

Mixed-Input Learning for Multi-point Landing Guidance with Hazard Avoidance Part II: Learning-based Guidance Algorithm

Sixiong You *

Purdue University, West Lafayette, IN 47907

Chaoying Pei[†]

Purdue University, West Lafayette, IN 47907

Vinay Kenny[‡]

Purdue University, West Lafayette, IN 47907

Ran Dai[§]

Purdue University, West Lafayette, IN 47907

Jeremy R. Rea[¶]

NASA Johnson Space Center, Houston, TX, 77058

This paper investigates the three-dimensional (3D) multi-point landing guidance (MLG) problem with hazard avoidance by developing a mixed-input learning-based method to achieve precise and fuel-efficient planetary landing in future Mars missions. Specifically, we aim to find a safe, fuel-efficient landing point and generate a fuel-optimal trajectory simultaneously in real-time. First, by introducing binary variables, the MLG problem is formulated as an optimal control problem with quadratic constraints. Then, by formulating the Hamiltonian function, the necessary conditions of optimality for the MLG problem are obtained, where the critical parameters are identified to represent the complete optimal solution. After that, to find the implicit relationship between the problem inputs and these critical parameters, a hybrid deep neural network is constructed. To be specific, on the one hand, the contour maps of the landing area, which are image inputs, are adopted to reflect the features of the pre-defined landing area. On the other hand, the velocity and position vectors, which belong to numeric inputs, are adopted to reflect the features of the initial state of the powered descent phase. Finally, with the constructed hybrid deep neural network well trained, the mixed-input learning-based optimal control solution can be computed onboard. To verify the effectiveness and accuracy of the proposed method, the simulation results of 3D MLG problems are presented and analyzed.

I. Nomenclature

$\mathbf{g}$	=	gravitational acceleration vector, m/s^2
g_0	=	gravitational acceleration at Mars surface, m/s^2
H	=	Hamiltonian
J	=	objective function
m	=	lander mass, kg
r	=	radial distance from Mars center to the landing vehicle, m
$\mathbf{r}$	=	position vector, m
R_0	=	Mars radius, m
t	=	time, s

*Graduate Research Associate, School of Aeronautics and Astronautics, 701 W. Stadium Ave, West Lafayette, IN

[†]Graduate Research Associate, School of Aeronautics and Astronautics, 701 W. Stadium Ave, West Lafayette, IN

[‡]Graduate Research Associate, School of Aeronautics and Astronautics, 701 W. Stadium Ave, West Lafayette, IN

[§]Associated Professor, School of Aeronautics and Astronautics, 701 W. Stadium Ave, West Lafayette, IN; AIAA Member.

[¶]Subsystem Manager, Flight Mechanics and Trajectory Design Branch, 2001 NASA Parkway - EG, 577058; AIAA Member.

- $\mathbf{T}_c$ = thrust vector, N
- $\mathbf{V}$ = lander velocity vector, m/s
- α = combustion efficiency of rocket, kg/[N·s]
- λ = adjoint variables
- ν = Lagrangian multiplier

II. Introduction

For the multi-point landing guidance (MLG) problem with hazard avoidance, the main purpose is to achieve safe, soft and fuel-optimal landing with multiple pre-selected landing points [1–3]. Generally, the multi-point landing problem includes two main parts: the target landing point selection and the optimal guidance commands generation [4, 5].

For the target landing point selection, several methods have been developed. In [1], the convex programming based algorithm is developed to solve the MLG problem, where the landing point is selected via interpolating a look-up table of the estimated propellant mass. In [6], via developing a new criterion, which estimates terrain safety, fuel consumption, and touchdown performance, the best landing point can be determined as the solution of the reformulated problem. In [7] and [4], the optimal sensitivity is applied to select the fuel-optimal landing site from a list of candidate landing sites in the target landing region. Besides, in [8], on the basis of the ballistic analysis, the target landing point can be selected in real-time.

The fuel-optimal landing guidance has also been extensively studied. As a special type of optimal control problem, both direct and indirect methods have been applied. For the direct methods, the idea is to convert the time-continuous optimal control problem into a parameter optimization problem via well-defined collocation methods. Then, the nonlinear programming (NLP) solver [9–12] can be applied to solve the parameter optimization problem. However, for most direct methods, the global convergence to a local optimum can not be guaranteed and a good initial guess of the unknown variables is generally required. In addition, due to the nonconvex nature of the vehicle dynamics and operational constraints, many relaxation methods [13, 14] and lossless convexification [15, 16] have been developed. However, all these methods face a trade-off between performance and computational time. On the other hand, based on Hamiltonian and first-order necessary conditions, indirect methods can convert the optimal guidance problem into a two-point-boundary-value problem (TPBVP). To solve the obtained TPBVPs, approximation [17–19] and shooting methods [20, 21] have been developed. However, existing methods cannot guarantee convergence to the exact solution due to the sensitivity of adjoint variables. Besides, for optimal control problems with varying/uncertain parameters in the dynamical model or states/controls, supervised learning and reinforcement learning has also been applied to find optimal control solutions in real-time [22–25]. In summary, due to the huge computational burden of the entire multi-point landing mission, the existing methods for solving the MLG problems tend to divide the whole mission into two separate steps. In this paper, we aim to solve the landing point selection and the optimal guidance generation simultaneously, which can generate the fuel-optimal trajectory from the start of the powered descent phase to a selected landing point in one integrated approach.

Our previous work in [25] proposed a learning-based optimal control method to solve the pinpoint fuel-optimal powered descent guidance problem in real-time. In this paper, we extend the learning-based and theory-supported optimal control method to solve the MLG problem. Different from pinpoint powered descent guidance, one of the challenges when solving the MLG problem is the additional requirement for selecting a target point from a pre-selected target set. This decision needs to be made onboard after obtaining the landing area image, which involves heavy computational loads for image processing and decision making. Therefore, to select a safe target point and generate an optimal path guiding to the selected point in real-time, the mixed-input learning-based method (MILM) is applied to solve the MLG problem, where both landing area images and numeric position/velocity vectors at the initial phase of the landing mission are taken as mixed-inputs. To be specific, guided by the optimal control theory, the optimal solution to the MLG problem is represented via a few critical parameters. Then, opposite to mapping the optimal controls and the states directly, the proposed approach only maps the relationship between the identified critical parameters and the given initial conditions. Besides, for the MLG problem, both the features of the desired target region and the initial states are taken as initial conditions. To efficiently process the mixed-inputs data, a hybrid deep neural network is adopted. The hybrid deep neural network has been widely applied to deal with mixed-inputs data, including but not limited to language detection, medical field, signal processing and so on [26–28]. Finally, with the well-trained hybrid deep neural network, the optimal trajectory to the selected landing point can be reconstructed in real-time via the identified critical parameters.

This paper is organized as follows. In Section III, the formulation of the MLG problem with hazard avoidance

is introduced. In Section IV, the MILM is described. In Section V, the simulation results of the MLG problem are provided and analyzed. Conclusions are addressed in Section VI.

