mathematics

MDPI

Article

Deep Neural Network-Based Footprint Prediction and Attack Intention Inference of Hypersonic Glide Vehicles

Jingjing Xu, Changhong Dong and Lin Cheng *

School of Astronautics, Beihang University, Beijing 100191, China

* Correspondence: chenglin5580@buaa.edu.cn

Abstract: In response to the increasing threat of hypersonic weapons, it is of great importance for the defensive side to achieve fast prediction of their feasible attack domain and online inference of their most probable targets. In this study, an online footprint prediction and attack intention inference algorithm for hypersonic glide vehicles (HGVs) is proposed by leveraging the utilization of deep neural networks (DNNs). Specifically, this study focuses on the following three contributions. First, a baseline multi-constrained entry guidance algorithm is developed based on a compound bank angle corridor, and then a dataset containing enough trajectories for the following DNN learning is generated offline by traversing different initial states and control commands. Second, DNNs are developed to learn the functional relationship between the flight state/command and the corresponding ranges; on this basis, an online footprint prediction algorithm is developed by traversing the maximum/minimum ranges and different heading angles. Due to the substitution of DNNs for multiple times of trajectory integration, the computational efficiency for footprint prediction is significantly improved to the millisecond level. Third, combined with the predicted footprint and the hidden information in historical flight data, the attack intention and most probable targets can be further inferred. Simulations are conducted through comparing with the state-of-the-art algorithms, and results demonstrate that the proposed algorithm can achieve accurate prediction for flight footprint and attack intention while possessing significant real-time advantage.

Keywords: reentry guidance; footprint prediction; attack intention inference; deep neural network

MSC: 85; 97R40

check for updates

Citation: Xu, J.; Dong, C.; Cheng, L. Deep Neural Network-Based Footprint Prediction and Attack Intention Inference of Hypersonic Glide Vehicles. *Mathematics* **2023**, *11*, 185. https://doi.org/10.3390/math11010185

Academic Editor: Daniel-Ioan Curiac

Received: 18 November 2022
Revised: 23 December 2022
Accepted: 23 December 2022
Published: 29 December 2022

1. Introduction

Hypersonic glide vehicles have attracted much attention in recent decades due to their dominant advantages on fast speed, wide attack range, and strong maneuverability. The whole flight time from the beginning entry to the final attack can be shortened to within one hour, which results in a daunting challenge for the interception system of the defensive side. Acquiring the feasible footprint and possible intention of an attacking HGV as early as possible and getting more time for the interception system is crucial for a successful defense. However, most of the existing methods/algorithms suffer the drawbacks of insufficient prediction accuracy and poor real-time performance. In this study, we focus on an online footprint prediction and attack intent inference algorithm for HGVs by leveraging the utilization of deep neural networks, and we achieve performance improvement for both prediction accuracy and computational speed.

The footprint is the set of terminals of all possible trajectories. Traditionally, the footprint is generated by traversing the maximum/minimum ranges and different heading angles using a baseline trajectory planner. The method difference for footprint generation is mainly reflected in the method difference for the planner design. The methods for footprint generation can be roughly divided into three categories. (1) Trajectory optimization methods, such as the Legendre pseudospectral method [1,2], the Gauss pseudospectral method [3], the hp-adaptive Radau pseudospectral method, and convex optimization [4–6] are utilized to generate the landing footprint. (2) The quasi-equilibrium glide condition (QEGC) [7] simplifies footprint generation. In [8], footprint generation is simplified to find the solutions to closest approaches to a moving virtual target. On the basis of [8], a selection scheme of a virtual target is proposed to increase the applicability of this method [9]. In [10], based on the simplified dynamics model, the convex optimization is utilized to generate the maximum crossrange trajectory. Footprint generation under the failure of the control components was solved by QEGC in [11]. (3) Based on the Evolved Acceleration Guidance Logic for Entry (EAGLE), drag acceleration-energy profiles can be designed for footprint generation [12]. On the basis of EAGLE, footprint generation algorithms follow drag acceleration–energy profiles tracking scheme satisfying the no-fly zone constraint in [13]. However, the existing footprint generation methods inevitably require numerical integration, and the time-consuming integration calculation and algorithm iteration cannot meet the real-time requirements for online footprint generation.

At the same time, few published studies pay attention to the discussion about attack intention for an attacking HGV, which specifically refers to information mining from past flight data and assists the defensive side to deploy interceptor systems for the most possible attack targets. Representative work is reviewed here. On the basis of the traditional state extrapolation prediction ideas in [14,15], a Bayesian trajectory prediction method based on intention inference is proposed. Aiming at the uncertainty of HGVs maneuvering, it is one of the effective ways to improve the accuracy of long-term trajectory prediction to reasonably infer the flight intention based on the characteristics of target motion. The trajectory prediction accuracy of this method is high in the short term, but the long-term prediction accuracy is low. The intention analysis only provides a reference for the long-term trajectory prediction, and cannot analyze the final attack target. In [16], a dynamic Bayesian network is used to infer the relationship between HGVs and attack targets to achieve attack intention prediction. However, this method can only determine the final attack target in the middle and later stages of the flight and cannot provide guidance for early warning and defense. DNN-based maneuver pattern recognition, such as penetration, attack, transportation, civil aviation flight, reconnaissance, etc., is designed in [17]. In summary, there are few studies to discuss the online attack intention for a flying HGV, and the main reasons are summarized as follows. (1) The real-time requirement for online footprint generation cannot be met. (2) The control strategy is unknown. Due to the aerodynamic force, HGVs perform non-inertial maneuver driven by the control command, and it is very difficult to exactly identify their control strategies. (3) HGVs are highly maneuverable. HGVs can theoretically attack every target within a large footprint through maneuvering changes. Due to the restrictions of the no-fly zone and the need for maneuver penetration, it is unavoidable that the control variable may change sharply. Compared with ballistic missiles, the intention analysis for HGVs is more difficult due to HGVs' strong maneuverability.

In recent years, DNNs have been widely used in the aerospace field [18]. Multi-layer feedforward neural networks are utilized to approximate the mapping relationship between the real-time flight states of high lifting vehicles and guidance commands in [19]. In [20,21], a neural network predictor assists in calculating guidance parameters. In our previous study [22], DNN is developed to replace the trajectory integrator to help achieve real-time numerical predictor–corrector guidance (NPCG). In this study, we focus on the online prediction of feasible footprint and attack intention for HGVs, and DNN and data mining technologies help achieve the combined advantages on real-time performance and prediction accuracy. Specifically, the following three contributions are emphasized. (1) A baseline multi-constrained entry guidance algorithm is developed based on a compound bank angle corridor, and then a dataset containing enough trajectories for the following DNN learning is generated offline by traversing different initial states and control commands. (2) DNNs are developed to learn the functional relationship between the flight state/command and the corresponding ranges; on this basis, an online footprint prediction algorithm is developed by traversing the maximum/minimum ranges and different heading angles. Due to the substitution of DNN for trajectory integration, the computational efficiency for footprint prediction is significantly improved to the millisecond level. (3) Combined with the predicted footprint and the hidden information in historical flight data, the attack intention and most probable targets can be further inferred. A forgetting mechanism helps the proposed attack intention inference algorithm to still be effective when the HGVs change their attack target during the flight. Simulations are given to estimate the effectiveness of the proposed techniques.

This study is organized as follows: The problem formulation of reentry is described in Section 2. In Section 3, a DNN is developed to approximate the ranges, following which an intelligent DNN-based footprint algorithm is proposed. In Section 4, an intent inference algorithm is proposed. Simulations are given in Section 5 to evaluate the performance of the proposed algorithm. Section 6 summarizes this study.

2. Problem Formulation

The purpose of this study is to achieve online footprint generation and attack intention inference of an enemy HGV. In this section, reentry dynamics, constraints, and control parameterization are provided and analyzed.

2.1. Reentry Dynamics

In order to achieve the footprint prediction of a flying HGV, it is necessary to describe its reentry dynamical motion. Without considering the influence of the Earth's rotation, the three-degree-of-freedom motion model is given as [23]:

$$\begin{aligned}
&\dot{r} = v\sin\theta \\
&\dot{\lambda} = v\cos\theta\sin\psi/(r\cos\phi) \\
&\dot{\phi} = v\cos\theta\cos\psi/r \\
&\dot{v} = -D/m_0 - g\sin\theta \\
&\dot{\theta} = (1/v)\left[L\cos\sigma/m_0 + \left(v^2/r - g\right)\cos\theta\right] \\
&\dot{\psi} = (1/v)\left[L\sin\sigma/(m_0\cos\theta) + v^2/r\cos\theta\sin\psi\tan\phi\right] \\
&\dot{S}_e = v\cos\theta/r
\end{aligned} \tag{1}$$

where r represents the geocentric distance of the vehicle, λ and ϕ represent the longitude and latitude, v is the speed of the vehicle relative to the Earth, θ represents the trajectory inclination angle, ψ represents the heading angle. S_e is the cumulative range angle of the vehicle; m_0 is the mass of the vehicle, which remains constant during reentry, σ denotes the bank angle, and g is the acceleration of gravity, which is calculated by a simple inverse square model [22]:

$$g = \frac{R_0^2}{(R_0+h)^2}g_0, \tag{2}$$

where $g_0 = 9.8\ \mathrm{m/s^2}$ is the gravitational acceleration at sea level, and $R_0 = 6378$ km is the average radius of the Earth.

