

LEARNING REACHABILITY FOR HAZARD DETECTION AND AVOIDANCE IN PLANETARY LANDING

Kento Tomita*, Byeong-Un, Jo†, Koki Ho‡

Autonomous hazard detection and avoidance (HD&A) poses a stochastic perception-aware guidance problem, where the visible surface depends on the trajectory, and the safest target locations are kept updated. For the concurrent optimization of the target and trajectory, evaluating the reachable surface under guidance constraints in real-time is critical, but it requires solving optimization problems multiple times. To bypass the optimization-based computation of the reachable surface, we propose to learn the parameterized reachable surface by a neural network, which ultimately enables the reachability-aware guidance algorithms. This paper presents the proposed parameterization method and validation results by numerical simulations.

INTRODUCTION

Autonomous hazard detection and avoidance (HD&A) is becoming a standard practice for space missions with a lander. While the first HD&A maneuver was successfully performed manually by Apollo astronauts in 1969, it took another 44 years before the Chang'e-3 lunar lander demonstrated autonomous HD&A capability in 2013.¹ In 2021, both Mars 2020 and Tianwen-1 demonstrated autonomous HD&A in Mars landing scenarios. Mars 2020 leveraged image-based terrain relative navigation to successfully avoid the pre-identified hazards,^{2,3} while Tianwen-1 used real-time HD&A based on Chang'e-3's experience.^{4,5} These missions demonstrated the indispensability of autonomous HD&A technology for future exploration missions, such as the Dragonfly mission to Titan.⁶ For further details, refer to a recent review study on HD&A for space missions.⁷

One of the unique challenges of HD&A is guidance optimization under safety map uncertainty. Due to physical limitations of the sensor, the spacecraft can only estimate the probability of safety over a certain area. The size, resolution, and prediction accuracy of the safety map depend on the sensor's position, pose, and local topography. The uncertainty of the safety map makes it unrealistic to make a definitive decision on the landing target and associated guidance trajectory in the early phases of the terminal descent.

However, the studies on propellant-optimal powered descent guidance (PDG) algorithms under target uncertainty remain limited. Despite the notable successes of propellant-optimal PDG algorithms over the last decade, particularly in closed-form guidance^{8,9} and computational guidance approaches,¹⁰⁻¹⁴ the majority of studies have assumed a predefined landing target. A few exceptions include the work by Iiyama et al.,¹⁵ in which deep reinforcement learning was applied to the

*PhD Student, School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA 30313, USA.

†Postdoctoral Fellow, Daniel Guggenheim School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA, 30313, USA.

‡Associate Professor, School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA 30313, USA.

simultaneous optimization of target selection and parameterized Zero-Effort Miss and Zero-Effort Velocity (ZEM/ZEV) guidance algorithm using a simple hazard detection model. Recently, Hayner, Buckner, et al.¹⁶ proposed the Adaptive Deferred Decision Trajectory Optimization (Adaptive-DDTO), an adaptive version of DDTO by Elango et al.¹⁷ that optimizes the guidance trajectory for potential divert maneuvers.

The critical components of HD&A are reliable and efficient algorithms for stochastic hazard detection (HD) and for reachability evaluation. For the stochastic HD algorithm, Ivanov et al.¹⁸ developed the probabilistic hazard detection algorithm considering uncertainty in the range measurements of LiDAR sensors. Tomita et al.¹⁹ applied Bayesian deep learning techniques to solve the same problem more efficiently with the improved performance. Recently, Tomita and Ho²⁰ developed the Gaussian random field-based algorithm to handle the topographic uncertainty, which is most critical due to the sparsity of LiDAR measurement from a high altitude. Given a stochastic safety map, efficiently evaluating reachable targets is necessary to optimally guide the spacecraft to the safest landing target. Eren et al.²¹ employed convex optimization to efficiently construct the inner and outer approximation of the reachable set, and Yang and Liu²² proved that the reachable set is indeed convex and compact under certain conditions.

Although convex optimization is efficient, it is impractical to solve them multiple times for real-time decision makings or more higher level guidance optimization whose inner loop involves the reachability evaluation. To further improve the efficiency of the reachability evaluation, we propose to learn the parameterized reachable surface by a neural network. We introduce the powered descent guidance problem with the target visibility constraint and the corresponding reachable surface with the parameterization method. Training data is generated by exploiting the convexity of the approximated reachable set and the convex optimization. We demonstrate the effectiveness of the proposed method with the numerical simulations.

REACHABLE SURFACE

This section introduces the notion of the reachable surface and the proposed parameterization method. Informally, the reachable surface is the set of points on the ground that are reachable by the lander while satisfying the constraints such as the soft landing constraints or the visibility constraints. We first define the dynamics and constraints of the lander, and then define the reachable surface. Finally, we introduce the proposed parameterization method for the reachable surface.

Equations of Motion and Control Constraints

The motion of a planetary lander in terminal descent is described in a topocentric Cartesian coordinate frame by the system of equations

$$\dot{\mathbf{r}} = \mathbf{v} \qquad \mathbf{r}(t_0) = \mathbf{r}_0 \qquad (1a)$$

$$\dot{\mathbf{v}} = \mathbf{T}_{\text{thrust}}/m + \mathbf{g} \qquad \mathbf{v}(t_0) = \mathbf{v}_0 \qquad (1b)$$

$$\dot{m} = -\alpha \|\mathbf{T}_{\text{thrust}}\| \qquad m(t_0) = m_0 \qquad (1c)$$

where $\mathbf{r} \in \mathbb{R}^3$ is the position, $\mathbf{v} \in \mathbb{R}^3$ is the velocity, $m \in \mathbb{R}$ is the mass, $\mathbf{T}_{\text{thrust}} \in \mathbb{R}^3$ is the thrust force, $\mathbf{g} \in \mathbb{R}^3$ is the gravitational acceleration, and $\alpha \in \mathbb{R}$ is a positive mass-flow parameter. The initial conditions are denoted by the subscript 0 and the time of the initial conditions is denoted by

t_0 . The thrust vector is constrained by

$$0 < \rho_1 \leq \|\mathbf{T}_{\text{thrust}}\| \leq \rho_2 \quad (2a)$$

$$\|\mathbf{T}_{\text{thrust}}\| \cos(\theta_p) \leq \mathbf{T}_{\text{thrust}} \cdot \mathbf{e}_z \quad (2b)$$

where ρ_1 and ρ_2 are the minimum and maximum thrust magnitudes, $\mathbf{e}_z = [0, 0, 1]^T$ is the unit vector in the z -direction, and θ_p is the maximum thrust pointing angle.