III. Problem Formulation

For the MLG problem with hazard avoidance, the landing vehicle is required to select a landing point from a target region containing hazards and then guide the vehicle to the selected landing point with high precision. Motion during the MLG is analyzed in a Cartesian coordinate system with the origin fixed to the surface of Mars. Without loss of generality, we assume the landing vehicle can be treated as a point mass and the origin of the Cartesian coordinate is located at the center of the target region. The 3D equations of motion of the landing vehicle are expressed as

$$\dot{\mathbf{r}} = \mathbf{V} \quad (1a)$$

$$\dot{\mathbf{V}} = \mathbf{g} + \frac{\mathbf{T}_c}{m(t)} \quad (1b)$$

$$\dot{m} = -\alpha \|\mathbf{T}_c(t)\| \quad (1c)$$

where $\mathbf{r} = [x, y, z]^T$ represents the position vector, and $\mathbf{V} = [v_x, v_y, v_z]^T$ represents the velocity vector. $\mathbf{g} = [0, 0, g]$ is the constant acceleration vector due to gravity. In addition, α is a constant value, which denotes the combustion efficiency of a rocket. $\mathbf{T}_c = [\|\mathbf{T}_c(t)\| \sin \theta \cos \phi, \|\mathbf{T}_c(t)\| \sin \theta \sin \phi, \|\mathbf{T}_c(t)\| \cos \theta]$ is the thrust vector of the landing vehicle, where $\|\mathbf{T}_c(t)\|$ is the thrust magnitude, θ is the angle between the thrust vector and z-axis, and ϕ is the angle between the thrust projection on x-y plane and x-axis. Besides the nonlinear dynamics, there exist several control and state constraints, including the upper and lower bounded thrust magnitude, i.e., $T_{\min} \leq \|\mathbf{T}_c(t)\| \leq T_{\max}$, and the glide-slope constraint on the position vector. The glide-slope constraint ensures that the vehicle is kept within a safe distance from the ground until it reaches the landing point. The glide-slope constraint can be described as a convex cone, expressed as

$$C = \{\mathbf{r} \in \mathbb{R}^2 : \frac{(\mathbf{r} - \mathbf{r}_f)^T \mathbf{c}}{\|\mathbf{r} - \mathbf{r}_f\| \cdot \|\mathbf{c}\|} \geq \cos(\beta)\}, \quad (2)$$

where $\mathbf{c} = [0, 0, 1]^T$ is the unit vector of glide-slope cone axis, β is the half-angle of the glide-slope cone, and $\mathbf{r}_f = [x_f, y_f, z_f]$ is the position vector of the selected landing point. Then, the glide-slope constraint in (2) can be simplified as

$$(z - z_f)^2 \sin^2 \beta - ((x - x_f)^2 + (y - y_f)^2) \cos^2 \beta \geq 0. \quad (3)$$

Besides, to guarantee that the selected landing point locates in the target region, we assume that there are n potential landing points in the target region, and for the i th potential landing point, its position vector is expressed as $\mathbf{r}_{fi}$. Then, via introducing n binary variables c_i , the constraints on the terminal states can be expressed as

$$\sum_{i=1}^n c_i = 1, \quad (4a)$$

$$\sum_{i=1}^n c_i (\mathbf{r}_f - \mathbf{r}_{fi}) = 0, \quad (4b)$$

$$c_i (c_i - 1) = 0, i = 1, \dots, n \quad (4c)$$

Finally, considering the feasibility of each potential landing point, the MLG problem can be formulated as

$$\min_{t_f, \mathbf{T}_c} \int_{t_0}^{t_f} \alpha \|\mathbf{T}_c(t)\| dt + \sum_{i=1}^n c_i J(\mathbf{r}_{fi}), \quad (5a)$$

$$\text{subject to } \dot{\mathbf{r}} = \mathbf{V}, \quad \dot{\mathbf{V}} = \mathbf{g} + \frac{\mathbf{T}_c}{m(t)}, \quad \dot{m} = -\alpha \|\mathbf{T}_c(t)\| \quad (5b)$$

$$0 \leq T_{\min} \leq \|\mathbf{T}_c(t)\| \leq T_{\max}, \quad (5c)$$

$$\mathbf{r}(t) \in \mathcal{C}, \quad \forall t \in [t_0, t_f], \quad (5d)$$

$$m(t_0) = m_{\text{wet}}, \quad m(t) \geq m_{\text{dry}}, \quad (5e)$$

$$\mathbf{r}(t_0) = \mathbf{r}_0, \quad \mathbf{V}(t_0) = \mathbf{V}_0, \quad (5f)$$

$$c_i(c_i - 1) = 0, \quad i = 1, \dots, n, \quad (5g)$$

$$\sum_{i=1}^n c_i = 1, \quad \sum_{i=1}^n c_i (\mathbf{r}_f - \mathbf{r}_{fi}) = 0, \quad (5h)$$

where m_{wet} is the gross mass of the vehicle at the beginning of the MLG phase, m_{dry} is the structural mass, $J(\mathbf{r}_{fi})$ is the estimated extra cost for the i th potential landing point. Our goal is to find the control vectors $\mathbf{T}_c(t)$ along the landing phase and its duration t_f to minimize the overall equivalent fuel consumption $\int_{t_0}^{t_f} \alpha \|\mathbf{T}_c(t)\| dt + \sum_{i=1}^n c_i J(\mathbf{r}_{fi})$ while satisfying all specified constraints. In the following section, the MILM is introduced to solve the MLG problem (5) onboard.

IV. Mixed-input Learning-based Method

A. Necessary Conditions of Optimality

For the MLG problem formulated in (5), the augmented Hamiltonian is expressed as

$$\begin{aligned} H = & \lambda_1 v_x + \lambda_2 v_y + \lambda_3 v_z + \lambda_4 \frac{\|\mathbf{T}_c(t)\| \sin \theta \cos \phi}{m} + \lambda_5 \frac{\|\mathbf{T}_c(t)\| \sin \theta \sin \phi}{m} + \lambda_6 (-g + \frac{\|\mathbf{T}_c(t)\| \cos \theta}{m}) \\ & - \lambda_7 \alpha \|\mathbf{T}_c(t)\| + \mu_1 ((z - z_f)^2 \sin^2 \beta - ((x - x_f)^2 + (y - y_f)^2) \cos^2 \beta) + \mu_2 (m - m_{\text{dry}}) \end{aligned} \quad (6)$$

and the first-order conditions of optimality are written as

$$\frac{\partial H}{\partial \|\mathbf{T}_c(t)\|} = \lambda_4 \frac{\sin \theta \cos \phi}{m} + \lambda_5 \frac{\sin \theta \sin \phi}{m} + \lambda_6 \frac{\cos \theta}{m} - \lambda_7 \alpha \quad (7a)$$

$$\frac{\partial H}{\partial \theta} = \frac{\|\mathbf{T}_c(t)\|}{m} (\lambda_4 \cos \theta \cos \phi + \lambda_5 \cos \theta \sin \phi - \lambda_6 \sin \theta) = 0 \quad (7b)$$

$$\frac{\partial H}{\partial \phi} = \frac{\|\mathbf{T}_c(t)\|}{m} \sin \theta (\lambda_5 \cos \phi - \lambda_4 \sin \phi) = 0 \quad (7c)$$

where $\lambda = [\lambda_1, \dots, \lambda_7]^T$ is the adjoint variable set associated with the dynamic constraints and $\mu = [\mu_1, \mu_2]^T$ is the Lagrange multiplier set associated with the state inequality constraints. The optimality condition in (7a) does not explicitly contain the thrust magnitude $\|\mathbf{T}_c(t)\|$, which yields the optimal control solution, expressed as

$$\|\mathbf{T}_c(t)\| = \begin{cases} T_{\min}, & \frac{\partial H}{\partial \|\mathbf{T}_c(t)\|} < 0 \\ T_{\max}, & \frac{\partial H}{\partial \|\mathbf{T}_c(t)\|} > 0 \\ \in [T_{\min}, T_{\max}], & \frac{\partial H}{\partial \|\mathbf{T}_c(t)\|} = 0 \end{cases} \quad (8)$$