The atmospheric density ρ is expressed as [24,25]:

$$\rho = \rho_0 e^{-h/\beta}, \tag{3}$$

where $\rho_0 = 1.225\ \mathrm{kg/m^3}$ represents the atmospheric density at sea level, and $\beta = 7200$ m.

The variables L and D represent the lift and drag of the vehicle, and the expressions are [23]:

$$L = 0.5\rho v^2 C_L S_{ref}, \tag{4}$$

$$D = 0.5\rho v^2 C_D S_{ref}, \tag{5}$$

where S_{ref} represents the aerodynamic reference area of the vehicle, and C_L and C_D represent the lift and drag coefficients of the vehicle, which are related to the speed of the vehicle and the angle of attack α. During reentry, α takes the form of a three-section profile related to flight speed v [22]:

$$\alpha = \begin{cases} \alpha_{\max} & v > v_1 \\ (\alpha_{\max} - \alpha_{\min})(v - v_2)/(v_1 - v_2) & v_2 < v < v_1 \\ \alpha_{\min} & v < v_2 \end{cases} \tag{6}$$

where $\alpha_{\max} = 20^\circ$ is the maximum allowable angle of attack, $\alpha_{\min} = 8.5^\circ$ is the angle of attack at the maximum lift-to-drag ratio, and $v_1 = 4700$ m/s, $v_2 = 3100$ m/s are velocity nodes.

As a result, the bank angle σ is the only control variable for reentry; $\mathbf{x}_{state} = [h, \theta, \phi, v, \gamma, \psi]^T$ represents the current state of the vehicle.

2.2. Reentry Constraints

In order to ensure the safety and meet the mission requirements of the vehicles, the HGVs need to meet the path constraints and terminal constraints during reentry. Path constraints include heating rate constraint, overload constraint, dynamic pressure constraint, and equilibrium glide condition. These expressions are as follows [23]:

$$\dot{Q}(t) = \frac{C_1}{\sqrt{R_d}} \left(\frac{\rho}{\rho_0}\right)^{0.5} \left(\frac{v}{V_C}\right)^{3.15} \leq \dot{Q}_{\max}, \tag{7}$$

$$n(t) = \sqrt{\left[\frac{L}{m_0 g_0}\right]^2 + \left[\frac{D}{m_0 g_0}\right]^2} = q(t)\sqrt{C_L^2 + C_D^2} \frac{S_{ref}}{m_0 g_0} \leq n_{\max}, \tag{8}$$

$$q(t) = \frac{1}{2}\rho v^2 \leq q_{\max}, \tag{9}$$

$$L \cos \sigma_{QEGC} / m_0 + \left(v^2 / r - g\right) = 0, \tag{10}$$

where $\dot{Q}(t)$ represents the heating rate at the stagnation point, $\dot{Q}_{\max}$ is the upper limit of the allowable heating rate, C_1 and R_d are the overall design parameters of the vehicle, V_c represents the first cosmic velocity, $n(t)$ represents the actual total overload, n_{max} is the upper limit of allowable overload, $q(t)$ represents the actual dynamic pressure, q_{max} is the upper limit of allowable dynamic pressure, and σ_{QEGC} is the equilibrium glide angle.

To ensure a successful transition to the terminal area energy management (TAEM), the final reentry segment must meet specific position and velocity requirements. Terminal constraints include [23]:

$$h\left(t_f\right) = h_f,\ v\left(t_f\right) = v_f,\ \lambda\left(t_f\right) = \lambda_f,\ \phi\left(t_f\right) = \phi_f, \tag{11}$$

where h_f, v_f, λ_f, ϕ_f are the altitude, speed, longitude, and latitude of the end of the reentry flight.

Terminal latitude and longitude constraints are usually transformed into the range constraint $S_e(t_f) = S_{go}$; S_{go} is defined as the spherical distance from the vehicle to the target [22]:

$$S_{go} = \arccos\left[\sin\lambda \sin\lambda_f + \cos\lambda \cos\lambda_f \cos\left(\phi - \phi_f\right)\right]. \tag{12}$$

2.3. Control Parameterization

In this subsection, the noted reentry constraints in Section 2.2 are transformed into the upper and lower boundaries for the bank angle. Then, by weighting the upper and lower boundaries, the bank angle profile can be determined, which corresponds to trajectories of different ranges. By traversing different initial reentry conditions and different weighting coefficients, a dataset composed of trajectories can be generated offline and used to train DNNs for range prediction.

2.3.1. Bank Angle Corridor

HGVs need to satisfy path constraints, terminal constraints, and control constraints. Constraining trajectories within reasonably constructed corridors is a way to deal with multiple constraints. Common corridors include height–velocity corridors, bank angle corridors, drag acceleration corridors, etc. In this study, a compound bank angle corridor is employed [22]. We substitute the atmospheric density Equation (3) and aerodynamic Equations (4) and (5) into the heating rate, overload, and dynamic pressure constraint Equations (7)–(9). Denoting $H_{\dot{Q}_{\max}}$, $H_{n_{\max}}$, $H_{q_{\max}}$ as the height boundary of the maximum heating rate, overload, and dynamic pressure according to the speed v, as shown in Figure 1, the height constraint according to v can be obtained as

$$H > \frac{2}{\beta} \ln\left[\frac{C_1}{\dot{Q}_{\max}\sqrt{R_d}}\left(\frac{V}{V_C}\right)^{3.15}\right] = H_{\dot{Q}_{\max}}(V), \tag{13}$$

$$H > \frac{1}{\beta} \ln\left[\frac{\rho_0 V^2 S\sqrt{C_D^2 + C_L^2}}{2n_{\max} m_0 g_0}\right] = H_{n_{\max}}(V), \tag{14}$$

$$H > \frac{1}{\beta} \ln\left(\frac{\rho_0 V^2}{2q_{\max}}\right) = H_{q_{\max}}(V), \tag{15}$$

$$H(V) > H_{\text{down}}(V) = \max\left(H_{\dot{Q}_{\max}}, H_{n_{\max}}, H_{q_{\max}}\right). \tag{16}$$

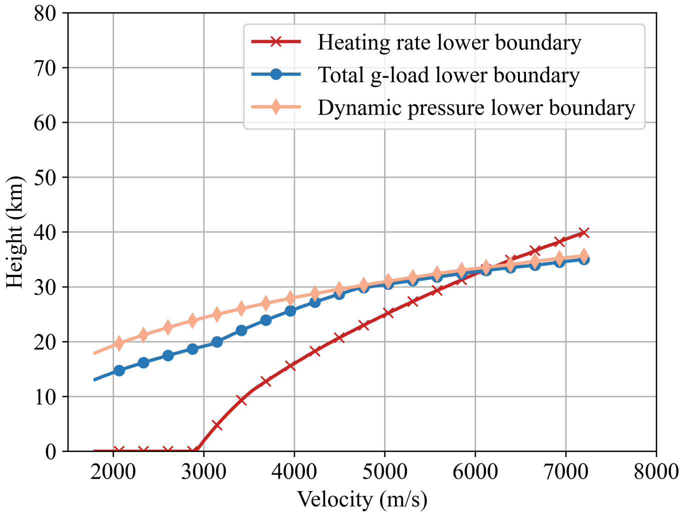


Figure 1. Lower boundary of the height–velocity corridor.

According to the QEGC, with known height and speed, the equilibrium glide angle σ_{QEGC} can be determined as

$$\sigma_{QEGC} = \cos^{-1}\frac{m_0(g - V^2/r)}{L}. \tag{17}$$

The $H_{down}(V)$ is a function of speed, and the $\sigma_{QEGC_{up}}$ corresponding to $H_{down}(V)$ can be obtained by Equation (17). The definition of QEGC determines

$$\sigma_{QEGC_{down}}(V) = 0. \tag{18}$$

Therefore, the bank angle needs to be less than $\sigma_{QEGC_{up}}(V)$ and greater than $\sigma_{QEGC_{down}}(V)$:

$$\sigma_{QEGC_{down}}(V) \leq \sigma(V) \leq \sigma_{QEGC_{up}}(V). \tag{19}$$

So far, we have obtained the upper and lower bounds of the bank angle corridor for the incoming HGVs to satisfy the path constraints. In the initial glide segment, due to the high flight altitude, the flight aerodynamic force is insufficient. In order to prevent the vehicle from falling too fast to generate a large amount of aerodynamic heat, the bank angle keeps a small constant value in the initial glide segment. In this paper, σ_{Imax} represents the maximum allowable bank angle amplitude during the initial glide segment. It is determined so that the heating rate in the initial glide segment is exactly equal to $\dot{Q}_{\max}$, that is, $\max(\dot{Q}(\sigma_{I\max})) = \dot{Q}_{\max}$. The velocity at the end of the initial glide segment is expressed as the velocity corresponding to the maximum heating rate, denoted as v_{If}.