Visibility Constraints

The unique constraint we introduce for HD&A is the target visibility constraint, which restricts the trajectory to remain within a cone of visibility of the target. In other words, the target must be visible from the lander at all times. The target visibility constraint is defined by

$$\|E(\mathbf{r}(t) - \mathbf{r}(t_f))\| \leq (\mathbf{r}(t) - \mathbf{r}(t_f)) \cdot \mathbf{e}_z \tan(\theta_{\text{FOV}}/2), \quad E = [e_x, e_y] \quad (3)$$

where $\mathbf{r}(t_f)$ is the position at the time of landing, e_x, e_y, e_z are the unit vectors in the $x, y,$ and z directions, respectively, and θ_{FOV} is the field of view (FOV) of the camera. Although the target visibility constraint is mathematically equivalent to the glide slope constraint, the visibility constraint imposes a more stringent constraint on the trajectory as the half of FOV angle (*i.e.*, 5-45 degrees) is typically much smaller than the regular maximum glide slope angle (*i.e.*, 85 degrees). Figure 1 illustrates the mathematical equivalence of the target visibility constraint and the glide slope constraint.

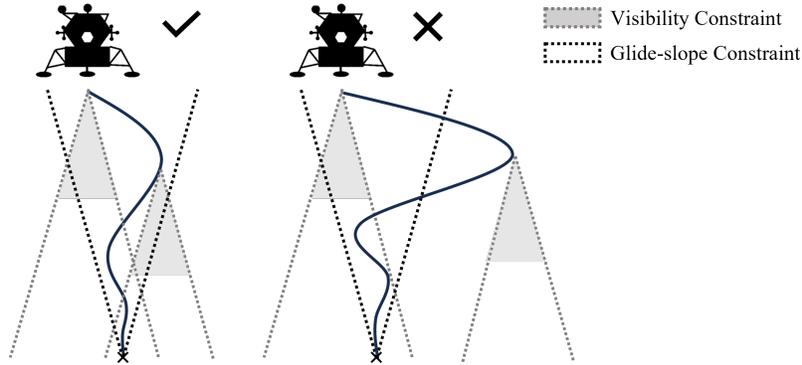


Figure 1: The target visibility constraint. The lander must remain within the cone of visibility of the target, which is mathematically equivalent to the glide slope constraint.

The remaining state constraints are the mass constraint and the boundary conditions for soft landing, which are defined by

$$m_{\text{dry}} \leq m(t) \quad (4a)$$

$$r_z(t_f) = 0, \quad \mathbf{v}(t_f) = \mathbf{0} \quad (4b)$$

where m_{dry} is the dry mass of the lander and we assume $r_z(t_f)$, the altitude of the lander at the time of landing, is zero.

Reachable Surface

The reachable surface is defined as the set of all horizontal positions at t_f that can be reached from the initial state at t_0 while satisfying the equations of motion Eq. (1), control constraints Eq. (2), and state constraints Eqs. (3) and (4).

$$\mathcal{R}_S(t_f) = \left\{ (r_x(t_f), r_y(t_f)) \mid \text{there exists a trajectory satisfying Eqs. (1), (2), (3) and (4) from } (\mathbf{r}(t_0), \mathbf{v}(t_0), m(t_0)) \text{ to } (\mathbf{r}(t_f), \mathbf{v}(t_f), m(t_f)) \right\} \quad (5)$$

We introduce the two-ellipse model to parameterize the boundary of the reachable surface. The two-ellipse model is defined in the two dimensional coordinate frame whose origin is at the initial horizontal position and x -axis is taken to be the downrange direction, i.e. the direction of the initial velocity projected onto the local horizontal plane. The reachable surface is axisymmetric about the downrange direction and approximated by two ellipses, one for the negative-downrange direction and one for the positive-downrange direction, whose origins are on x -axis.

Let a_i, b_i, x_{ci} be the semi-major axis, semi-minor axis, and the x -coordinate of the center of the i th ellipse where $i = 1, 2$, respectively. Suppose ellipse 1 forms the negative-downrange side of the reachable surface and ellipse 2 forms the positive-downrange side, and they intersect at $(x_p, \pm y_p) \in \mathcal{R}_S$ where $y_p > 0$. Let $x_{\min} = x_{c1} - a_1$ and $x_{\max} = x_{c2} + a_2$ be the minimum and maximum x -coordinates of the reachable surface, respectively. Then, the two-ellipse model is defined by the parameters $\mathbf{p} = [x_{\min}, x_{\max}, x_p, y_p, a_1, a_2]$ and is given by

$$\begin{aligned} \hat{\mathcal{R}}_S(\mathbf{p}) &= \hat{\mathcal{R}}_{S1} \cup \hat{\mathcal{R}}_{S2} \\ \hat{\mathcal{R}}_{S1} &= \left\{ (x, y) \mid \frac{(x - x_{c1})^2}{a_1^2} + \frac{y^2}{b_1^2} \leq 1, \quad x \in [x_{\min}, x_p] \right\} \\ \hat{\mathcal{R}}_{S2} &= \left\{ (x, y) \mid \frac{(x - x_{c2})^2}{a_2^2} + \frac{y^2}{b_2^2} \leq 1, \quad x \in [x_p, x_{\max}] \right\} \\ x_{c1} &= x_{\min} + a_1, \quad x_{c2} = x_{\max} - a_2, \\ b_i &= \frac{a_i y_p}{\sqrt{a_i^2 - (x_p - x_{ci})^2}}, \quad i = 1, 2. \end{aligned} \quad (6)$$

Figure 2 illustrates the two-ellipse model.

