51]	2×10^2
$\tilde{q}$ (% of reading) [28]	1
$\tilde{\dot{Q}}_s$ (% of reading)	1
$\tilde{\boldsymbol{r}}_{\mathrm{GPS}}$ (m) [36]	18

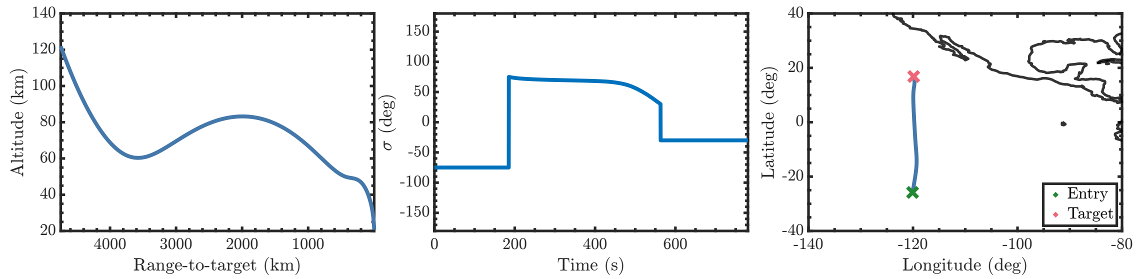

Fig. 8 Altitude as a function of range-to-target (left), bank angle as a function of time (center), and flight trajectory (right) for the Earth re-entry scenario.

the bank angle follows a linear profile in the prediction step.

3. Monte Carlo Results

The left panel of Figure 9 shows the final crossrange and downrange error for the three guidance solutions, and the center panel shows the final range-to-go cumulative densities. The results obtained with GORI show that linearly adapting the nominal model helps bring the vehicle closer to the target, but it still undershoots the trajectory. In the case of GADA with $\kappa = 0$, the landing profile moves closer to the TG solution. However, for this particular re-entry scenario, the combination of both models, $\kappa = 5$, produces results closest to the TG solution.

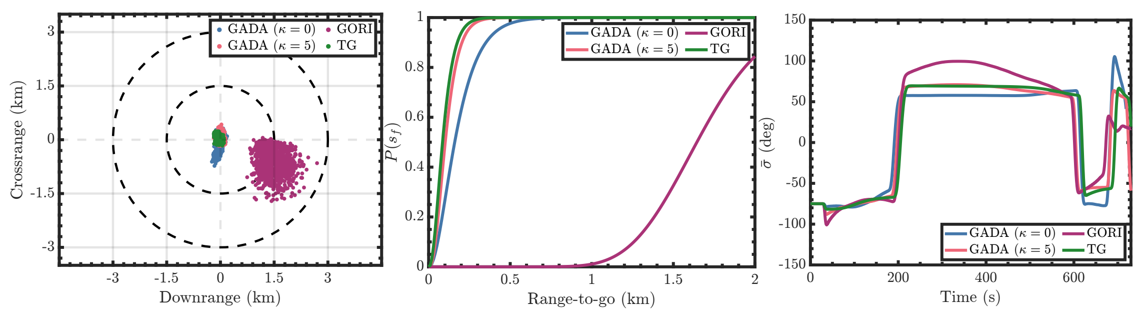

Fig. 9 Crossrange error vs. downrange error (left), final range-to-go (center), average bank angle profile (right) for the Earth re-entry scenario.

The right panel of Fig. 9 shows the average bank angle as a function of time for the three guidance solutions. Similarly to the results obtained for the Mars entry scenario, for GORI, the adapted model still underestimates the true density, causing the vehicle to fly at higher bank angles (specifically between $t \in [200, 600]$ s), depleting energy

Table 10 Final target state for the Earth re-entry scenario

Parameter	Value
r_f^* (m)	6.3989×10^6
ϕ_f^* (deg)	1.6799×10^1
θ_f^* (deg)	-1.1986×10^2
v_f^* (m/s)	3.0480×10^2

faster and, consequently, undershooting the target. Note that in this case, the linear bank angle profile is used, but the mismatch in atmospheric density still leads to faster energy depletion. In the case of GADA with $\kappa = 0$, the atmospheric estimate slightly overestimates the true density, causing the vehicle to fly at shallower bank angles. This results in energy depletion occurring closer to the target. For GADA with $\kappa = 5$, the effect of both adaptations is compensated, and the bank angle profile follows the TG solution, landing closer to the target.