In order to ensure the smooth handover of the reentry flight and the TAEM, the terminal of the reentry flight must satisfy both the range constraints and the terminal constraints. Combined with the QEGC, the terminal equilibrium glide angle σ_{TAEM} can be obtained by bringing the terminal states h_f, v_f into Equation (10).

Based on the above analysis, a bank angle corridor must comprehensively consider path constraints and terminal constraints. In the initial glide segment, the constant bank angle needs to be less than $\sigma_{I\max}$, and in the equilibrium glide segment, the bank angle profile is restricted between $\sigma_{E\max}$ and $\sigma_{E\min}$. In addition, in order to ensure the terminal altitude h_f and speed v_f constraints, the bank angle at the end of reentry is set as σ_{TAEM}. Considering the analysis results of the three sections, the bank angle corridor of the entire reentry flight can finally be obtained, where the upper bound is composed of $\sigma_{I\max}$, $\sigma_{E\max}$, and σ_{TAEM}, and the lower bound is composed of $\sigma_{E\min}$ and σ_{TAEM}. The final compound bank angle corridor is shown in Figure 2, with the upper bound denoted as $\sigma_{\max}(v)$ and the lower bound denoted as $\sigma_{\min}(v)$. In the process of trajectory planning, as long as the bank angle is limited within the corridor, the path constraints and the terminal height and speed constraints are satisfied to a certain extent.

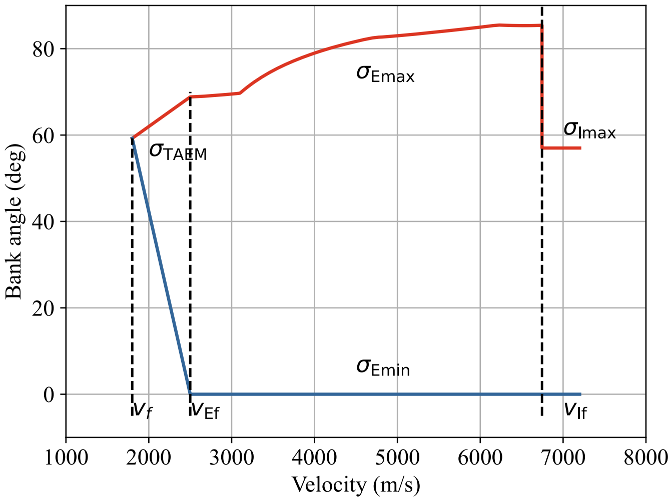


Figure 2. Compound bank angle corridor.

2.3.2. Control Parameterization

According to Equation (1), it is easy to obtain the derivative relationship between range and speed as

$$\left|\frac{dS_e}{dv}\right| = \left|\frac{v\cos\theta/r}{-D/m_0 - g\sin\theta}\right|. \tag{20}$$

It can be known from QEGC that $r \approx R_0$, $\theta \approx 0$. Bringing Equation (5) into Equation (20), the derivative relationship between range and speed becomes

$$\left|\frac{dS_e}{dv}\right| = \left|\frac{m_0}{-1/2\rho v C_D S_{ref} R_0}\right|. \tag{21}$$

Here, $|dS_e/dv|$ decreases exponentially with an increase in height h. Combined with QEGC, h is inversely proportional to σ_{TAEM}. Finally, it can be concluded that the smaller the bank angle, the smaller the $|dS_e/dv|$, and the stronger the gliding capability of the vehicle. Therefore, the bank angle profile $\sigma_{design}(v)$ is weighted by the upper and lower bounds of the compound corridor, and its expression is

$$\sigma_{design}(v) = \omega \cdot \sigma_{\min}(v) + (1-\omega) \cdot \sigma_{\max}(v), \tag{22}$$

where ω is the weighting coefficient, which can adjust the height of the entire bank angle profile to obtain different downrange, and the downrange increases monotonically with ω.

Traditionally, given the initial states $\mathbf{x}_{state0}$ and weighting coefficient ω, a complete trajectory is obtained by integrating dynamic differential equations:

$$\mathbf{x}_{state} = \mathbf{x}_{state0} + \int_{t_0}^{t_f} \dot{\mathbf{x}}_{state} \mathrm{d}t. \tag{23}$$

where $\dot{\mathbf{x}}_{state}$ is calculated by Equation (1). Generally, the fourth-order Runge–Kutta is selected as the trajectory integrator.

Different weighting coefficients ω can integrate different feasible trajectories. As shown in Figure 3, when ω traverses $[0,1]$, all possible trajectories at the current state of the HGV can be generated, and the corresponding terminal points compose the footprint. The generation of a trajectory is inseparable from the long-term integration, and the long flight time of the HGVs and the large number of trajectories required to form the footprint lead to the exponential increase for computational burden, which makes it difficult to meet the real-time requirements of online footprint generation. Therefore, in the next section, we try to use DNNs to replace the traditional integrator to solve the problem in which the traditional footprint generation methods cannot meet the real-time performance.

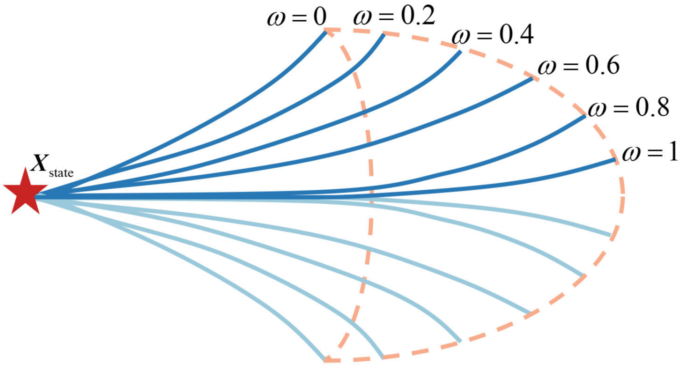


Figure 3. The footprint formed by trajectories with different weighting coefficients.

As such, the reentry problem is formulated, and a baseline predictor–corrector reentry guidance is developed based on a compound bank angle corridor, which can be used to generate feasible reentry trajectories. In the remainder of this study, we will focus on the following two study purposes: online footprint prediction and online attack intention inference. The details are given in the following two sections.

3. DNN-Based Footprint Prediction

This section focuses on the first purpose of this study, that is, online footprint prediction. Specifically, first, a DNN is trained to approximate the nonlinear functional relationship between flight states and ranges. Second, the trained DNN is leveraged to achieve the real-time performance and accuracy for range prediction. Third, on the basis of the trained DNN, an online footprint prediction algorithm is developed.

3.1. DNN Development for Range Prediction

This subsection focuses on the generation of the dataset and the construction of the DNN. First, based on Section 2.3.2, reentry trajectories of different ranges can be obtained offline. The downrange and crossrange of the reentry flight is uniquely determined by the three flight states h, v, θ, and the weighting coefficient ω of the compound bank corridor. Different initial states and weighting coefficients are randomly selected to generate 10,000 trajectories, and 100 sample points are randomly selected from each trajectory. Specifically, the selection rules are shown in Table 1. Finally, a total of 1 million data samples are obtained, as shown in Figure 4, where the input are the flight states h, v, θ, and the weighting coefficients ω, and the output is the downrange and crossrange. The dataset is further divided into three sub-datasets, namely the training set, validation set, and test set, according to the ratio of 0.8:0.1:0.1. On this basis, Scikit-learn further normalizes the input and output of the dataset.

Table 1. Initial state space.