As in our previous work [25], we have proved the following Lemma,

Lemma 1 (Lemma 1 in [25]) *For the fuel optimal MLG problem with glide-slope constraint, the optimal solution of the thrust magnitude is a bang-bang control with at most three sub-arcs, in the order max-min-max.*

Therefore, according to Lemma 1, for the studied problem (5), the optimal solution of the thrust magnitude is a bang-bang control, which can be expressed as

$$\|\mathbf{T}_c(t)\| = \begin{cases} T_{\min}, & \frac{\partial H}{\partial \|\mathbf{T}_c(t)\|} < 0 \\ T_{\max}, & \frac{\partial H}{\partial \|\mathbf{T}_c(t)\|} > 0 \end{cases} \quad (9)$$

In addition, we also have proved the following proposition in [25],

Proposition 1 (Proposition 1 in [25]) *For an optimal solution of the 3D MLG problem with glide-slope constraint, the glide-slope constraint will only be active with nontrivial duration $\Delta t > 0$ when the following equation is satisfied:*

$$\frac{1}{\sqrt{(\tan^2 \beta (\ddot{Z}^2 + 4\dot{Z}^2\dot{\gamma}^2 + Z^2\dot{\gamma}^4 + Z^2\ddot{\gamma}^2 - 2Z\ddot{Z}\dot{\gamma}^2 + 4\dot{Z}Z\ddot{\gamma}\dot{\gamma}) + \ddot{Z}^2 + 2g\ddot{Z} + g^2)}} = \frac{m}{\|\mathbf{T}_c(t)\|}. \quad (10)$$

For the studied MLG problem (5), we found that the equation (10) does not hold for any cases in the range we have studied so far. Therefore, the glide-slope constraint will only be active at isolated points with duration $\Delta t \rightarrow 0$ in the studies cases. Additionally, considering that the adjoint variables λ_i , $i = 1, \dots, 7$ do not necessarily have to be continuous at the isolated point [25], λ_i , $i = 1, \dots, 7$, are handled as piecewise functions and the switching points are the isolated points.

For the optimal control, when an inequality constraint is inactive, the corresponding Lagrange multiplier equals to zero. Thus, for the MLG problem, we have $\mu_1 = 0$ when the glide-slope constraint (2) is inactive. Therefore, when the glide-slope constraint is inactive, combining with the necessary conditions of adjoint variables, we have $\dot{\lambda} = -\frac{\partial H}{\partial \mathbf{r}}$, which can be extended as

$$\dot{\lambda}_1 = -\frac{\partial H}{\partial x} = 0 \quad (11a)$$

$$\dot{\lambda}_2 = -\frac{\partial H}{\partial y} = 0 \quad (11b)$$

$$\dot{\lambda}_3 = -\frac{\partial H}{\partial z} = 0 \quad (11c)$$

$$\dot{\lambda}_4 = -\frac{\partial H}{\partial v_x} = -\lambda_1 \quad (11d)$$

$$\dot{\lambda}_5 = -\frac{\partial H}{\partial v_y} = -\lambda_2 \quad (11e)$$

$$\dot{\lambda}_6 = -\frac{\partial H}{\partial v_z} = -\lambda_3 \quad (11f)$$

By integrating (11), it is straight forward to find the solutions of adjoint variables, written as

$$\lambda_1 = C_1, \lambda_2 = C_2, \lambda_3 = C_3, \lambda_4 = -C_1t + C_4, \lambda_5 = -C_2t + C_5, \lambda_6 = -C_3t + C_6, \quad (12)$$

where C_i , $i = 1, \dots, 6$, are constant and can be determined from the value of λ at the initial point or isolated point. Then, for each segment between the isolated point, via combining (12), (7b) and (7c), we have

$$\tan \phi = \frac{\lambda_5}{\lambda_4} \quad (13a)$$

$$\tan \theta = \frac{\lambda_4}{\lambda_6} (\cos \phi + \frac{\lambda_5}{\lambda_4} \sin \phi) = \frac{\lambda_4}{\lambda_6} (\cos \phi + \tan^2 \phi \cos \phi) = \frac{\lambda_4}{\lambda_6 \cos \phi}. \quad (13b)$$

By substituting the value of λ in (13) using the solution obtained from (12), the optimal thrust direction angles, ϕ and θ can be determined by

$$\tan \phi = \frac{-C_2t + C_5}{-C_1t + C_4} \quad (14a)$$

$$\cos \phi \tan \theta = \frac{-C_1t + C_4}{-C_3t + C_6}. \quad (14b)$$

Therefore, when C_i , $i = 1, \dots, 6$, are determined, we can compute ϕ and θ accordingly. Besides, with $\mu_1 = 0$, the optimality conditions of the MLG problem with the glide-slope constraint are the same as those of the MLG problem without the glide-slope constraint. In literature [29], it has been rigorously proved that under the constant-gravity assumption, the optimal solution of the minimum-fuel powered descent guidance problem without the glide-slope constraint has at most three sub-arcs, with the thrust magnitude in the order of max-min-max. In other words, there are at most two switching time points, shown in Fig. 1, where t_{s1} denotes the switching time from maximum thrust magnitude to minimum thrust magnitude, t_{s2} is the switching time from minimum thrust magnitude to maximum thrust magnitude, and t_f is the terminal time. For the cases that have only one sub-arc, t_{s1} and t_{s2} are set as the terminal time t_f if it is the maximum sub-arc, otherwise t_{s1} is set as the starting time t_0 and t_{s2} is set as the terminal time t_f . For cases that have two sub-arcs, t_{s2} is set as the terminal time t_f if the two sub-arc are in the order of max-min, otherwise t_{s1} is set as the starting time t_0 . Therefore, once t_{s1} , t_{s2} and t_f are obtained, the optimal control profile of the thrust magnitude, $\|\mathbf{T}_c(t)\|$, during the powered descent phase can be determined.

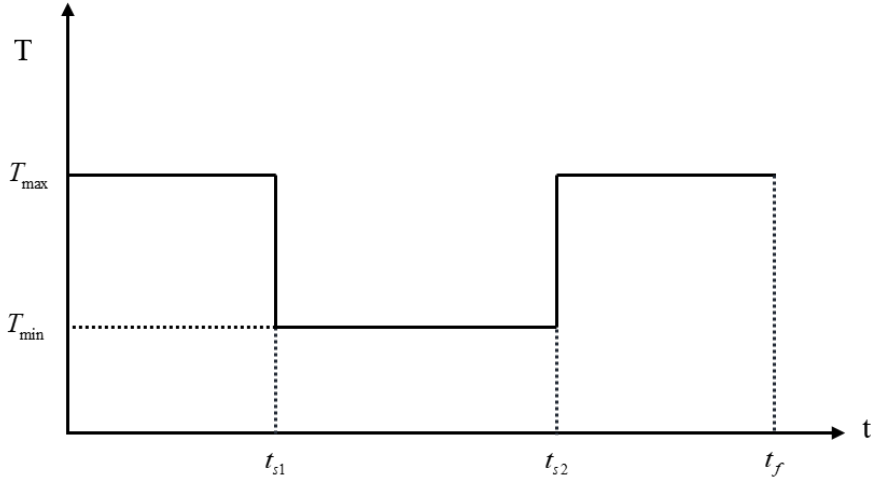


Fig. 1 Bang-bang control and switching time t_{s1} , t_{s2}

From the above analysis on necessary conditions, the optimal guidance law for the 3D MLG problem can be determined once the following parameters, which are named critical parameters, are obtained:

- 1) The switching time t_{s1} and t_{s2} for the bang-bang control;
- 2) Duration of the powered descent phase t_f ;
- 3) The glide-slope boundary touching time t_b for each segment;
- 4) The coefficients $C_1, C_2, C_3, C_4, C_5, C_6$ for each segment.