It is important to note that the targeting error obtained with GORI could be reduced by improving the onboard nominal model to better capture the true atmospheric density. Common onboard models used for Earth re-entry include the 1976 U.S. Standard Atmosphere model [3]. In such cases, the network could also be trained to match these models. Nevertheless, this is not the primary goal of this paper. The objective of this paper is to show that this new adaptive method can adapt to mismodeled dynamics more effectively than existing methods, improving the prediction step in NPC algorithms. The results shown in this section suggest that common adaptation schemes may have a smaller domain of adaptation convergence and, therefore, require initialization closer to the truth than the neural network approach.

Fig. 10 shows the estimation errors as a function of time. Similarly to the Mars entry scenario, the filter is consistent, with the estimation errors staying close to zero mean and the filter's covariance aligning with the variance of the MC errors. In particular, the error decreases during the skip phase of the trajectory as GPS measurements are used.

Table 11 Timing results for the Earth re-entry scenario.

Algorithm	Nav. Runtime (s)	ML Iter.	Gui. Runtime (s)	FNPEG Iter.
TG	-	-	0.161	3.094
GADA ($\kappa = 5$)	0.049	9.870	0.012	2.047

Table 11 reports the average runtime for the different algorithm stages in a single trajectory (MATLAB 2025b, M1 MacBook Pro, 16 GB RAM). No navigation runtime is reported for TG, as it uses the true state, and only the results for GADA ($\kappa = 5$) are presented, as it performs closest to TG. The guidance runtime shows the same trend as in the Mars scenario, where GADA is faster than TG, indicating that interpolating GRAM profiles is more expensive than using the linear combination for propagation. The navigation runtime for GADA is slightly lower than in the Mars scenario, due to fewer average iterations in the ML optimization step. This is due to the increase of the loss threshold from 1 to 10. As in the previous case, the threshold could be replaced by a maximum number of iterations to reduce runtime.

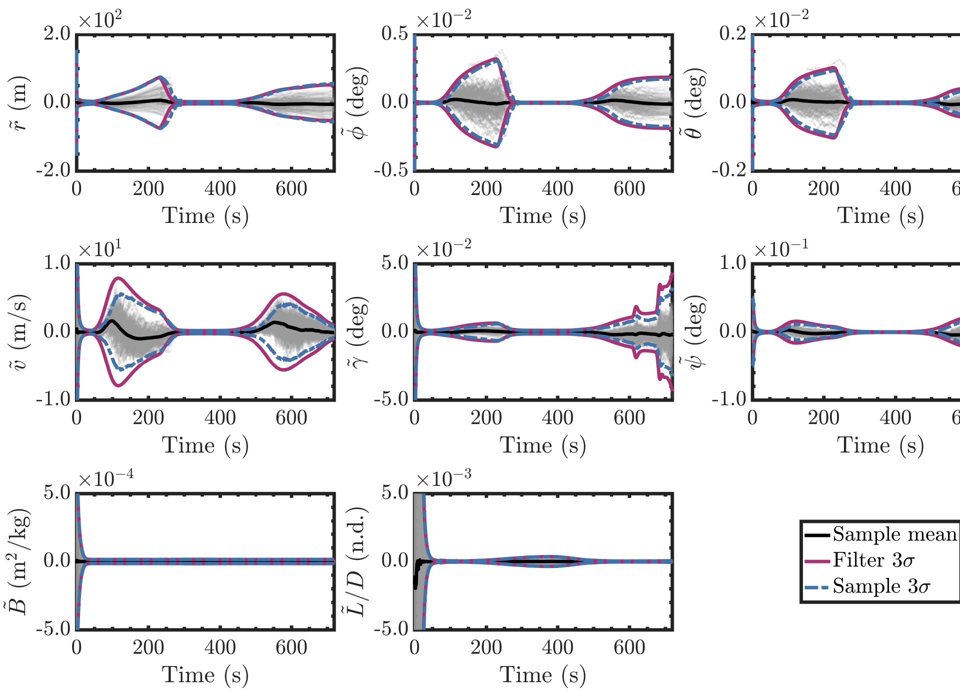

Fig. 10 Estimation error as a function of time for the Earth skip re-entry scenario. The gray lines show 100 MC trajectories out of the 1000 runs.