DATA GENERATION

This section describes the reachable surface data generation algorithms. We first review the important results of lossless convexification of the constrained reachable set first proposed and numerically demonstrated by Eren et al.²¹ and later formally proved by Yang et al.²² Then, we formulate the problems specific to the reachable set computation. Finally, we present reachability-based efficient domain selection of the reachable set data generation.

Constrained Reachable Set via Convex Optimization

Constrained reachable set for our problem is defined as follows.

$$\mathcal{R}(t_f) = \{ \mathbf{x}(t_f) \in \mathbb{R}^7 \mid \text{there exists a trajectory satisfying Eqs. (1)(2)(3)(4)} \} \quad (7)$$

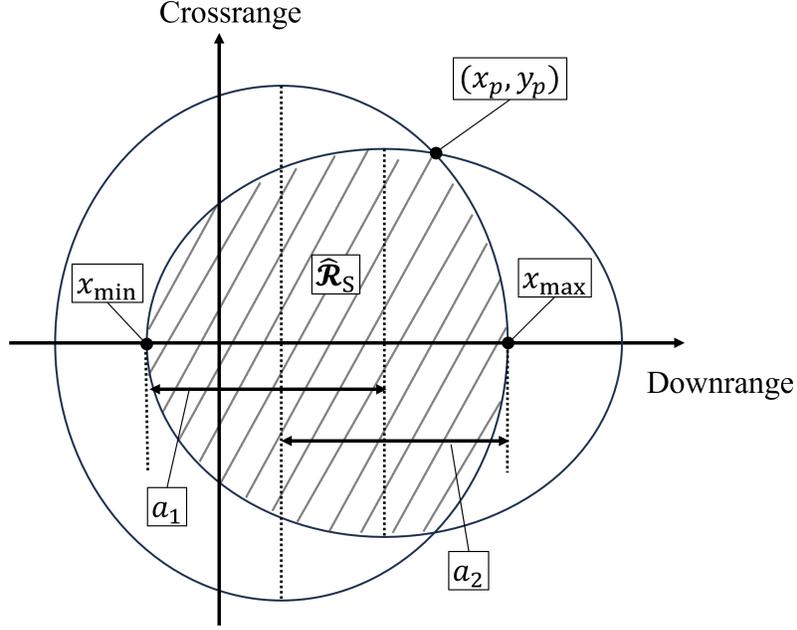


Figure 2: The two-ellipse model. The reachable surface is approximated by two ellipses, one for the negative-downrange half and one for the positive-downrange half.

where $\mathbf{x} = (\mathbf{r}, \mathbf{v}, m)$ is the state of the powered descent dynamics. To apply the lossless convexification to the nonconvex control constraint of Eq. (2), we first convert the nonlinear dynamics to the linear dynamics.¹¹ By introducing the following change of variables,

$$\sigma = \frac{\|\mathbf{T}_{\text{thrust}}\|}{m}, \quad \mathbf{u} = \frac{\mathbf{T}_{\text{thrust}}}{m}, \quad \zeta = \ln m \quad (8)$$

we obtain the linear dynamics with nonconvex control constraint:

$$\dot{\mathbf{r}} = \mathbf{v}, \quad \dot{\mathbf{v}} = \mathbf{u} + \mathbf{g}, \quad \dot{\zeta} = -\alpha\sigma. \quad (9)$$

subject to

$$\|\mathbf{u}\| = \sigma \quad (\text{Nonconvex}) \quad (10a)$$

$$\mathbf{u} \cdot \hat{\mathbf{e}}_z \geq \sigma \cos(\theta_p) \quad (\text{Convex}) \quad (10b)$$

$$\rho_1 e^{-\zeta} \leq \sigma \leq \rho_2 e^{-\zeta} \quad (\text{Nonconvex}). \quad (10c)$$

Acikmese et al.^{11,21} introduced the tighter approximation at the cost of marginal errors to convexify the control constraint of Eq. (10c). Introducing the deterministic lower bound of ζ as $\underline{\zeta} = \ln(m_0 - \rho_2 t)$, the approximated control constraints are given by

$$\|\mathbf{u}\| = \sigma \quad (\text{Nonconvex}) \quad (11a)$$

$$\mathbf{u} \cdot \hat{\mathbf{e}}_z \geq \sigma \cos(\theta_p) \quad (\text{Convex}) \quad (11b)$$

$$\rho_1 e^{-\zeta} \left(1 - (\zeta - \underline{\zeta}) + \frac{(\zeta - \underline{\zeta})^2}{2} \right) \leq \sigma \leq \rho_2 e^{-\zeta} (1 - (\zeta - \underline{\zeta})) \quad (\text{Convex}). \quad (11c)$$

The fully convex controll constrains are obtained by replacing the remaining nonconvex constraint of Eq. (11a) with the convex constraint

$$\|\mathbf{u}\| \leq \sigma. \quad (12)$$

Then, the main results of lossless convexification of reachable set (Theorem 10 in (Reference 22)) are summarized as follows.

Assumption 1 *The state constraint of Eq. (3) is active at most over a discrete set; i.e., $\{t \in (0, t_f) \mid \text{Eq. (3) is active}\}$ is empty or a discrete set.*

Theorem 1 *Let $\mathcal{R}_\ell(t_f)$ be the reachable set of the linear dynamics with nonconvex control constraints and $\mathcal{R}_c(t_f)$ be the reachable set of the linear dynamics with convex control constraints:*

$$\mathcal{R}_\ell(t_f) = \{\mathbf{x}(t_f) \in \mathbb{R}^7 \mid \text{Eqs. (9)(11)(3)(4)}\} \quad (13)$$

$$\mathcal{R}_c(t_f) = \{\mathbf{x}(t_f) \in \mathbb{R}^7 \mid \text{Eqs. (9)(12)(11b)(11c)(3)(4)}\} \quad (14)$$

If Assumption 1 holds, we have $\mathcal{R}_\ell(t_f) = \mathcal{R}_c(t_f)$, and they are convex and compact.