Note that, if the glide-slope constraint is not active along the optimized trajectory, the fitted curves for $\tan \phi^*$ and $\tan \theta^* \cos \phi^*$ only contain one segment, and the glide-slope boundary touching time t_b is set as t_f . After determining the above identified critical parameters, we can find the control switching time points from t_{s1}, t_{s2} and the duration of descent phase from t_f . Meanwhile, the piece-wise thrust direction angles θ and ϕ can be determined from corresponding $C_1, C_2, C_3, C_4, C_5, C_6$ according to (14). Then the time-continuous optimal control vector $\mathbf{T}_c(t)$ during the powered descent phase can be determined according to the switching time points and thrust direction angles θ and ϕ . With the optimal control profile determined, the state variables in continuous time can be obtained by integrating the vehicle dynamics in (1) from t_0 to t_f .

B. Dataset Generation

Based on the necessary conditions of optimality for the MLG problem, the identified critical parameters have been derived. Then, to realize the learning-based method, the next step is to generate a dataset offline, where sufficient inputs and outputs need to be determined and collected. In our previous work [25], the learning-based method has been applied to solve the pinpoint MLG problem. As the landing point is fixed for the pinpoint powered descent guidance problem, with a given initial position and velocity, the optimization problem only needs to generate an optimal path with a fixed terminal condition. Therefore, the inputs of the dataset are the initial position and velocity vectors, and the outputs of

the dataset are the identified critical parameters.

Different from the pinpoint powered descent guidance problem, one of the challenges for solving the MLG problem is the additional requirement for selecting a target point from the pre-selected target set once the landing area image is available. Therefore, to select a safe target point and generate the optimal path simultaneously in real-time, both the landing area image and the initial states are considered as mixed inputs.

The primary standard for judging whether the selected landing area is feasible is the flatness and altitude of the landing area. For the sake of simplicity, the contour map of the target region is adopted to represent the features of the target landing region, which is handled as an image input. In addition, to reflect the features of the initial state of the powered descent phase, the velocity and position vectors are adopted, which are taken as numeric inputs. Specifically, the size of the input image is selected to be $64 \times 64 \times 1$, which is shown in Fig. 2. It can be found the grayscale image is adopted in this paper, and the altitude of each point is represented via its grayscale. Note that, all hazards in the target landing area are randomly generated to imitate the Mars surface environments.

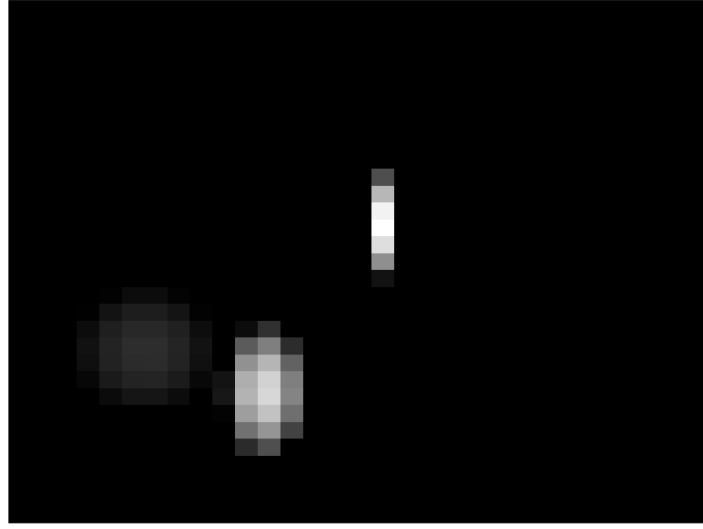


Fig. 2 The grayscale image of the target landing region

To generate the database, a unified algorithm based on iterative second-order cone programming (SOCP) is applied to solve the MLG problem off-line [30]. The details of the SOCP algorithm used to solve each MLG cases is described in Part I of this article [31]. Then, based on the obtained solutions, where the optimal states and controls are included, the critical parameters can be further derived. To be specific, from solutions of the unified algorithm, the critical parameters t_f , t_{s1} and t_{s2} can be obtained directly. In addition, for the MLG in the studied range, the optimal thrust magnitude will have only two sub-arcs, Min-Max. Therefore, only one switching time parameter needs to be considered here.

In addition, the remaining critical parameters, $C_1, C_2, C_3, C_4, C_5, C_6$, cannot be derived directly from the off-line optimal states and controls. From (14), it can be found that $C_1, C_2, C_3, C_4, C_5, C_6$ can be calculated by fitting the optimized thrust angle θ and ϕ at all discrete nodes. However, due to the integration errors introduced by discretization, the curve fitting results cannot get desired value of the critical parameters for most cases. Therefore, a nonlinear optimization problem is further formulated. To improve the accuracy of the identified critical parameters, both the final landing error and the glide-slope constraint are considered in the objective function, and the reformulated nonlinear optimization problem can be written as

$$J = \sum_{i=1}^{N_p} \max(0, x(i)^2 + y(i)^2 - z(i)^2 \tan^2 \beta) + \alpha \|\mathbf{V}(N_p) - \mathbf{0}\|^2 + \beta \|\mathbf{r}(N_p) - \mathbf{r}_f\|^2 \quad (15)$$

s.t. (5b), (5c), (5d), (5e), (5f), (5g), (5h).

where N_p refers to the number of discrete nodes during the MLG phase, α and β denote the coefficients for the velocity error and position error, respectively. Then, with all the constraints of (5) considered, the curve fitting problem can be further formulated as an optimization problem. Moreover, as only 20 discrete nodes are adopted for the unified algorithm, the obtained time parameters t_s and t_f are also not very precise. Therefore, the terminal time t_f and switching time t_s are also optimized to improve the accuracy of all identified critical parameters. Besides, (14) can be further simplified. Let the numerator and denominator on the right side divided by C_1 simultaneously, we can have

$$\tan \phi = \frac{-C'_2 + C'_5}{-t + C'_4} \quad (16a)$$

$$\cos \phi \tan \theta = \frac{-t + C'_4}{-C'_3 t + C'_6}. \quad (16b)$$

where $C'_i = C_i/C_1, i = 2, 3, 4, 5, 6$. In summary, for the MLG problem in the studied range, only 7 identified critical parameters are finally considered, including $C'_2, C'_3, C'_4, C'_5, C'_6, t_s, t_f$, which are the outputs of the generated dataset.

C. Hybrid Deep Neural Network

Considering the generated mixed-inputs dataset, where both image inputs and numeric inputs are included, the hybrid deep neural network is adopted for data training. The framework of the constructed hybrid deep neural network is shown in Fig. 3, containing two main parts. First, the convolutional neural network and multilayer perception network are constructed independently to deal with image inputs and numeric inputs independently. After that, the extracted features are concatenated and fed to multiple fully connected layers for regression.