VI. Conclusions

In this work, a new atmospheric entry adaptive guidance algorithm is presented. This algorithm combines the adaptive neural network density model from a recently developed navigation solution [22] with an onboard nominal model. The neural network is adapted online each time a measurement is obtained, by minimizing the measurement residuals. As a result, the network locally fits the atmospheric density the vehicle is flying through. By linearly combining the neural network with the nominal model, the atmospheric density estimate used in the prediction step and lateral channel propagation in the Fully Numerical Predictor-Corrector Entry Guidance (FNPEG) algorithm is improved. Incorporating this adaptive model into FNPEG results in better targeting precision.

The proposed algorithm was tested through a Monte Carlo analysis in two different atmospheric entry scenarios. For the first scenario, a Mars entry mission was considered. In this use case, the neural network was coupled with an exponential density model. The results showed that using the neural network for propagation can increase targeting accuracy as the network is adapted online to better represent the true atmospheric density. For the second scenario, an Earth re-entry mission was considered. For this application, the neural network was coupled with an online adaptive exponential model, representative of the techniques used in real re-entry missions. The results showed that using a combination of the two models can increase targeting accuracy, as both models are adapted simultaneously online.

In addition, the consistency of the navigation solution was evaluated in a closed-loop format. In both entry scenarios, the filter was shown to be consistent and robust to control inputs determined using the adaptive guidance algorithm.

Acknowledgment

This material is based on research sponsored by Air Force Office of Scientific Research (AFOSR) under award number: FA9550-22-1-0419.

References

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

[2] Braun, R. D., and Manning, R. M., "Mars exploration entry, descent and landing challenges," *2006 IEEE Aerospace Conference*, 2006, pp. 1–18. https://doi.org/10.1109/AERO.2006.1655790.

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

[4] Rea, J., and Putnam, Z., "A Comparison of Two Orion Skip Entry Guidance Algorithms," *AIAA Guidance, Navigation and Control Conference and Exhibit*, 2007, pp. 1–19. https://doi.org/10.2514/6.2007-6424.

[5] Rea, J., McNamara, L., and Kane, M., "Orion Artemis I Entry Performance," *AAS Guidance, Navigation and Control (GN&C) Conference*, 2024, pp. 1–19.

[6] 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.

[7] 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.

[8] Lee, Y., Lee, D. Y., and Wie, B., "A comparative study of entry guidance for Mars robotic and human landing missions," *Acta Astronautica*, Vol. 215, 2024. https://doi.org/10.1016/j.actaastro.2023.11.045.

[9] Ridderhof, J., Tsiotras, P., and Johnson, B. J., "Stochastic Entry Guidance," *Journal of Guidance, Control, and Dynamics*, Vol. 45, No. 2, 2022, pp. 320–334. https://doi.org/10.2514/1.G005964.

[10] Lee, Y., Lee, D. Y., and Wie, B., "Autonomous Numerical Predictor-Corrector Guidance for Human Mars Landing Missions," *SSRN*, 2024. https://doi.org/10.2139/ssrn.4811597.

[11] 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.

[12] Sagliano, M., Lu, P., Johnson, B., Seelbinder, D., and Theil, S., "Six-Degrees-of-Freedom Aero-Propulsive Entry Trajectory Optimization," *AIAA SciTech Forum*, 2024, pp. 1–22. https://doi.org/10.2514/6.2024-1171.

[13] Sagliano, M., and Mooij, E., "Optimal drag-energy entry guidance via pseudospectral convex optimization," *Aerospace Science and Technology*, Vol. 117, 2021, p. 106946. https://doi.org/10.1016/j.ast.2021.106946.

[14] Giraldo-Grueso, F., Popov, A. A., and Zanetti, R., "Adaptive Mars Entry Guidance with Atmospheric Density Estimation," *Proceedings of the AAS/AIAA Astrodynamics Specialist Conference*, Broomfield, Colorado, 2024, pp. 1–16.