In summary, under the control input such that Assumption 1 holds; i.e., the visibility constraint is active at most over a discrete set, the original reachable set can be well approximated by a convex and compact reachable set. It is also an inner approximation because of the tighter approximation of the control constraint of Eq. (10c) by Eq. (11c).

$$\mathcal{R}(t_f) \approx \mathcal{R}_\ell(t_f) = \mathcal{R}_c(t_f), \quad \mathcal{R}(t_f) \subseteq \mathcal{R}_\ell(t_f) = \mathcal{R}_c(t_f). \quad (15)$$

Maximum Range Problem

The reachable surface $\mathcal{R}_S(t_f)$ is a subspace of the reachable set $\mathcal{R}(t_f)$ (i.e., $\mathcal{R}_S(t_f)$ is the set of x, y coordinates of $\mathcal{R}(t_f)$), and we define $\mathcal{R}_{S,\ell}(t_f)$ and $\mathcal{R}_{S,c}(t_f)$ as the reachable surface of $\mathcal{R}_\ell(t_f)$ and $\mathcal{R}_c(t_f)$, respectively:

$$\mathcal{R}_S(t_f) = \{(r_x(t_f), r_y(t_f)) \mid (\mathbf{r}(t_f), \mathbf{v}(t_f), m(t_f)) \in \mathcal{R}(t_f)\} \quad (16a)$$

$$\mathcal{R}_{S,\ell}(t_f) = \{(r_x(t_f), r_y(t_f)) \mid (\mathbf{r}(t_f), \mathbf{v}(t_f), m(t_f)) \in \mathcal{R}_\ell(t_f)\} \quad (16b)$$

$$\mathcal{R}_{S,c}(t_f) = \{(r_x(t_f), r_y(t_f)) \mid (\mathbf{r}(t_f), \mathbf{v}(t_f), m(t_f)) \in \mathcal{R}_c(t_f)\}. \quad (16c)$$

and

$$\mathcal{R}_S(t_f) \approx \mathcal{R}_{S,\ell}(t_f) = \mathcal{R}_{S,c}(t_f), \quad \mathcal{R}_S(t_f) \subseteq \mathcal{R}_{S,\ell}(t_f) = \mathcal{R}_{S,c}(t_f). \quad (17)$$

Note that the reachable surface $\mathcal{R}_{S,c}(t_f)$ is a convex and compact set, and we can obtain its boundary points by solving the *maximum range problem*, which is a convex optimization and defined as follows.

Problem 1 (maximum range problem) *Given an interior point $\mathbf{r}_{int} = (x_{int}, y_{int}, 0)$ where $(x_{int}, y_{int}) \in \mathcal{R}_{S,c} \setminus \partial\mathcal{R}_{S,c}$ and a direction $\mathbf{d} = (d_x, d_y, 0) \in \mathbb{R}^3$, we solve the maximization problem*

$$\max_{\mathbf{u}(\cdot), \sigma(\cdot)} \mathbf{d} \cdot (\mathbf{r}(t_f) - \mathbf{r}_{int}) \quad (18)$$

subject to Eqs. (9), (12), (11b), (11c), (3), and (4) and the following boundary conditions

$$\mathbf{d} \times (\mathbf{r}(t_f) - \mathbf{r}_{int}) = 0 \quad (19)$$

Equation (19) enforces the landing position to be on the line passing through \mathbf{r}_{int} and parallel to \mathbf{d} . The maximum range problem is solved for a number of directions \mathbf{d} to obtain a set of discrete points of the reachable surface boundary. The axisymmetry of the reachable surface is exploited to reduce the number of directions \mathbf{d} to be considered and also to find the interior point \mathbf{r}_{int} . The interior point is obtained by solving Problem 1 with the specific parameters as follows.

Problem 2 (interior point search) *Given the initial state $\mathbf{x}(t_0) = (\mathbf{r}(t_0), \mathbf{v}(t_0), m(t_0))$, consider the coordinate frame whose origin is $(r_x(t_0), r_y(t_0), 0)$, x -axis is parallel to the horizontal velocity (i.e., downrange direction), and y -axis is the crossrange direction. Obtain the interior point*

$$\mathbf{r}_{\text{int}} = \left(\frac{1}{2}(x_{\min} + x_{\max}), 0, 0 \right) \quad (20)$$

where x_{\min} and x_{\max} are the solutions of Problem 1 where $(\mathbf{d}, \mathbf{r}_{\text{int}}) = (-\mathbf{e}_x, \mathbf{0})$ and $(\mathbf{e}_x, \mathbf{0})$, respectively.

Initial Condition Search

The initial conditions, $(\mathbf{x}(t_0), t_{go})$, exists in 8 dimensional space, which makes it difficult to generate sufficient initial conditions and solving for the corresponding reachable sets. Exploiting the symmetricity of the reachable set, we can reduce the dimensionality of the initial condition space to 5; altitude, downrange velocity, horizontal velocity, mass, and time-to-go, which are denoted as $(r_z(t_0), v_x(t_0), v_z(t_0), m(t_0), t_{go})$. Although the initial condition space is still too large to explore extensively, we can further downselect them by introducing the notion of the feasible initial conditions, which are defined as follows:

Definition 1 (feasible initial condition) *An initial condition $(\mathbf{x}(t_0), t_{go})$ is feasible if there exists a control input $(\mathbf{u}(\cdot), \sigma(\cdot))$ such that the trajectory $\mathbf{x}(\cdot)$ satisfies the dynamics (9) and the constraints (12), (11b), (11c), (3), and (4).*

The feasible initial conditions are essentially the constrained reachable set at $t = t_0 \in [0, t_f]$, whose boundary is constrained such that $\mathbf{x}(t = 0) \in \chi_0$ and the terminal condition of Eq. (19) with $t_f = t_0 + t_{go}$, where χ_0 is an appropriately chosen convex and compact set of states at $t = 0$. Therefore, we can obtain the set of feasible initial conditions by solving the convex optimization problems.²¹

Once the set of feasible initial condition is obtained, we can generate the initial conditions by sampling from an outer approximation of the feasible initial condition set. Note that it is important to sample from the outer approximation because it also generates infeasible initial conditions, which are necessary to train the neural network to distinguish between feasible and infeasible initial conditions.