State	Values	Distribution
$h(t_0)$	100 ± 20 km	Uniform
$v(t_0)$	7200 ± 200 m/s	Uniform
$\theta(t_0)$	-1 ± 1 deg	Uniform
ω	$[0, 1]$	Uniform

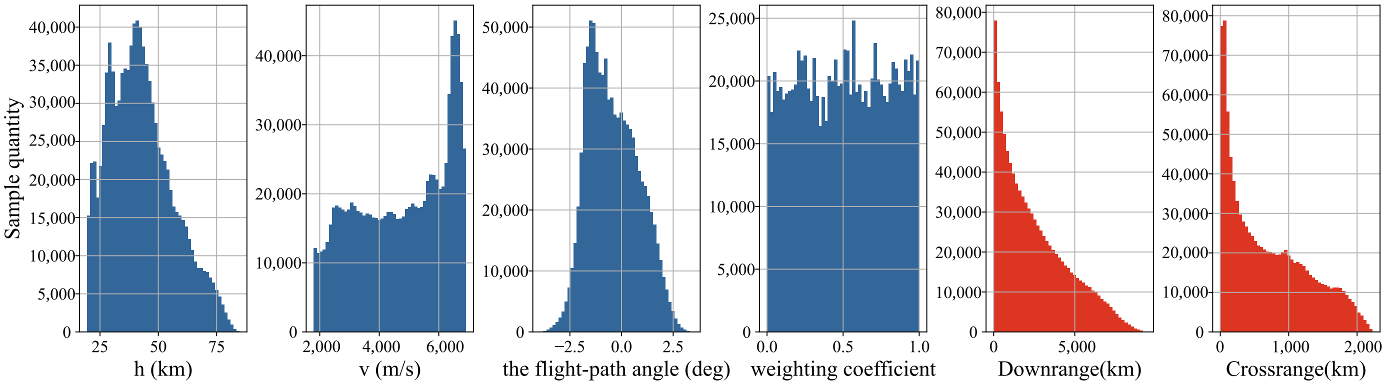


Figure 4. Distribution histograph of input and output of the dataset.

Second, a fully connected feedforward neural network is utilized to approximate the ranges. The input of the network is $\mathbf{x} = [h,\ v,\ \theta]$ and ω, and the prediction results of the network is the downrange Net_{dpre} and the crossrange Net_{cpre}. The design of the network refers to the optimized network design in [20,22,26]. The network is set up with 6 hidden layers and 128 neural units per layer. The activation functions of the hidden layers adopt Tanh $[-1, 1]$; the activation function of the output layer adopts ReLU $[0, +\infty]$. The Adam algorithm is used to adjust the network weights to minimize the mean square error. The initial learning rate is 0.001, and the exponential decay coefficient is 10^{-6}.

3.2. Accuracy and Rapidity Analysis

In this subsection, learning results and figures are used to illustrate the real-time performance and accuracy of the DNN approximation for ranges. We use the Pearson product–moment correlation coefficient to quantitatively evaluate the learning effect of the DNN. Table 2 gives the statistical results of the approximation error. At the same time, Figure 5 shows the error distribution histogram. Considering the very large range of the reentry, this error is reasonable and acceptable.

Table 2. Error statistics of the DNN-based range prediction.

	Net_{dpre}	Net_{cpre}
Mean error	0.508 km	0.196 km
Mean square error	0.458 km	0.152 km
Maximum error	17.01 km	9.642 km
Correlation coefficient	0.9999906	0.9999728

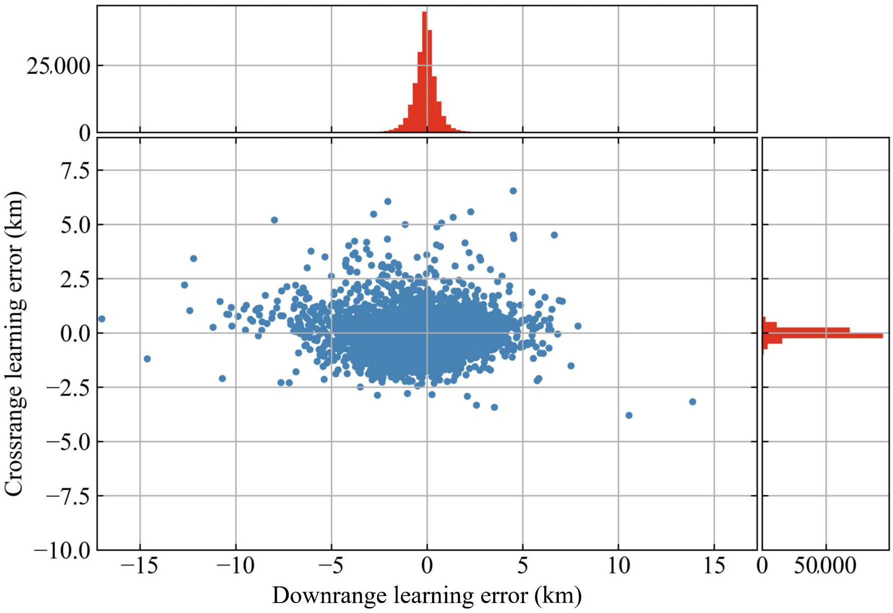


Figure 5. Error histogram of the DNN-based range prediction.

Table 3 gives the computational time it takes to predict ranges for different times by DNN and integrator. We can see that a single prediction by integrator takes 50 times as long as a single prediction by DNN. As the number of predictions grows, the rapidity of DNN becomes more and more significant, and the average time spent on a prediction by DNN becomes less and less. DNN not only has good real-time performance for a single prediction, but also is more suitable for multiple predictions. As footprint generation and intent inference require multiple range predictions, DNN is a good way to meet real-time requirements.

Table 3. The time consumption of range prediction by trajectory integrator and DNN.

Method	**DNN**			**Integrator**	
The Number of Times of Range Prediction	**1**	**10**	**10,000**	**1**	**10**
Total Time Consumption	0.041 s	0.054 s	0.29 s	1.86 s	13.07 s
Average Time Consumption	0.041 s	5.43 ms	0.029 ms	1.86 s	1.31 s

In summary, we can conclude that the trained DNN can meet the real-time requirement and has a good fitting accuracy on the ranges. Based on the above conclusions, DNN can be utilized for online footprint prediction and intent inference.

3.3. Real-Time Footprint Prediction

In this subsection, an online footprint prediction algorithm is developed on the basis of the trained DNN. Traditionally, the footprint is generated by traversing the ranges using a baseline trajectory planner. However, the traditional planner cannot meet the real-time requirements due to the long integration time. As discussed in Section 3.2, DNN can replace the integrator to predict the ranges according to the flight states and the weighting coefficient meeting the real-time requirement and ensuring a good fitting accuracy. Therefore, in this study, DNN is utilized to generate the footprint by traversing the ranges.

Knowing state $\mathbf{x}_{in} = [h_{in}, v_{in}, \theta_{in}]$, when $\omega_i \subseteq [0,1]$, the corresponding downrange $Net_{dpre}\big|_{\omega=\omega_i}$ and crossrange $Net_{cpre}|_{\omega=\omega_i}$ are obtained. Here, $Net_{cpre}|_{\omega=\omega_i}$ only represents the maximum lateral maneuverability and does not mean that the value of the crossrange must be $Net_{cpre}|_{\omega=\omega_i}$. By the bank angle reversals, the actual crossrange can be any number between $[-Net_{cpre}|_{\omega=\omega_i}, Net_{cpre}|_{\omega=\omega_i}]$.

As shown in Figure 6, the boundaries of the footprint consist of the following four sides:

- Lower boundary: The downrange is $Net_{dpre}\big|_{\omega=0}$, and the crossrange traverses $[-Net_{cpre}|_{\omega=0}, Net_{cpre}|_{\omega=0}]$. The downrange of this edge is the minimum downrange;
- Upper boundary: The downrange is $Net_{dpre}\big|_{\omega=1}$, and the crossrange traverses $[-Net_{cpre}|_{\omega=1}, Net_{cpre}|_{\omega=1}]$. The downrange of this edge is the maximum downrange;
- Right boundary: When $\omega_i \subseteq (0,1)$, the downrange is $Net_{dpre}\big|_{\omega=\omega_i}$, and the crossrange is $Net_{cpre}|_{\omega=\omega_i}$ (Based on the heading direction of the enemy vehicle at the current moment, the lateral range of the left deviation is negative, and the right deviation is positive);
- Left boundary: When $\omega_i \subseteq (0,1)$, the downrange is $Net_{dpre}\big|_{\omega=\omega_i}$, and the crossrange is $-Net_{cpre}|_{\omega=\omega_i}$.

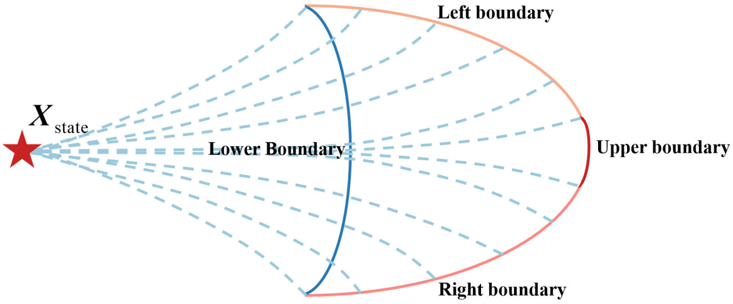


Figure 6. A footprint diagram.