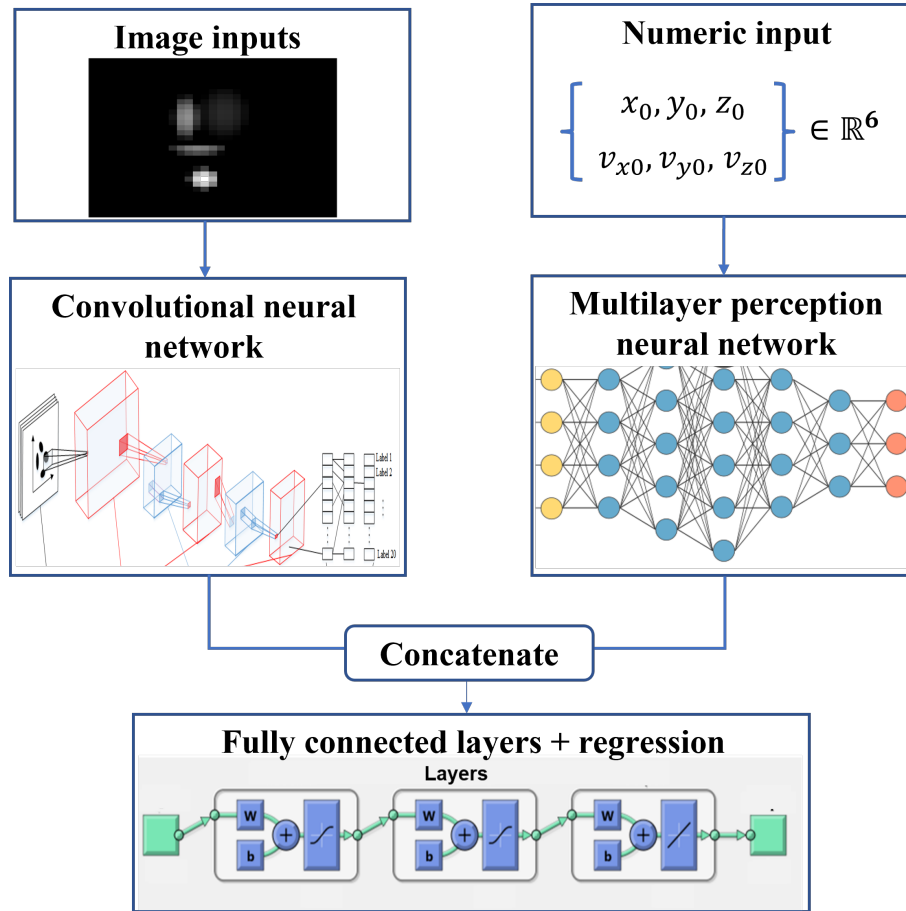


Fig. 3 The framework of the constructed hybrid deep neural network

For the general convolutional neural network, four typical types of hidden layers are considered and combined,

including convolutional layer, rectified linear unit (ReLU) layer, pooling layer, and batch normalization layer. For the convolutional layer, a convolution operation is included, where the input images are resized and features of each filter are collected for the following layer. In the ReLU layer, an activation function is applied, where the negatives of inputs are eliminated, which can be effective to filter out useless information. As for the pooling layer, the spatial space of the inputs is shrunk and the extracted features are remained, which is useful for reducing the computation load. Therefore, it is usually inserted between two convolutional layers. For the batch normalization layer, the inputs are normalized and the input data are divided into multiple mini-batch, which can dramatically reduce the number of training epochs required to train deep networks. Therefore, for the MLG problem, considering the size of the image inputs is $64 \times 64 \times 1$, by trying different combinations of the four typical types of hidden layers, 8 convolutional layers with increasing number of filters (32-32-64-64-128-128-256-256) are currently adopted. In addition, to avoid losing the image information when combined with numeric inputs, the max pooling layer is added after every two convolutional layers, and the global average pooling layer is added before concatenating.

For the numeric inputs, 1-2 hidden layers are adopted for the multilayer perception network, where the activation functions are combining the hyperbolic tangent function and the ReLU function. Considering that for some critical parameters, the loss function cannot always converge, the batch normalization layers are applied between input layer, hidden layer, and output layer. Besides, to further improve the accuracy, the critical parameters, which are also outputs of the hybrid deep neural network, are normalized to zero-mean vectors. To guarantee robust learning performance, 2-3 hidden fully connected layers are applied to the concentrated data, where the activation functions are also combination of the hyperbolic tangent function and the ReLU function.

D. Mixed-input Learning-based Method

The flowchart of the proposed MILM is presented in Table .1, which can be divided into two parts. The off-line part aims to generate the dataset and train the hybrid deep neural network, where four main steps are included. First, problem (5) is discretized and reformulated as a nonconvex quadratically constrained quadratic programming (QCQP) problem. Then, the unified algorithm proposed in [31] is adopted to find the discrete optimal solutions with different initial conditions and landing regions. Second, guided by the optimal control theory, a few critical parameters are identified to represent the optimal solution to the MLG problem. To be specific, 7 critical parameters are identified in this paper, including C'_2 , C'_3 , C'_4 , C'_5 , C'_6 , t_s , t_f . After that, via the reformulated nonlinear optimization problem (15), the values of these identified critical parameters are calculated. Via collecting the identified critical parameters as outputs, and the features of the desired target region and the initial states as inputs, the dataset is constructed. Finally, based on the generated dataset, the hybrid deep neural network is constructed and trained off-line to map the relationship between inputs and outputs.

For the on-line part, with the given initial states and descent image, the trained hybrid deep neural network is utilized to calculate the values of the identified critical parameters. After that, the demanded optimal control commands can be reconstructed from the calculated critical parameters in real-time.

V. Simulation

To verify the effectiveness of the proposed MILM, simulation results of MLG are presented and analyzed. In addition, the unified algorithm based on iterative SOCP [31] is also applied for comparison. In this paper, all simulations were run in Matlab environments on a 2.1 GHz Laptop with 16 GB RAM. The problem parameter settings are listed in the following Table 2 and can be referred to [25]:

To test the effectiveness of the proposed MILM, for the generated dataset, the initial states are randomly generated from the following ranges: $z(t_0) \in [6700, 8100]$ m, $x(t_0) = -1025$ m, $y(t_0) = -512$ m, $v_x(t_0) \in [19, 23]$ m/s, $v_y(t_0) \in [38, 46]$ m/s, $v_z(t_0) \in [-220, -190]$ m/s, and hazard areas on the landing region are also randomly created. Then, the size of the current dataset is selected to be 2624, where 20% are taken as testing dataset, 10% are taken as validation dataset, and the remaining are taken as training dataset.