NEURAL NETWORK ARCHITECTURE

We use a multilayer perceptron (MLP) model whose architecture is shown in Figure 3. The input layer has 5 neurons, one for each of the initial condition elements; altitude, downrange velocity, horizontal velocity, mass, and time-to-go, which are denoted as $\mathbf{x}_{\text{IC}} = (r_z(t_0), v_x(t_0), v_z(t_0), m(t_0), t_{go})$. The output layer has 7 neurons, one for the feasibility evaluation and the others for the parameter

vector $\mathbf{p} = [x_{\min}, x_{\max}, x_p, y_p, a_1, a_2]$ of the two-ellipse model of the reachable surface. The activation function is the Rectified Linear Unit (ReLU) for the hidden layer, and the sigmoid function for the output layer.

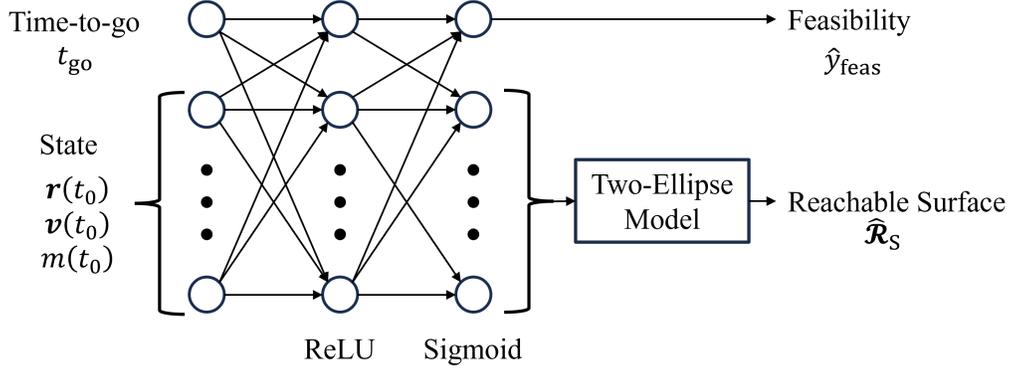


Figure 3: Neural network architecture.

Our loss function is defined as the sum of the mean squared error (MSE) of the predicted feasibility and the reachable surface. The loss function is given by

$$\mathcal{L} = \mathcal{L}_{\text{feas}} + \mathcal{L}_{\text{reach}}, \quad (21)$$

where $\mathcal{L}_{\text{feas}}$ is the MSE of the feasibility evaluation, and $\mathcal{L}_{\text{reach}}$ is the MSE of the reachable surface boundary errors. Let $y_{\text{feas}} \in \{0, 1\}$ and $\hat{y}_{\text{feas}} = f_{\text{feas}}(\mathbf{x}_{\text{IC}}) \in [0, 1]$ be the ground truth and the predicted value of the feasibility evaluation, respectively. Note that y_{feas} is a binary variable but \hat{y}_{feas} is a continuous variable from 0 to 1. The MSE of the feasibility evaluation is given by

$$\mathcal{L}_{\text{feas}} = \frac{1}{N} \sum_{i=1}^N \sqrt{(y_{\text{feas}}^{(i)} - \hat{y}_{\text{feas}}^{(i)})^2}, \quad (22)$$

where N is the batch size of training data.

To compute $\mathcal{L}_{\text{reach}}$, we first compute the reachable surface boundary from the predicted parameter vector $\hat{\mathbf{p}} = [\hat{x}_{\min}, \hat{x}_{\max}, \hat{x}_p, \hat{y}_p, \hat{a}_1, \hat{a}_2]$ with Eq. (6). Let $\{(x_j^{(i)}, y_j^{(i)})\}_j$ be the ground truth boundary points of the reachable surface, obtained by solving Problem 1 with the initial condition $\mathbf{x}_{\text{IC}}^{(i)}$ and a set of direction vectors $\{\mathbf{d}_j\}_j$ where $j = 1, \dots, M$. Then, we compute the predicted boundary points $\{\hat{x}_j^{(i)}, \hat{y}_j^{(i)}\}_j$ with Eq. (6) where $\hat{x}_j^{(i)}$ is computed as follows:

$$\hat{x}_j^{(i)} = \hat{x}_{\min}^{(i)} + \frac{\hat{x}_{\max}^{(i)} - \hat{x}_{\min}^{(i)}}{x_{\max}^{(i)} - x_{\min}^{(i)}} (x_j^{(i)} - x_{\min}^{(i)}). \quad (23)$$

Then, the MSE of the reachable surface boundary error is given by

$$\mathcal{L}_{\text{reach}} = \frac{1}{N} \sum_{i=1}^N \frac{1}{M} \sum_{j=1}^M \sqrt{(\hat{x}_j^{(i)} - x_j^{(i)})^2 + (\hat{y}_j^{(i)} - y_j^{(i)})^2}. \quad (24)$$

EXPERIMENTS AND RESULTS

Reachable Set Data Analysis

We generated 996,509 pairs of initial conditions, each of which are used as the input of Problem 1 for 15 times to obtain the corresponding set of boundary points of reachable surface. Note that we only store the results satisfying Assumption 1. The number of feasible initial conditions are 321,924, which is 32.3% of the total data. Each initial condition is generated by sampling the initial altitude $r_z(t_0)$, the initial horizontal velocity $v_x(t_0)$, the initial vertical velocity $v_z(t_0)$, the initial mass $m(t_0)$, and the time-to-go t_{go} from the uniform distribution over the outer approximation of the reachable set of interest. The outer approximation of the reachable set, is obtained by solving the convex optimization problems with an appropriate initial state set,²¹ where we removed the visibility constraint. Figure 4 shows the distribution of 10,000 random samples of generated data with their feasibility. We can see the feasible points have a clear upper bound of the initial horizontal velocity over altitudes, which is due to the visibility constraint, which is not imposed at the initial condition search.