[15] Roelke, E., McMahon, J. W., Braun, R. D., and Hattis, P. D., "Atmospheric Density Estimation Techniques for Aerocapture," *Journal of Spacecraft and Rockets*, Vol. 60, 2023, pp. 942–956. https://doi.org/10.2514/1.A35197.

[16] Mehra, R., "Approaches to adaptive filtering," *IEEE Transactions on Automatic Control*, Vol. 17, No. 5, 1972, pp. 693–698. https://doi.org/10.1109/TAC.1972.1100100.

[17] Myers, K., and Tapley, B., "Adaptive sequential estimation with unknown noise statistics," *IEEE Transactions on Automatic Control*, Vol. 21, No. 4, 1976, pp. 520–523. https://doi.org/10.1109/TAC.1976.1101260.

[18] Tracy, K., Falcone, G., and Manchester, Z., "Robust Entry Guidance with Atmospheric Adaptation," *AIAA SciTech Forum*, 2023, pp. 1–16. https://doi.org/10.2514/6.2023-0301.

[19] Amato, D., and McMahon, J. W., "Deep Learning Method for Martian Atmosphere Reconstruction," *Journal of Aerospace Information Systems*, Vol. 18, 2021, pp. 728–738. https://doi.org/10.2514/1.I010922.

[20] Rataczak, J. A., Amato, D., and McMahon, J. W., "Density Estimation for Entry Guidance Problems using Deep Learning," *Journal of Guidance, Control, and Dynamics*, Vol. 48, No. 5, 2025, pp. 1042–1053. https://doi.org/10.2514/1.G008498.

[21] Sagliano, M., Mooij, E., and Theil, S., "Adaptive Disturbance-Based High-Order Sliding-Mode Control for Hypersonic-Entry Vehicles," *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 3, 2017, pp. 521–536. https://doi.org/10.2514/1.G000675.

[22] Giraldo-Grueso, F., Popov, A. A., and Zanetti, R., "Precision Mars Entry Navigation with Atmospheric Density Adaptation via Neural Networks," *Journal of Aerospace Information Systems*, Vol. 21, No. 12, 2024. https://doi.org/10.2514/1.I011426.

[23] Zewge, N. S., and Bang, H., "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. https://doi.org/10.3390/rs15041139.

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

[25] Lévesque, J., "Advanced Navigation and Guidance for High-Precision Planetary Landing on Mars," Ph.D. thesis, University of Sherbrooke, 2006.

[26] Christian, J., Verges, A., and Braun, R., "Statistical Reconstruction of Mars Entry, Descent, and Landing Trajectories and Atmospheric Profiles," *AIAA SPACE 2007 Conference & Exposition*, 2007, pp. 1–26. https://doi.org/10.2514/6.2007-6192.

[27] Lévesque, J. F., and de Lafontaine, J., "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. https://doi.org/10.2514/1.25107.

[28] Gazarik, M. J., Wright, M. J., Little, A., Cheatwood, F. M., Herath, J. A., Munk, M. M., Novak, F. J., and Martinez, E. R., "Overview of the MEDLI Project," *2008 IEEE Aerospace Conference*, 2008, pp. 1–12. https://doi.org/10.1109/AERO.2008.4526285.

[29] Hwang, H., Bose, D., Wright, H., White, T. R., Schoenenberger, M., Santos, J., Karlgaard, C. D., Kuhl, C., Oishi, T., and Trombetta, D., "Mars 2020 Entry, Descent and Landing Instrumentation 2 (MEDLI2)," *46th AIAA Thermophysics Conference*, 2016, pp. 1–13. https://doi.org/10.2514/6.2016-3536.

[30] White, T. R., Mahzari, M., Miller, R. A., Tang, C. Y., Karlgaard, C. D., Alpert, H., Wright, H. S., and Kuhl, C., "Mars Entry Instrumentation Flight Data and Mars 2020 Entry Environments," *AIAA SCITECH 2022 Forum*, San Diego, CA, 2022, pp. 1–17. https://doi.org/10.2514/6.2022-0011.

[31] Edquist, K., Mahzari, M., Muppidi, S., and White, T., "Lessons Learned from Aerothermal Flight Data on NASA Mars Entry Vehicles," *2nd International Conference on Flight Vehicles, Aerothermodynamics and Re-entry Missions Engineering*, 2022, pp. 1–8.