To handle the mixed inputs, the hybrid deep neural networks are constructed via the Matlab deep learning toolbox [32]. The specified architecture of the constructed neural network for the critical parameter t_f is shown in Figure. 4. By changing the structure of the hybrid neural network slightly, e.g., the layers of the multilayer perception network, the hybrid neural network for the other critical parameters is obtained. To show the training results for all critical parameters, the root means square error (RMSE) of all critical parameters is shown in Table. 3. It can be found that all critical parameters can achieve similar precision. To further demonstrate the training results, the comparison of the learned value and desired value of the critical parameters t_f is presented in Fig. 5. In addition, all the compared cases

Table 1 Flowchart of mixed-input learning-based method**Algorithm:** Mixed-input learning-based method**Off-line part**

1. Generate the off-line discrete optimal solutions of problem (5) via the unified algorithm based on iterative SOCP;
2. Identify the critical parameters via the optimal control theory;
3. Extract values of the critical parameters based on (15), and generate the dataset;
4. Construct and Train the hybrid deep neural network to map the relationship between the inputs and outputs.
Specially, the inputs include the initial states of powered descent phase and the descent image of the target region.
the outputs are the identified critical parameters.

On-line part

5. According to the given inputs, calculate the values of the identified critical parameters via the trained hybrid neural network;
6. Reconstruct the optimal control solution using the identified critical parameters.

Table 2 Problem parameter settings

Parameter	$g \text{ (m/s}^2\text{)}$	$m_{\text{pdi}} \text{ (kg)}$	$\alpha \text{ (s/m}^2\text{)}$	$m_{\text{dry}} \text{ (kg)}$	$T_{\text{max}} \text{ (kN)}$	$T_{\text{min}} \text{ (kN)}$
Value	-3.711400	51099	4.53×10^{-5}	40880	640	240

are taken from the testing dataset. From Fig. 5, it can be found that the terminal time t_f is well fitted, where the mean absolute relative error is 0.7%, and the mean absolute error is 0.31 seconds. Therefore, we can conclude that the trained hybrid neural networks have relatively high precision.

Table 3 Training results for all critical parameters

Parameter	C'_2	C'_3	C'_4	C'_5	C'_6	t_s	t_f
RMSE	0.103	0.196	0.09	0.175	0.187	0.102	0.105

With the well-trained hybrid neural networks, the MLG problem is simulated, where $[x(t_0), y(t_0), z(t_0)] = [-1025, -512, 7403.9] \text{ m}$, $[v_x(t_0), v_y(t_0), v_z(t_0)] = [21.1, 42.0, -203.75] \text{ m/s}$, and the grayscale image of the target landing region is shown in Fig. 6a. The results from MILM and the unified algorithm are shown in Fig. 6b. It can be found that both MILM and the unified algorithm can select a feasible landing point and achieve safe landing. The fuel consumptions of MILM and the unified algorithm are 8.19 tons and 8.14 tons, respectively, where the relative error is only 0.6%. In other words, the proposed MILM can also achieve near optimal solutions. Besides, the time histories of thrust, velocity, mass, and position of both methods are presented in Figs. 7 - 9. The landing velocity error of the MILM is $[v_x(t_f), v_y(t_f), v_z(t_f)] = [1.48, 8.71, 0.57] \text{ m/s}$, and the terminal altitude error of the MILM is $z(t_f) = 22.9 \text{ m}$. the computational time for the MILM method is less than 0.2 seconds, while that of the unified algorithm is 30 seconds. In conclusion, the real-time performance and the landing accuracy of the proposed MILM are further verified.

VI. Conclusion

This paper develops a mixed-input learning-based method for solving the multi-point landing guidance (MLG) problem in real-time. First, the MLG problem is formulated as an optimal control problem with quadratic constraints via introducing binary decision variables. Then, supported by the necessary conditions for optimality of MLG, the

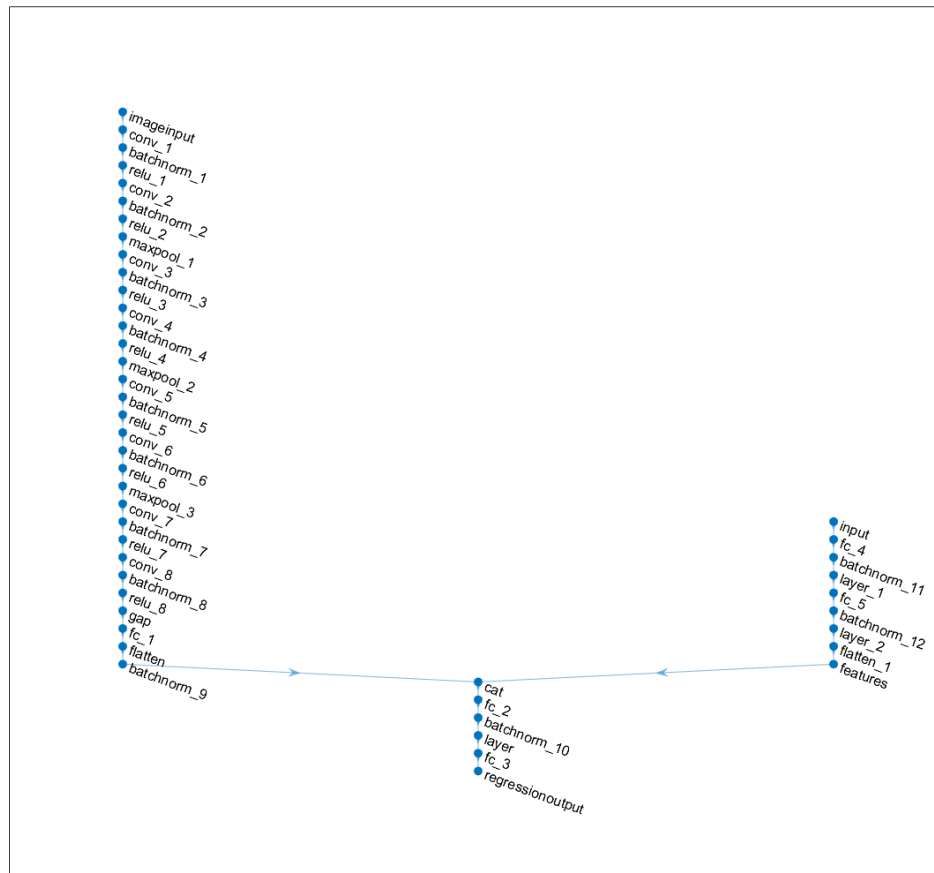


Fig. 4 The graph plot of specified architecture of the constructed hybrid neural network for t_f

optimal guidance law of the MLG is represented by a few critical parameters. After that, a mixed-input deep neural network is constructed and trained to map the implicit relationship between the problem inputs and the identified critical parameters. Moreover, the mixed-input learning-based method can solve the MLG problem in less than 0.2 seconds, which verifies its potential for on-board application.

References

- [1] Scharf, D. P., Ploen, S. R., and Acikmese, B. A., "Interpolation-enhanced powered descent guidance for onboard nominal, off-nominal, and multi-x scenarios," *AIAA Guidance, Navigation, and Control Conference*, 2015, p. 0850.
- [2] Robertson, E. A., "Synopsis of precision landing and hazard avoidance (PL&HA) capabilities for space exploration," *AIAA Guidance, Navigation, and Control Conference*, 2017, p. 1897.
- [3] Trawny, N., Benito, J., Tweddle, B. E., Bergh, C. F., Khanoyan, G., Vaughan, G., Zheng, J., Villalpando, C., Cheng, Y., Scharf, D. P., et al., "Flight testing of terrain-relative navigation and large-divert guidance on a VTVL rocket," *AIAA SPACE 2015 Conference and Exposition*, 2015, p. 4418.
- [4] Ma, L., Wang, K., Xu, Z., Shao, Z., Song, Z., and Biegler, L. T., "Multi-point powered descent guidance based on optimal sensitivity," *Aerospace Science and Technology*, Vol. 86, 2019, pp. 465–477.