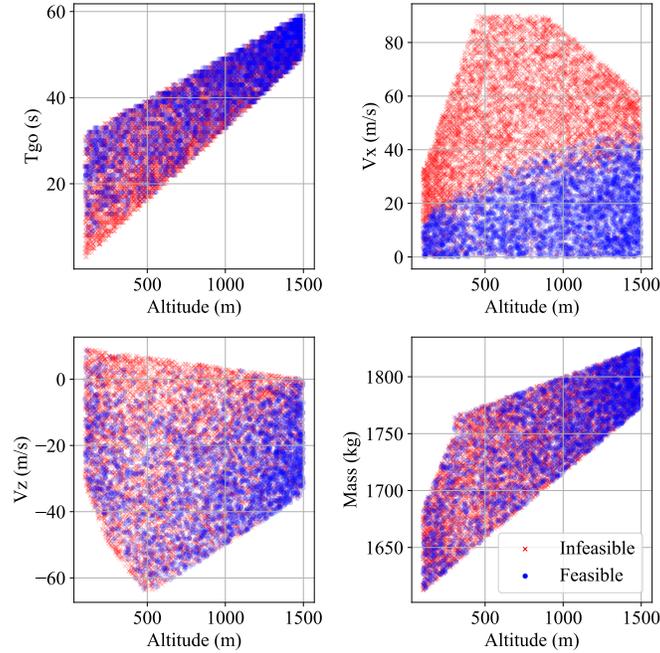


Figure 4: Distribution of 10,000 random samples of generated data points with their feasibility.

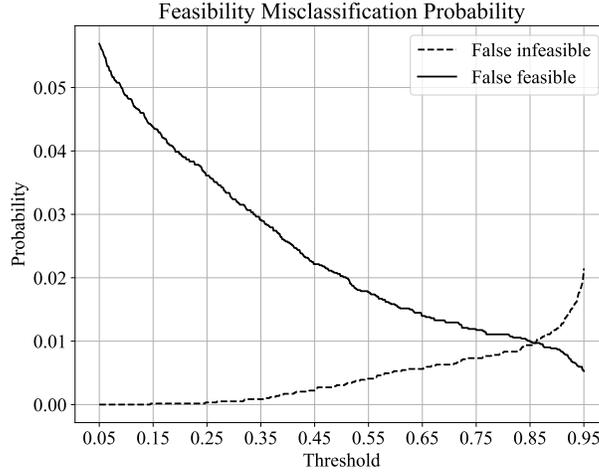
Quantitative Analysis of the Reachable Surface Prediction

Table 1 shows the errors of the feasibility prediction and the reachable surface prediction. The feasibility prediction output from the neural network model is continuous from 0 to 1, and we set the threshold to 0.5, which results in the error of 1.54%. The reachable surface prediction error is measured as the mean squared error of Euclidian distances between the predicted boundary points and the ground truth data points, which results in the error of 6.77 meters on average with standard deviation of 4.55 meters. The accuracy of the feasibility prediction depends on the threshold value.

Table 1: Neural network prediction error.

Feasibility (%)	Reachable Surface (m)
1.54%	6.77 ± 4.55

Figure 5 shows the rate of false feasible (*i.e.*, infeasible but predicted feasible) and false infeasible (*i.e.*, feasible but predicted infeasible) predictions for different threshold values. The fraction of false feasible predictions decreases as the threshold value increases, while the fraction of false infeasible predictions increases. At threshold of about 0.85, the fraction of false feasible and false infeasible predictions intersect each other at 1.0%. The distribution of the reachable surface prediction error

**Figure 5:** Fraction of false feasible and false infeasible predictions for different threshold values.

is shown in Figure 6. Even though the prediction error is small on average, the error becomes large for some initial conditions. For further investigation, we analyzed the prediction error for different initial conditions, specifically for the different initial altitudes $r_z(t_0)$ and initial horizontal velocities $v_x(t_0)$. Figure 7a shows the reachable surface prediction error for different initial altitudes. The prediction error is larger for the higher initial altitudes, which aligns with the fact that the reachable set is larger for the higher initial altitudes. Figure 7b shows the reachable surface prediction error for different initial horizontal velocities. The effective prediction error range is largest for the middle range of the initial horizontal velocities, which is from 20 to 30 meters per second.

Qualitative Analysis of the Reachable Surface Prediction

Figure 8 shows the reachable surface prediction for different initial conditions. Due to the visibility constraint, the reachable surface is bounded by the circle with radius of $r_z(t_0) \tan(\theta_{\text{FOV}}/2)$ whose origin is the initial horizontal position. Figure 8 demonstrates that our neural network model successfully predict the reachable surface with the proposed two-ellipse model. Figure 9 shows the minimum-fuel trajectory with predicted reachable surface. It demonstrates the predicted reachable surface changes its shape with the horizontal velocity and its size with the altitude.