[32] Karlgaard, C. D., Stoffel, T. D., White, T., and West, T. K., "Data Fusion of In-Flight Aerothermodynamic Heating Measurements Using Kalman Filtering," *AIAA AVIATION 2022 Forum*, 2022, pp. 1–13. https://doi.org/10.2514/6.2022-3794.

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

[34] Benito, J., and Mease, K. D., "Reachable and Controllable Sets for Planetary Entry and Landing," *Journal of Guidance, Control, and Dynamics*, Vol. 33, No. 3, 2010, pp. 641–654. https://doi.org/10.2514/1.47577.

[35] Engel, D. L., Putnam, Z. R., Woollands, R. M., and Dutta, S., "Optimal Mars Entry Trajectories for Bank-Angle and Alpha-Beta Steering," *2024 IEEE Aerospace Conference*, 2024, pp. 1–16. https://doi.org/10.1109/AERO58975.2024.10521117.

[36] D'Antuono, V., Zavoli, A., Matteis, G. D., Zanetti, R., Pizzurro, S., and Cavallini, E., "Post-Flight Estimation of Aerodynamic Angles for a Launch Vehicle," *AIAA SciTech Forum*, 2024, pp. 1–22. https://doi.org/10.2514/6.2024-2121.

[37] Regan, F. J., and Anandakrishnan, S. M., *Dynamics of Atmospheric Re-Entry*, American Institute of Aeronautics and Astronautics, 1993.

[38] White, P. W., and Hoffman, J., *Earth Global Reference Atmospheric Model (Earth-GRAM): User Guide*, NASA, 2021.

[39] Justh, H., *Mars Global Reference Atmospheric Model 2010 Version: Users Guide*, NASA, 2014.

[40] Zanetti, R., and D'Souza, C., "Recursive Implementations of the Consider Filter," *Journal of the Astronautical Sciences*, Vol. 60, No. 3, 2013, p. 672–685. https://doi.org/10.1007/s40295-015-0068-7.

[41] Zanetti, R., and DeMars, K. J., "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. https://doi.org/10.2514/1.59935.

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

[43] Kingma, D. P., and Ba, J., "Adam: A Method for Stochastic Optimization," *3rd International Conference for Learning Representations*, 2017, pp. 1–15. https://doi.org/10.48550/arXiv.1412.6980.

[44] Stauch, J., and Jah, M., "Unscented Schmidt–Kalman Filter Algorithm," *Journal of Guidance, Control, and Dynamics*, Vol. 38, No. 1, 2015, pp. 117–123. https://doi.org/10.2514/1.G000467.

[45] Liang, Z., and Zhu, S., "Constrained predictor-corrector guidance via bank saturation avoidance for low L/D entry vehicles," *Aerospace Science and Technology*, Vol. 109, 2021, p. 106448. https://doi.org/10.1016/j.ast.2020.106448.

[46] Armijo, L., "Minimization of functions having Lipschitz continuous first partial derivatives," *Pacific Journal of Mathematics*, Vol. 16, No. 1, 1966, pp. 1–3.

[47] Smith, K. M., "Predictive Lateral Logic for Numerical Entry Guidance Algorithms," *AAS/AIAA Space Flight Mechanics Meeting*, 2016, pp. 1–6.

[48] Lee, Y., "Entry Guidance With Final State Targeting Capability Using Numerical Predictor-Corrector Guidance," Ph.D. thesis, Iowa State University, 2024.

[49] Dutta, S., and Braun, R. D., "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. https://doi.org/10.2514/1.A32937.

[50] 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.

[51] Brown, R. G., and Hwang, P. Y. C., *Introduction to Random Signals and Applied Kalman Filtering with Matlab Exercises*, Wiley, 2012.

[52] Mendeck, G., and Craig, L., "Entry Guidance for the 2011 Mars Science Laboratory Mission," *AIAA Atmospheric Flight Mechanics Conference*, 2010, pp. 1–22. https://doi.org/10.2514/6.2011-6639.

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

[54] Bairstow, S., and Barton, G., "Orion Reentry Guidance with Extended Range Capability Using PredGuid," *AIAA Guidance, Navigation and Control Conference and Exhibit*, 2007, pp. 1–17. https://doi.org/10.2514/6.2007-6427.