Now we know the current longitude λ_{in} , latitude ϕ_{in}, heading angle ψ_{in} of the flying HGV, and the predicted range sequence of the boundary of the footprint. It is also necessary to convert the downrange sequence $Net_{dpre}\big|_{\omega=\omega_i}$ and the crossrange sequence $Net_{cpre}|_{\omega=\omega_i}$ into the longitude sequence λ_f and latitude sequence ϕ_f of the boundary of the footprint.

$$A_d = \frac{Net_{dpre}\big|_{\omega=\omega_i}}{R_0} \tag{24}$$

$$A_c = \frac{Net_{cpre}|_{\omega=\omega_i}}{R_0} \tag{25}$$

$$\psi_f = \arcsin\frac{A_c}{A_d} + \psi_{in} \tag{26}$$

$$\phi_f = \pi/2 - \arccos(\sin(\phi_{in})\cos(A_d) + \cos(\phi_{in})\sin(A_d)\cos(\psi_f)) \tag{27}$$

$$\lambda_f = \lambda_{in} + \arcsin(\sin(\psi_f)\sin(A_d)/\cos(\phi_f)) \tag{28}$$

So far, as long as the current states $\mathbf{x}_{in} = [h_{in}, \lambda_{in}, \phi_{in}, v_{in}, \theta_{in}, \psi_{in}]$ are known, the latitude and longitude of the footprint can be calculated. After the footprint is known, it can be determined which targets are within the attack zone at the current moment. One simple way of finding whether a target is inside or outside the footprint is a ray casting algorithm [27]. As shown in Figure 7, one can draw a ray from this target and count the number of points at which the ray intersects the footprint. If the number is odd, the target is inside the footprint; otherwise, it is outside the footprint.

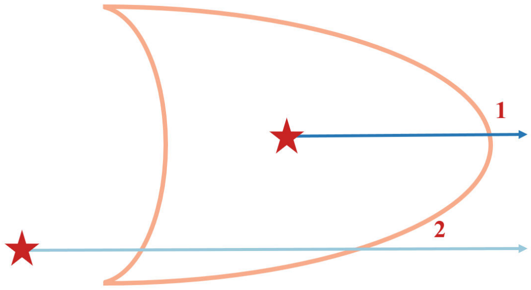


Figure 7. Ray casting algorithm.

This section focuses on the problem of HGVs footprint prediction. Fast and accurate prediction of the footprint can maximize the effectiveness of pre-deployed interceptor forces. First, we generate a large number of HGV trajectories offline and use DNN to learn the range accurately. Results show that DNN approximation for ranges has real-time performance and accuracy. Then, based on it, the footprint can be generated rapidly by predicting a series of ranges, and targets can be judged as to whether they are within the footprint. However, there are many targets in the footprint. In the following section, an online attack intention inference algorithm that calculates the target probability to be attacked in the footprint is proposed.

4. Attack Intention Inference

This section focuses on the second purpose of this study, that is, online attack intention inference. This section contains three parts. The first part introduces the criteria of intention inference. Target reachability and the historical data, including the orientation and the control strategy of the HGVs, are taken into consideration. A forgetting mechanism is proposed in the second part in case the enemy changes the target during flight. The third part describes the attack intention inference system and potential performance.

4.1. Intention Inference Criteria

Although landmarks outside the footprint can be excluded according to Section 3.3, there are many landmarks remaining in the footprint and we cannot judge which one is the attack target. During the flight of HGVs, as long as the landmarks still stay in the footprint, they can be attacked by proactive maneuvers of the vehicle. Using only the flight state at the current moment, the attack intention cannot be inferred. How to combine the current footprint prediction with the historical flight data is the key point of this section.

Through in-depth mining of the internal information hidden in the trajectories, three evaluation criteria are set. First, the control strategy of HGVs will not change sharply when the HGV aims at one specific target. Second, in order to achieve precise strikes, the HGVs' heading angle should be maintained around the target direction. Third, landmarks with higher importance are more likely to be attacked.

4.1.1. Change Detection of the Control Strategy

Due to the high speed of HGVs, a slight change in the control command will lead to a obvious change in the trajectory, and it is very likely to lead to an increase in miss distance and lose stability. Therefore, in the actual flight process, frequent and sharp changes of the control command are avoided. The bank angle shows piecewise linear characteristics when the guidance strategy is reference trajectory-based guidance (RTG) [28,29]. The bank angle is constantly corrected in small amplitudes according to the remaining downrange when the guidance strategy is NPCG [22,30].

As shown by the red line in Figure 8, if the landmark is the attack target, the weighting coefficient ω inferred from the detected historical flight state to attack this landmark is almost unchanged. As shown by the blue line in Figure 8, if the landmark is not the target, the distance between the vehicle and the landmark is not always decreasing. The weighting coefficient ω inferred from the detected historical flight state to attack this landmark changes sharply.

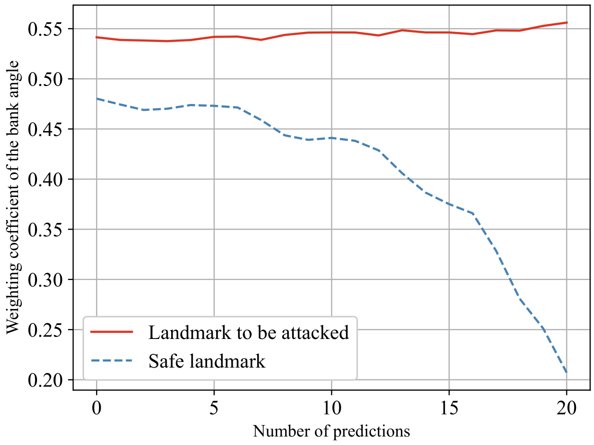
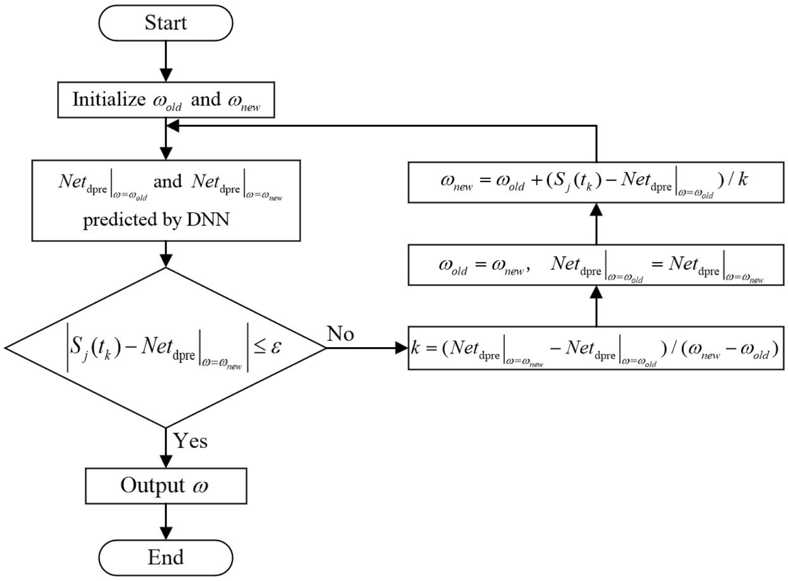


Figure 8. Change tendency of inferred ω.

When the defender does not know the enemy's attack intention and can only detect the flight states, how to infer the weighting coefficient ω when an HGV is going to attack one specific landmark will be explained below. Suppose there are m landmarks, and we need to infer whether they are likely to be attacked and which landmark is most likely to be attacked. The state $\mathbf{x}_k = [h_k, \lambda_k, \phi_k, v_k, \theta_k, \psi_k]$ of the HGV at the current time t_k and the latitude ϕ_j and longitude λ_j of the jth landmark are known. The steps of estimating the weighting coefficient $\omega_j(t_k)$ if a HGV is going to attack the jth landmark at the current moment t_k by the gradient descent method are as follows. The flowchart is shown in Figure 9.



Start
Initialize ω_{old} and ω_{new}
$Net_{\text{dpre}}\big|_{\omega=\omega_{old}}$ and $Net_{\text{dpre}}\big|_{\omega=\omega_{new}}$ predicted by DNN
$\left|S_j(t_k) - Net_{\text{dpre}}\big|_{\omega=\omega_{new}}\right| \le \varepsilon$
No
$k = (Net_{\text{dpre}}\big|_{\omega=\omega_{new}} - Net_{\text{dpre}}\big|_{\omega=\omega_{old}}) / (\omega_{new} - \omega_{old})$
$\omega_{old} = \omega_{new}$, $Net_{\text{dpre}}\big|_{\omega=\omega_{old}} = Net_{\text{dpre}}\big|_{\omega=\omega_{new}}$
$\omega_{new} = \omega_{old} + (S_j(t_k) - Net_{\text{dpre}}\big|_{\omega=\omega_{old}}) / k$
Yes
Output ω
End

Figure 9. Flowchart for estimating ω using gradient descent.