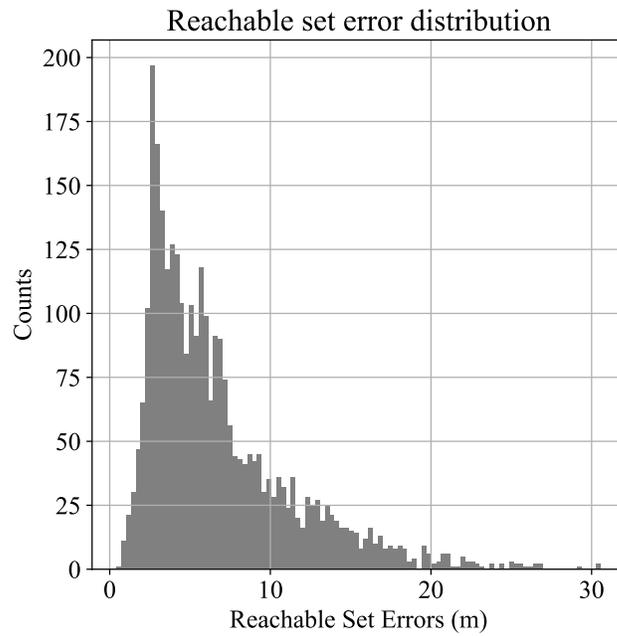
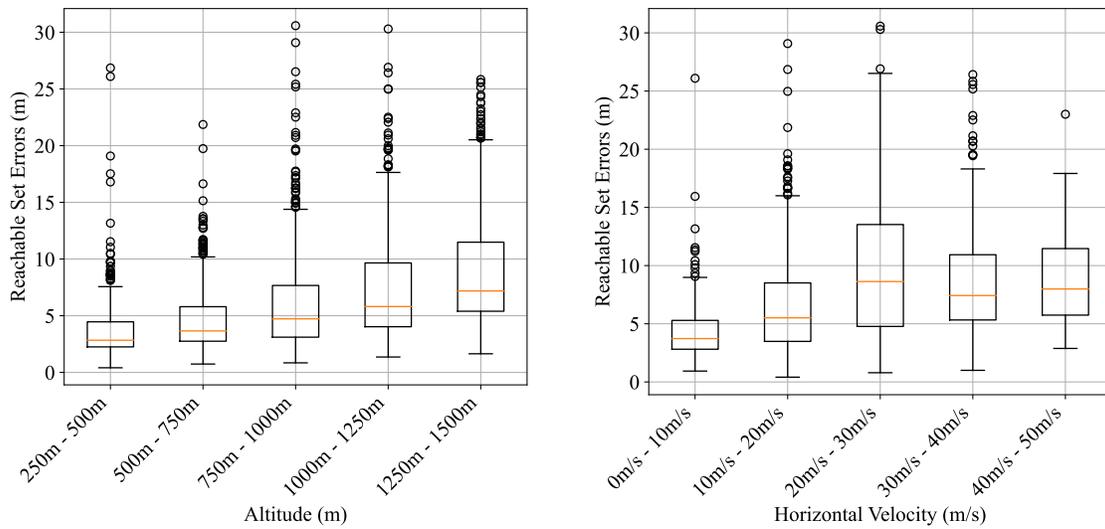


Figure 6: Distribution of the reachable surface prediction error.



(a) Reachable surface prediction error for different initial altitudes.

(b) Reachable surface prediction error for different initial horizontal velocities.

Figure 7: Analysis of reachable surface prediction error for various initial conditions.

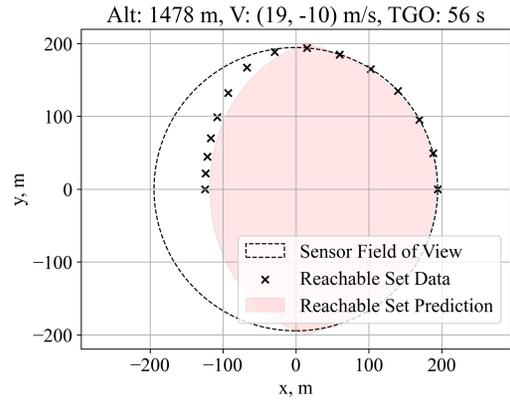
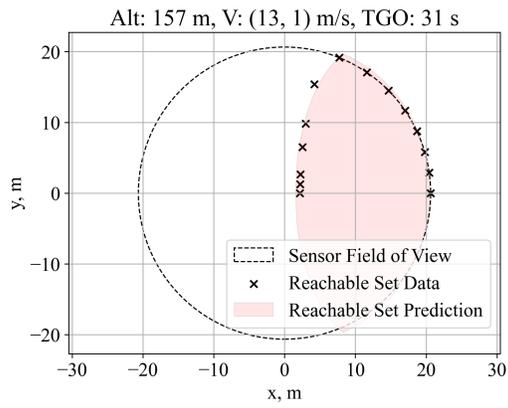


Figure 8: Reachable surface predictions for different initial conditions.

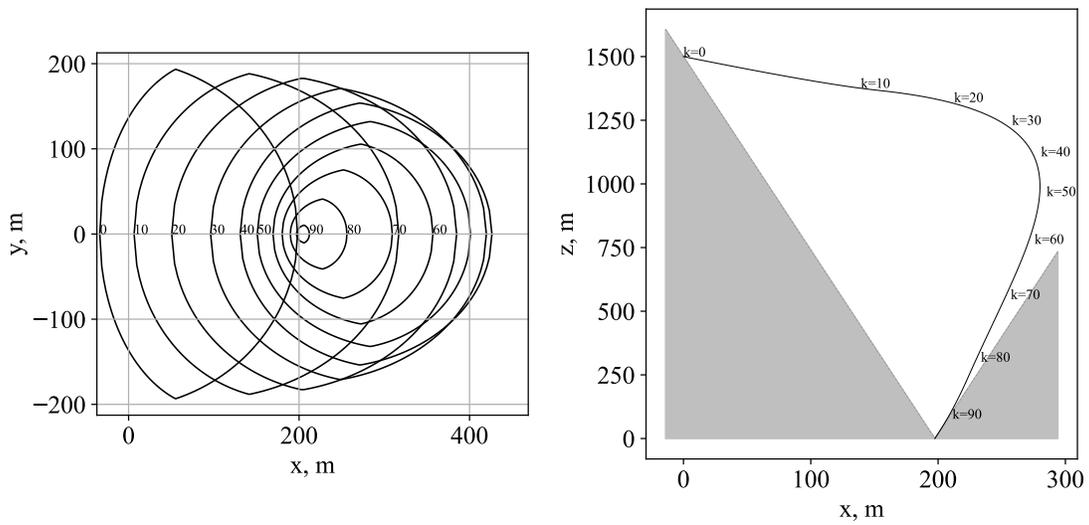


Figure 9: Minimum-fuel trajectory with a set of predicted reachable surfaces.