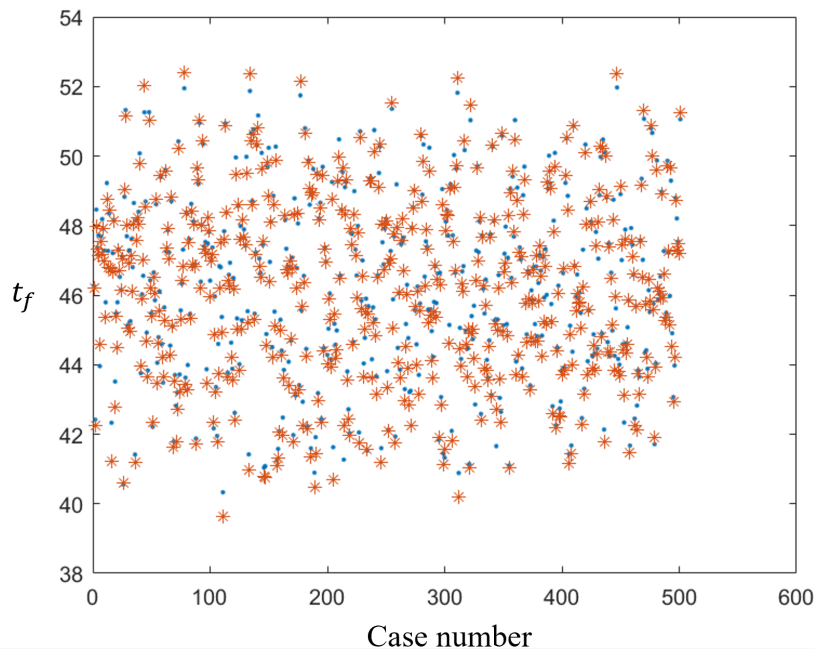
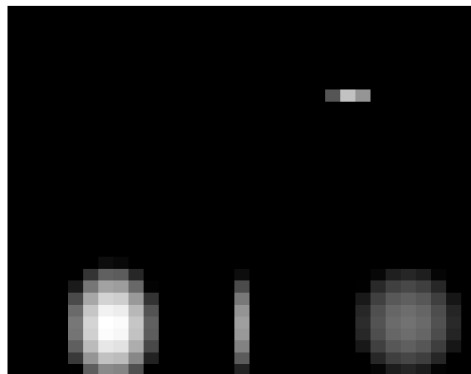
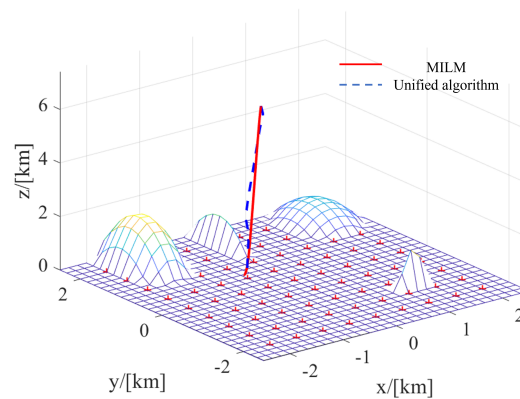


Fig. 5 The comparison between the learned value and the desired value of t_f



(a) The grayscale image of the target landing region



(b) The trajectory plots of the simulation cases

Fig. 6 The plot of target landing region and the 3D trajectory

- [5] Wolf, A. A., Acikmese, B., Cheng, Y., Casoliva, J., Carson, J. M., and Ivanov, M. C., "Toward improved landing precision on Mars," *2011 Aerospace Conference*, IEEE, 2011, pp. 1–8.
- [6] Cui, P., Ge, D., and Gao, A., "Optimal landing site selection based on safety index during planetary descent," *Acta Astronautica*, Vol. 132, 2017, pp. 326–336.
- [7] Ma, L., Wang, K., Shao, Z., Song, Z., and Biegler, L. T., "Trajectory optimization for planetary multi-point powered landing," *IFAC-PapersOnLine*, Vol. 50, No. 1, 2017, pp. 8291–8296.
- [8] Ploen, S. R., Seraji, H., and Kinney, C. E., "Determination of spacecraft landing footprint for safe planetary landing," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 45, No. 1, 2009, pp. 3–16.

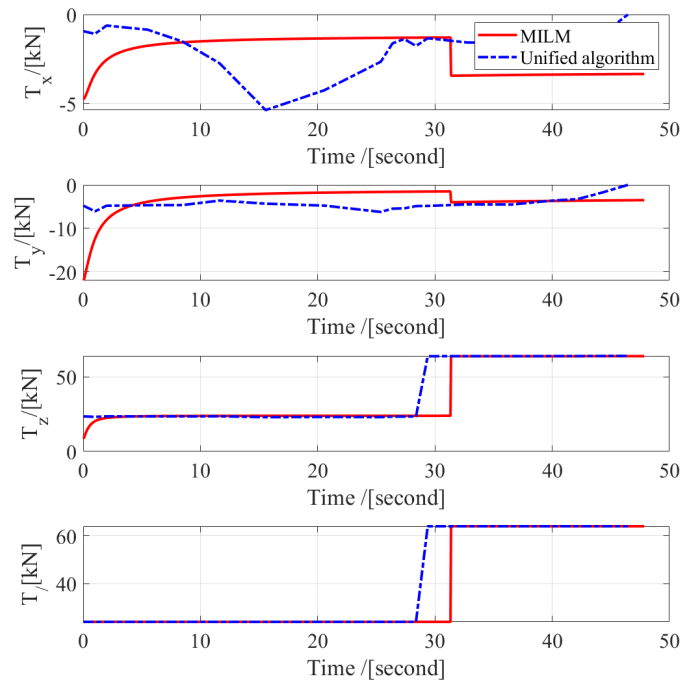


Fig. 7 The time history of thrust components and magnitude from MILM and the unified algorithm

- [9] Fahroo, F., and Ross, I. M., "Direct trajectory optimization by a Chebyshev pseudospectral method," *Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 1, 2002, pp. 160–166.
- [10] Bertsekas, D. P., "Nonlinear programming," *Journal of the Operational Research Society*, Vol. 48, No. 3, 1997, pp. 334–334.
- [11] Josselyn, S., and Ross, I. M., "Rapid verification method for the trajectory optimization of reentry vehicles," *Journal of Guidance, Control, and Dynamics*, Vol. 26, No. 3, 2003, pp. 505–508.
- [12] Schierman, J., and Hull, J., "In-flight entry trajectory optimization for reusable launch vehicles," *AIAA guidance, navigation, and control conference and exhibit*, 2005, p. 6434.
- [13] Liu, X., Shen, Z., and Lu, P., "Entry trajectory optimization by second-order cone programming," *Journal of Guidance, Control, and Dynamics*, Vol. 39, No. 2, 2016, pp. 227–241.
- [14] Lee, U., and Mesbahi, M., "Constrained autonomous precision landing via dual quaternions and model predictive control," *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 2, 2017, pp. 292–308.
- [15] Scharf, D. P., Regehr, M. W., Vaughan, G. M., Benito, J., Ansari, H., Aung, M., Johnson, A., Casoliva, J., Mohan, S., Dueri, D., et al., "ADAPT demonstrations of onboard large-divert guidance with a VTVL rocket," *2014 IEEE Aerospace Conference*, IEEE, 2014, pp. 1–18.
- [16] Scharf, D. P., Açıkmüş, B., Dueri, D., Benito, J., and Casoliva, J., "Implementation and experimental demonstration of onboard powered-descent guidance," *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 2, 2017, pp. 213–229.
- [17] Beard, R. W., Saridis, G. N., and Wen, J. T., "Galerkin approximations of the generalized Hamilton-Jacobi-Bellman equation," *Automatica*, Vol. 33, No. 12, 1997, pp. 2159 – 2177.
- [18] Saridis, G. N., and Lee, C.-S. G., "An approximation theory of optimal control for trainable manipulators," *IEEE Transactions on systems, Man, and Cybernetics*, Vol. 9, No. 3, 1979, pp. 152–159.
- [19] Evans, M., and Swartz, T., *Approximating integrals via Monte Carlo and deterministic methods*, Vol. 20, OUP Oxford, 2000.