- Step1: Set $\omega_{old} = 0.5$, $\omega_{new} = 0.6$. According to $\mathbf{x}_k = [h_k, v_k, \theta_k]$, ω_{old} and ω_{new}, $Net_{dpre}\Big|_{\omega=\omega_{old}}$ and $Net_{dpre}\Big|_{\omega=\omega_{new}}$ can be predicted by DNN. Then, calculate the distance $S_j(t_k)$ between the current latitude ϕ_k and longitude λ_k of the HGV and the latitude ϕ_j and longitude λ_j of the jth landmark.
- Step2: Calculate $k = \left(Net_{dpre}\Big|_{\omega=\omega_{new}} - Net_{dpre}\Big|_{\omega=\omega_{old}} \right) / (\omega_{new} - \omega_{old})$. Then, set $\omega_{old} = \omega_{new}$, $Net_{dpre}\Big|_{\omega=\omega_{old}} = Net_{dpre}\Big|_{\omega=\omega_{new}}$.
- Step3: Calculate $\omega_{new} = \omega_{old} + \left(S_j(t_k) - Net_{dpre}\Big|_{\omega=\omega_{old}}\right)/k$. The new downrange $Net_{dpre}\Big|_{\omega=\omega_{new}}$ is obtained by DNN.
- Step4: If $\left| S_j(t_k) - Net_{dpre}\Big|_{\omega=\omega_{new}} \right| \leq \varepsilon$ (ε is the allowable deviation of the downrange), output $\omega_j(t_k) = \omega_{new}$. If $\left| S_j(t_k) - Net_{dpre}\Big|_{\omega=\omega_{new}} \right| > \varepsilon$, return to Step2.

So far, we have obtained $\omega_j(t_k)$ if the enemy HGV intends to attack the jth landmark at t_k. Similarly, we can calculate the weighting coefficient $\omega_j(t_k)(j = 1, 2, \ldots, m)$ if the vehicle intends to attack each remaining landmark at t_k. Not only that, given the current states and historical states $\mathbf{x}_i = [h_i, \lambda_i, \phi_i, v_i, \theta_i, \psi_i](i = 1, 2, \ldots, k)$ of the vehicle, we can get the sequence of weighting coefficients to attack each landmark corresponding to time $\omega_j(t_i(i = 1, 2, \ldots, k))(j = 1, 2, \ldots, m)$.

For each landmark j, there is a sequence of weighting coefficients $\omega_j(t_i(i = 1, 2, \ldots, k))$ with respect to time $t_i(i = 0, 1, \ldots, k)$. Use variance $\mathrm{var}(\omega_j(t_i(i = 1, 2, \ldots, k)))$ to characterize the change detection of the control strategy of the jth landmark. Normalize $\mathrm{var}(\omega_j(t_i(i = 1, 2, \ldots, k)))(j = 1, 2, \ldots, m)$ of each landmark to $[0, 1]$ and sort it. The smaller the variance $\mathrm{var}(\omega_j(t_i(i = 1, 2, \ldots, k)))$ is, the more likely the jth landmark is to be attacked. By continuously detecting new states of the vehicle in a new moment, the variance sequence can be updated and sorted to predict which landmark is the most likely to be attacked.

4.1.2. The Cumulative Deviation of ψ and the LOS

In order to strike one specific target, the heading angle ψ of the HGV is maintained around the target direction, and the deviation between the ψ and the line of sight (LOS) is not very large. As shown by the red line in Figure 10, if the landmark is the attack target, due to the bank angle reversals, ψ will swing left and right around the LOS. As shown by the blue line in Figure 10, if the landmark is not the target, the deviation between ψ and the LOS of the landmark is relatively large. As the flight time becomes longer, the deviation becomes larger and larger, and ψ is likely to be on one side of the LOS and will no longer swing left and right around the LOS.

When the defender does not know the enemy's attack intention and can only detect the flight states, how to characterize and rank the deviation between the ψ and LOS will be explained below. The expression $\psi(t_k)$ is the heading angle of the vehicle at the current moment t_k; $\mathrm{LOS}_j(t_k)$ is the LOS between the vehicle and the jth landmark at t_k; $\psi(t_k) - \mathrm{LOS}_j(t_k)$ represents the deviation of the heading angle of the vehicle and the LOS of the jth landmark. Given $\psi(t_i)(i = 1, 2, \ldots, k)$ and $\mathrm{LOS}_j(t_i)(i = 1, 2, \ldots, k)$ at $t_i(i = 0, 1, \ldots, k)$, the mean of the cumulative deviation $\left|\mathrm{mean}(\psi(t_i) - \mathrm{LOS}_j(t_i)(i = 1, 2, \ldots, k))\right|$ between the heading angle of the vehicle and the LOS of the jth landmark at all times can be calculated. For each landmark, the mean sequence $\left|\mathrm{mean}(\psi(t_i) - \mathrm{LOS}_j(t_i)(i = 1, 2, \ldots, k))\right|$ $(j = 1, 2, \ldots, m)$ can be calculated, normalized to $[0, 1]$, and sorted. The smaller the mean $\left|\mathrm{mean}(\psi(t_i) - \mathrm{LOS}_j(t_i)(i = 1, 2, \ldots, k))\right|$, the more likely the jth landmark is to be attacked. Continuously detecting the states of the vehicle in a new moment can update and sort the mean sequence of the deviation of the ψ and the LOS.

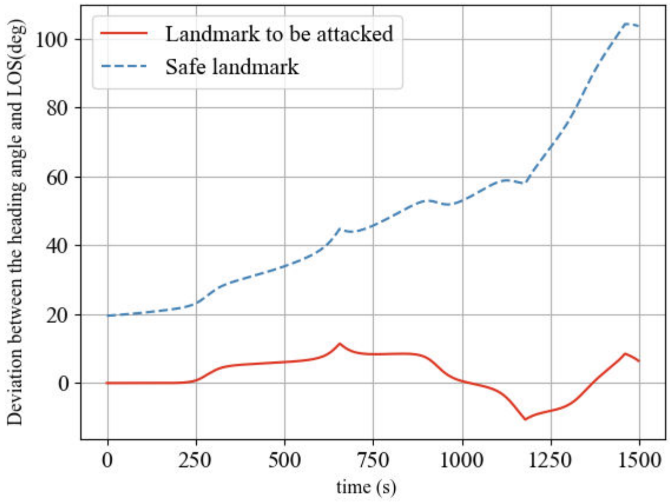


Figure 10. Deviation of the heading angle and the LOS.

4.1.3. The Importance of Landmarks

Among the multitudinous input landmarks, it is impossible that all the landmarks have the same degree of importance. In actual wars, attackers are more inclined to attack large cities, economically developed areas, and areas with high population density. Therefore, the importance of the landmarks needs to be scored on a $[0,1]$ scale. The smaller the importance degree weight I_j, the more important the jth landmark is. Because some of the less important landmarks are close to the important landmarks, $\mathrm{var}(\omega(t_i(i=1,2,\ldots,k)))$ and $|\mathrm{mean}(\psi_i - LOS_i)(i=1,2,\ldots,k)|$ of the less important landmarks are numerically close to those of the important landmarks, which may lead the system to judge that less important landmarks are also likely to be attacked. The importance weights I solve the problem of less important landmarks interfering with the attack intention inference system.

4.2. Forgetting Mechanism

With the development of reentry guidance, the maneuvering form of HGVs has become more and more complex. In actual flight, in order to bypass the no-fly zone and avoid being detected by radar, HGVs may conduct proactive maneuvers or change the attack target during flight. In order to instantly identify an HGV's changing target and re-predict its new target, this subsection proposes a forgetting mechanism.

Exponentially weighted moving average (EWMA) is a variable-weight mean calculation method, which has been widely used in machine learning and technical analysis of financial data [31]. EWMA is designed as such that older data are given lower weights. The weights fall exponentially as the data get older. EWMA is used because recent input data have a greater impact on the average value, and it can better reflect the recent data information.

Let $x_i(i=1,2,\ldots,t)$ be the dynamic input data, then the average μ_t at time t can be expressed as

$$\mu_t = \beta\mu_{t-1} + (1-\beta)x_t = (1-\beta)(x_t + \beta x_{t-1} + \beta^2 x_{t-2} + \ldots + \beta^{t-1}x_1), \tag{29}$$

where β is the decay rate, and the value is between $[0,1]$. The smaller the β, the more the mean is affected by the recent input data. However, μ_t is very different from the mean value due to too little input data at the beginning. Therefore, the deviation correction is performed on Equation (29):

$$\mu_t = (1-\beta)/(1-\beta^t)(x_t + \beta x_{t-1} + \beta^2 x_{t-2} + \ldots + \beta^{t-1}x_1). \tag{30}$$

So far, we have obtained the average value μ_t of the dynamic input data represented by EWMA. In [31], exponentially weighted moving variance (EWMV) is proposed to describe the variance of variable weights. The formula of EWMV is expressed as

$$\sigma_t^2 = \beta\sigma_{t-1}^2 + \beta(1-\beta)(x_t - \mu_{t-1})^2. \tag{31}$$

Similarly, EWMV is designed to be more affected by the recent input data, so the variance of the older data is given lower weights that fall exponentially as the data get older. On the basis of Equations (30) and (31), EWMV is improved. We replace the mean μ_t in the variance Equation (31) with the latest data x_t from the dynamic input data. The improved EWMV is expressed as

$$\sigma_t{}^2 = (1-\beta)/(1-\beta^t)[(x_t - x_t)^2 + \beta(x_{t-1} - x_t)^2 + \ldots + \beta^{t-1}(x_1 - x_t)^2]. \tag{32}$$

In summary, we design the mean and variance with forgetting properties. We substitute EWMV for the variance in Section 4.1.1 and EWMA for the mean in Section 4.1.2. In this way, the change detection of the control strategy and the cumulative deviation between the ψ and the LOS is more affected by the recent flight states. This is conducive to predicting the enemy's attack intention at every moment without knowing whether and when to change the attack target during flight.