CONCLUSION

We proposed to learn the parameterized reachable surface by a neural network for hazard detection and avoidance in planetary landing. We introduced the visibility constraint and the two-ellipse model for the reachable surface parameterization. Lossless convexification techniques were utilized to efficiently generate the training data. We also presented the neural network training architecture where the true reachable surface parameters are not required, which enables the optimal prediction from the data generated by a number of convex optimization problems. The trained neural network was validated by numerical simulations and we demonstrated that the proposed approach enables efficient and accurate evaluation of the reachable surface.

ACKNOWLEDGMENT

This work is supported by the National Aeronautics and Space Administration under Grant No.80NSSC20K0064 through the NASA Early Career Faculty Program.

REFERENCES

- [1] X. Jiang, S. Li, and T. Tao, "Innovative Hazard Detection and Avoidance Strategy for Autonomous Safe Planetary Landing," *Acta Astronautica*, Vol. 126, Sept. 2016, pp. 66–76, 10.1016/j.actaastro.2016.02.028.
- [2] 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*, Big Sky, MT, USA, IEEE, Mar. 2019, pp. 1–20, 10.1109/AERO.2019.8742167.
- [3] A. E. Johnson, Y. Cheng, N. Trawny, J. F. Montgomery, S. Schroeder, J. Chang, D. Clouse, S. Aaron, and S. Mohan, "Implementation of a Map Relative Localization System for Planetary Landing," *Journal of Guidance, Control, and Dynamics*, Vol. 46, Apr. 2023, pp. 618–637, 10.2514/1.G006780.
- [4] X. Huang, M. Li, X. Wang, J. Hu, Y. Zhao, M. Guo, C. Xu, W. Liu, Y. Wang, C. Hao, and L. Xu, "The Tianwen-1 Guidance, Navigation, and Control for Mars Entry, Descent, and Landing," *Space: Science & Technology*, Vol. 2021, Jan. 2021, p. 2021/9846185, 10.34133/2021/9846185.
- [5] X. Huang, C. Xu, J. Hu, M. Li, M. Guo, X. Wang, Y. Zhao, B. Hua, and Y. Wang, "Powered-Descent Landing GNC System Design and Flight Results for Tianwen-1 Mission," *Astrodynamic*, Vol. 6, Mar. 2022, pp. 3–16, 10.1007/s42064-021-0118-9.
- [6] R. D. Lorenz, "Dragonfly : Entry and Descent One Titan Year after Huygens," *AIAA SCITECH 2023 Forum*, National Harbor, MD & Online, American Institute of Aeronautics and Astronautics, Jan. 2023, 10.2514/6.2023-0599.
- [7] R. D. Lorenz, "Planetary Landings with Terrain Sensing and Hazard Avoidance: A Review — Elsevier Enhanced Reader," Nov. 2022, 10.1016/j.asr.2022.11.024.
- [8] Y. Guo, M. Hawkins, and B. Wie, "Applications of generalized zero-effort-miss/zero-effort-velocity feedback guidance algorithm," *Journal of Guidance, Control, and Dynamics*, Vol. 36, No. 3, 2013, pp. 810–820.
- [9] S. Han and K. Ho, "Terminal landing guidance law using analytic gravity turn trajectory," *The 33rd AAS/AIAA Space Flight Mechanics Meeting*, Austin, Texas, USA, 2023.
- [10] P. Lu, "Introducing computational guidance and control," *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 2, 2017, pp. 193–193.
- [11] B. Acikmese and S. R. Ploen, "Convex programming approach to powered descent guidance for mars landing," *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 5, 2007, pp. 1353–1366, 10.2514/1.27553.
- [12] L. Blackmore, B. Açikmeşe, and D. P. Scharf, "Minimum-landing-error powered-descent guidance for Mars landing using convex optimization," *Journal of guidance, control, and dynamics*, Vol. 33, No. 4, 2010, pp. 1161–1171.
- [13] L. Blackmore, "Autonomous precision landing of space rockets," *Frontiers of Engineering: Reports on Leading-Edge Engineering from the 2016 Symposium*, Vol. 46, The Bridge Washington, DC, 2016, pp. 15–20.

- [14] P. Lu and R. Callan, "Propellant-Optimal Powered Descent Guidance Revisited," *Journal of Guidance, Control, and Dynamics*, Vol. 46, No. 2, 2023, pp. 215–230.
- [15] K. Iiyama, K. Tomita, B. A. Jagatia, T. Nakagawa, and K. Ho, "Deep Reinforcement Learning for Safe Landing Site Selection with Concurrent Consideration of Divert Maneuvers," Feb. 2021.
- [16] C. R. Hayner, S. C. Buckner, D. Broyles, E. Madewell, K. Leung, and B. Acikmese, "HALO: Hazard-Aware Landing Optimization for Autonomous Systems," Apr. 2023.
- [17] P. Elango, S. B. Sarsılmaz, and B. Açıkmeşe, "Deferring Decision in Multi-target Trajectory Optimization," *SciTech 2022*, 2022.
- [18] T. Ivanov, A. Huertas, and J. M. Carson, "Probabilistic hazard detection for autonomous safe landing," *AIAA Guidance, Navigation, and Control (GNC) Conference*, 2013, p. 5019.
- [19] K. Tomita, K. A. Skinner, and K. Ho, "Bayesian deep learning for segmentation for autonomous safe planetary landing," *Journal of Spacecraft and Rockets*, Vol. 59, No. 6, 2022, pp. 1800–1808.
- [20] K. Tomita and K. Ho, "Stochastic Hazard Detection For Landing Under Topographic Uncertainty," *arXiv preprint arXiv:2305.04249*, 2023.
- [21] U. Eren, D. Dueri, and B. Açıkmeşe, "Constrained reachability and controllability sets for planetary precision landing via convex optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 38, No. 11, 2015, pp. 2067–2083.
- [22] R. Yang and X. Liu, "Reachable Set Computation of Linear Systems with Nonconvex Constraints via Convex Optimization," Vol. 146, p. 110632, 10.1016/j.automatica.2022.110632.