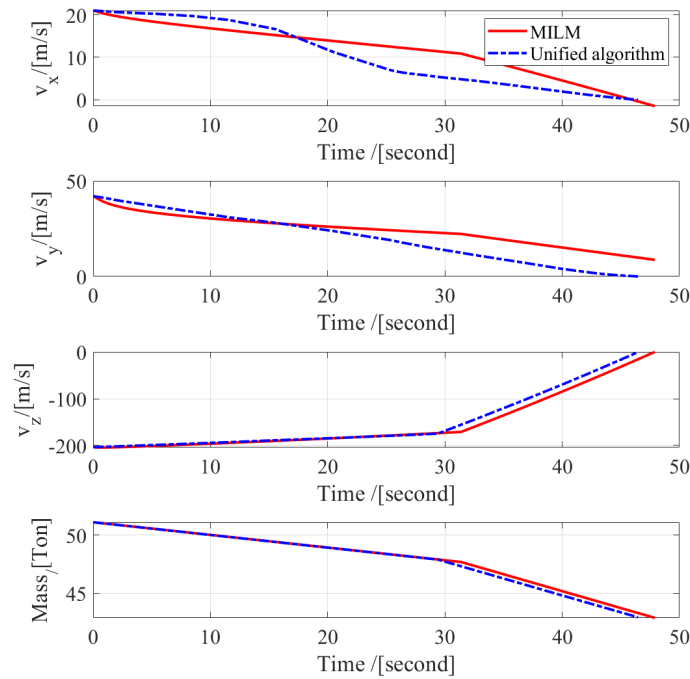


Fig. 8 The time history of velocity components and mass from MILM and the unified algorithm

- [20] Bock, H., Diehl, M., Leineweber, D., and Schlöder, J., “A direct multiple shooting method for real-time optimization of nonlinear DAE processes,” *Nonlinear model predictive control*, Springer, 2000, pp. 245–267.
- [21] Tsai, Y.-H. R., Cheng, L.-T., Osher, S., and Zhao, H.-K., “Fast sweeping algorithms for a class of Hamilton–Jacobi equations,” *SIAM journal on numerical analysis*, Vol. 41, No. 2, 2003, pp. 673–694.
- [22] Sánchez-Sánchez, C., and Izzo, D., “Real-time optimal control via Deep Neural Networks: study on landing problems,” *Journal of Guidance, Control, and Dynamics*, Vol. 41, No. 5, 2018, pp. 1122–1135.
- [23] Furfaro, R., Bloise, I., Orlandelli, M., Di Lizia, P., Topputo, F., Linares, R., et al., “A Recurrent Deep Architecture for Quasi-Optimal Feedback Guidance in Planetary Landing,” *IAA SciTech Forum on Space Flight Mechanics and Space Structures and Materials*, 2018, pp. 1–24.
- [24] Gaudet, B., Linares, R., and Furfaro, R., “Deep reinforcement learning for six degree-of-freedom planetary powered descent and landing,” *arXiv preprint arXiv:1810.08719*, 2018.
- [25] You, S., Wan, C., Dai, R., and Rea, J. R., “Learning-based onboard guidance for fuel-optimal powered descent,” *Journal of Guidance, Control, and Dynamics*, Vol. 44, No. 3, 2021, pp. 601–613.
- [26] Kumar, A., and Sachdeva, N., “Multi-input integrative learning using deep neural networks and transfer learning for cyberbullying detection in real-time code-mix data,” *Multimedia systems*, 2020, pp. 1–15.
- [27] Herault, J., and Jutten, C., “Space or time adaptive signal processing by neural network models,” *AIP conference proceedings*, Vol. 151, American Institute of Physics, 1986, pp. 206–211.
- [28] Bacchi, S., Gluck, S., Tan, Y., Chim, I., Cheng, J., Gilbert, T., Jannes, J., Kleinig, T., and Koblar, S., “Mixed-data deep learning in repeated predictions of general medicine length of stay: a derivation study,” *Internal and emergency medicine*, Vol. 16, No. 6, 2021, pp. 1613–1617.
- [29] Lu, P., “Propellant-Optimal Powered Descent Guidance,” *Journal of Guidance, Control, and Dynamics*, Vol. 41, No. 4, 2018, pp. 813–826.

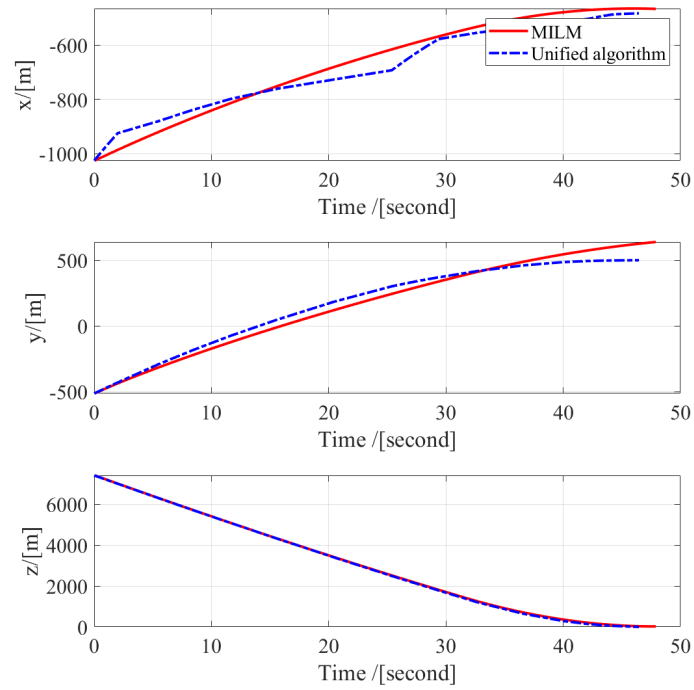


Fig. 9 The time history of position components from MILM and the unified algorithm

- [30] Pei, C., You, S., Dai, R., and Rea, J. R., "A Unified Optimization Algorithm for Bang-bang Optimal Control," *AIAA SCITECH 2022 Forum*, 2022, p. 0953.
- [31] Pei, C., You, S., Dai, R., and Rea, J. R., "Mixed-Input Learning for Multi-point Landing Guidance with Hazard Avoidance Part I: Offline Mission Planning based on Multi-Stage Optimization," *AIAA SCITECH 2022 Forum*, under review, 2023.
- [32] Kim, P., "Matlab deep learning," *With machine learning, neural networks and artificial intelligence*, Vol. 130, No. 21, 2017.