4.3. Attack Intention Inference System and Algorithm

This section will briefly describe the algorithm of the attack intention inference system and its potential performance. Referring to the pseudocode below, we will describe the Algorithm 1 of the attack intention inference system in detail.

Algorithm 1 Attack Intention Inference Algorithm

```
Data: new flight states of the enemy HGV $\mathbf{x}_k = [h_k, \lambda_k, \phi_k, v_k, \theta_k, \psi_k]$ at time $t_k$,
      landmarks, importance of landmarks $I$
Result: ranked $P_k$
while detect the new states $\mathbf{x}_k$ at $t_k$ do
    predict $footprint(t_k)$ by DNN ;
    for each landmark $j$ in landmarks do
        if landmark $j$ within $footprint(t_k)$ then
            Calculate $\omega_j(t_k)$ by DNN $\rightarrow \omega_j[k]$;
            Calculate $\psi(t_k) - \text{LOS}_j(t_k) \rightarrow \Delta\psi_j[k]$;
            $\text{Var}(\omega_j) \leftarrow \text{EWMV}(\omega_j)$;
            $\text{Mean}(\Delta\psi_j) \leftarrow \text{EWMA}(\Delta\psi_j)$;
            $P(t_k)[j] = \alpha_1\text{Var}(\omega_j) + \alpha_2|\text{Mean}(\Delta\psi_j)| + \alpha_3 I_j$;
    rank $P(t_k)$;
    return ranked $P(t_k)$;
```

First, we predict the current footprint by DNN approximation for ranges and determine whether the input landmarks are within the footprint according to Section 3.3. If the landmark is within the footprint, the probability of it being attacked will be estimated. Combined with the forgetting mechanism in Section 4.2, the normalized var$(\omega(t_i)$ $(i = 1, 2, \ldots, k))$ and$|$mean$(\psi(t_i) - \text{LOS}(t_i)(i = 1, 2, \ldots, k))|$ of each landmark can be calculated according to Sections 4.1.1 and 4.1.2. Weighting them with landmark importance weights I yields the probability of each landmark of being attacked,

$$P = \alpha_1\text{var}(\omega(t)) + \alpha_2|\text{mean}(\psi(t) - \text{LOS}_j(t))| + \alpha_3 I, \tag{33}$$

where $\alpha_1, \alpha_2, \alpha_3$ are the weights of intention inference criteria, satisfying $\alpha_1 + \alpha_2 + \alpha_3 = 1$; $\alpha_1, \alpha_2, \alpha_3$ can be set and adjusted according to actual operational needs and the importance of each criteria. Then, we calculate and sort P of each landmark. The smaller the P, the greater the probability of the landmark to be attacked. The defending side continuously detects the current states of the flying HGV, updates and sorts in real-time P to infer the intention of the HGV. As time progresses, the accuracy of the intention inference system greatly improves.

This proposed algorithm has the following potential advantages.

1. Ability to quickly and accurately predict the footprint: Since the DNN replaces the traditional integrator to approximate the ranges, the footprint prediction meets the real-time requirements while ensuring high accuracy;
2. Ability to infer the intention of enemy HGVs: The underlying logic of the attack intention inference system in this study is that the defender infers from the perspective of the attacker. First, the control strategy of HGVs will not change sharply when the HGV aims at one specific target. Second, in order to achieve precise strikes, the HGVs' heading angle should be maintained around the target direction. Third, attackers are more willing to attack important landmarks. It is precisely because these criteria are condensed from the rules discovered from the trajectory planning of the attacker side that the defending side can use these criteria to infer the enemy's intention;
3. Ability to identify enemy HGVs changing attack intention online and re-predict new targets: Mean and variance with forgetting properties makes the system more affected by the recent flight states. Therefore, the forgetting mechanism can gradually forget the early intention and predict new target based on the data of the recent period;
4. Good real-time performance: Both the footprint prediction and the change detection of the control strategy avoid long-term trajectory integration, and DNN greatly improves the computational efficiency.

The next section uses simulation examples to verify the aforementioned performance of the algorithm.

5. Simulations and Results

This section focuses on the performance verification of the proposed algorithm. Specifically, three experiments are conducted. The first experiment is to verify the accuracy of the footprint prediction algorithm. The second experiment verifies the effectiveness of the intention inference system. The third experiment verifies that the system is suitable for inferring the intention of the enemy to change the target during the flight. The forth subsection discusses the advantages and disadvantages of the algorithm. The common areo vehicle (CAV) designed by Boeing Company in 1998 has two configurations. In this study, the vehicle for simulation is the CAV-L with lower lift-to-drag ratio, and its aerodynamic data and overall design parameters are shown in [32].

5.1. Evaluation of Footprint Prediction

First, the accuracy and real-time performance of the DNN-based footprint predictions are evaluated by comparing it with the traditional integrator. Given the flight states of an enemy CAV, the results of footprint prediction using DNN and traditional integrator are shown in Figure 11. It can be seen that the footprint predicted by the two methods almost overlap. Table 4 shows the comparison of the coordinates of the following five typical points calculated by DNN and integrator. Point A has the largest downrange and crossrange of 0; B has the largest downrange and the corresponding maximum lateral maneuverability; C has the largest crossrange; D has the minimum downrange and the corresponding maximum lateral maneuverability; and E has the minimum downrange and crossrange of 0. It can be seen that the deviation between the coordinates predicted by the two methods is very small. The time consumption of one footprint prediction by DNN is 0.85 s, and at the same time, the time consumption of one footprint prediction by integrator is 52.71 s. Consequently, DNN-based footprint prediction is accurate and greatly improves real-time performance.

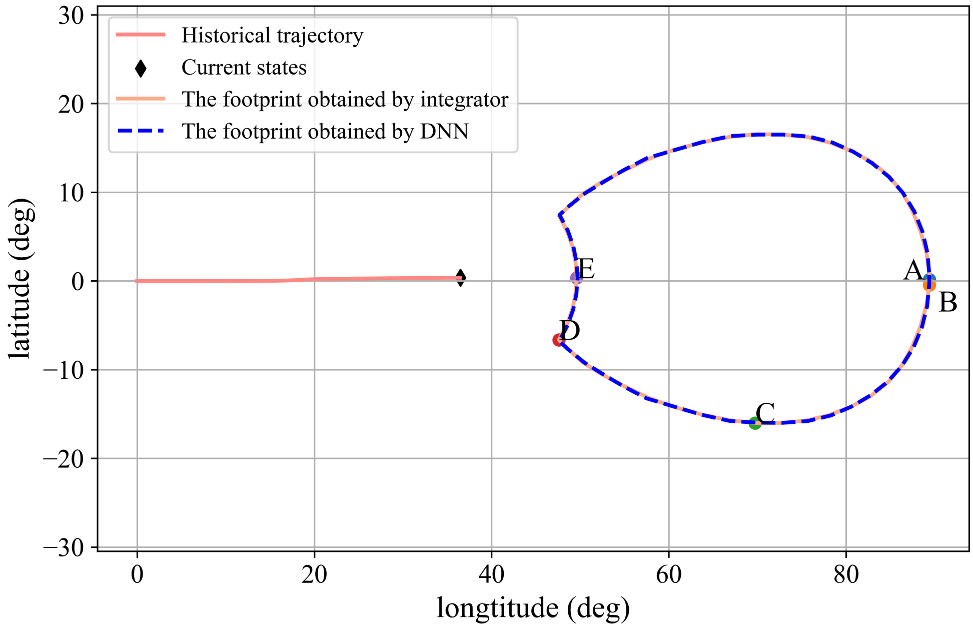


Figure 11. Comparison of footprint predicted by integrator and DNN.

Table 4. Comparison of the footprint prediction by integrator and DNN.

	Integrator	**DNN**	**Deviation**
Coordinates of A (deg)	$(89.397, 0.160)$	$(89.381, 0.160)$	$(1.6 \times 10^{-2}, -9.2 \times 10^{-5})$
Coordinates of B (deg)	$(89.391, -0.459)$	$(89.376, -0.358)$	$(1.5 \times 10^{-2}, -0.101)$
Coordinates of C (deg)	$(69.712, -16.017)$	$(69.849, -15.973)$	$(-0.137, -4.4 \times 10^{-2})$
Coordinates of D (deg)	$(47.587, -6.653)$	$(47.632, -6.772)$	$(-4.5 \times 10^{-2}, 0.119)$
Coordinates of E (deg)	$(49.592, 0.330)$	$(49.693, 0.330)$	$(-0.101, 2.6 \times 10^{-4})$

5.2. Typical Intention Inference Simulation

In this section, we turn our attention to demonstrating the effectiveness of the typical intention inference. Part of the information of the flying CAV is shown in Table 5. Of course, this information is unknown to the defender, and the defender can only detect the states of the CAV during the flight. Twenty-five landmarks B∼Z with longitude between [90° W, 140° W], latitude between [20° S, 60° S], and the importance weights I between $[0, 1]$ are generated by random numbers. We set a landmark A with longitude of 100° W, latitude of 40° S, and importance I of 0. We set the weight of each intention inference criterion to $\alpha_1 = 0.8$, $\alpha_2 = 0.1$, $\alpha_3 = 0.1$, respectively. The probability of attacking these 26 landmarks is predicted and ranked in real time. The first jump of CAV was detected at 282 s. Due to the complex dynamic characteristics of the initial descent phase of HGV, intention inference was not performed in the initial glide segment. The first prediction is at 285 s. The prediction interval is 5 s.

Table 5. The information of the incoming CAV (unknown to the defender).

Initial Longitude	**Initial Latitude**	**Longitude of the Target**	**Latitude of the Target**	**Guidance Law**
180°	0°	100° W	40° S	NPCG

The result is shown in the Figure 12, which contains the following information.

- Historical trajectory of the CAV;
- Footprint at the current state;
- Landmarks that may be attacked within the footprint;
- Temporarily safe landmarks outside the footprint;
- Landmarks within the footprint are ranked by probability of being attacked.

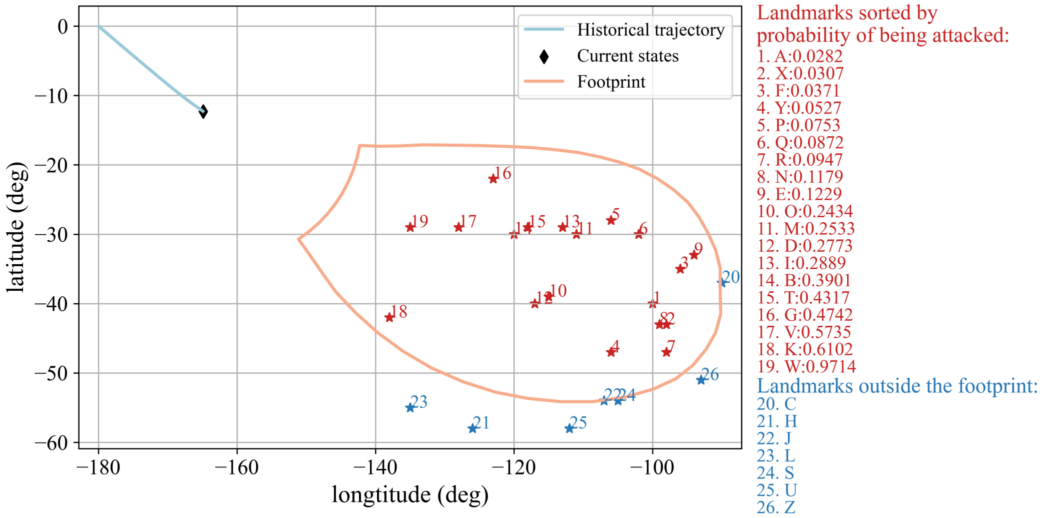


Figure 12. Footprint and intention inference at 305 s.

After five predictions (at 305 s), it is predicted that the target of the incoming CAV is A, as shown in Figure 12. The reason for needing five predictions is that the change detection of the control strategy and the cumulative deviation between the ψ and the LOS are characterized by the variance and mean in the intention inference system, requiring >2 sample data to calculate variance and mean. Predicting the correct attack intention within 25 s shows that the system has good foresight.

The prediction at 800 s is shown in Figure 13. It can be seen that the footprint becomes smaller and the number of landmarks that may be attacked decreases. The landmark that is most likely to be attacked is still A, which shows the intention inference system has good stability. As shown in Figure 14, the footprint is shrunk to the range containing only A at 1195 s. Consequently, the DNN-based intention inference system has good performance of effectiveness, perspective, and stability.

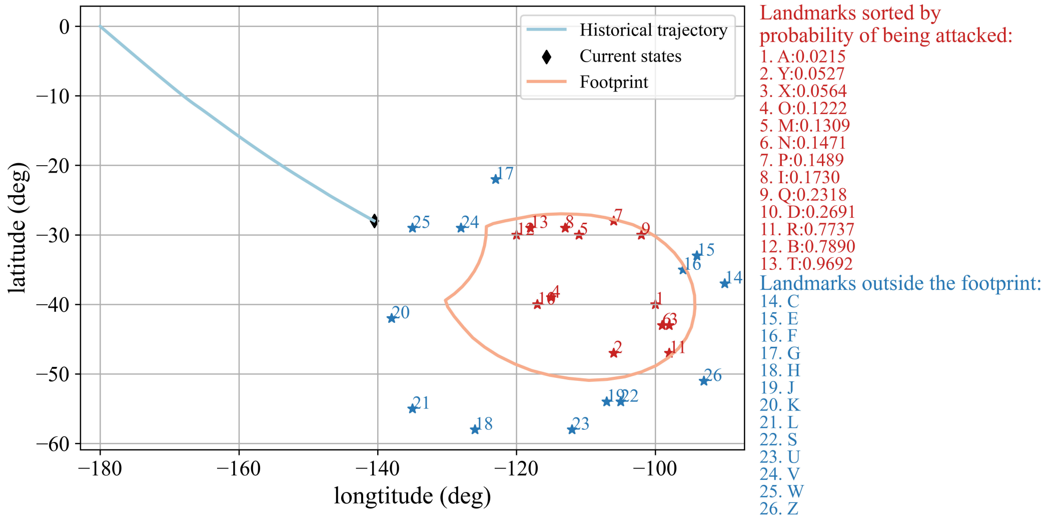


Figure 13. Footprint and intention inference at 800 s.

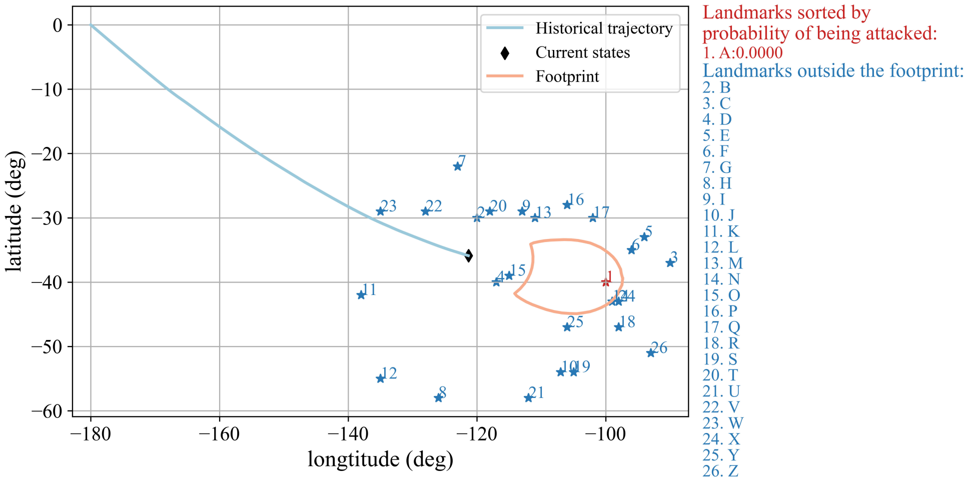


Figure 14. Footprint and intention inference at 1195 s.

5.3. Evaluation of Forgetting Mechanism

This section verifies the effectiveness of the forgetting mechanism by predicting the intention of a CAV that changes its attack target during the flight. Part of the information of the flying CAV is shown in Table 6. The vehicle originally intended to attack Landmark A at 100° W, 40° S, but changed its attack intention at 500 s to attack Landmark B at 120° W, 30° S. Of course, the above information and whether and when to change the attack target is unknown to the defender. Twenty-four landmarks C∼Z with longitude between [90° W, 140° W], latitude between [20° S, 60° S], and the importance weights I between $[0, 1]$ are generated by random numbers. We set a landmark A with longitude of 100° W, latitude of 40° S, and importance I of 0, and set a landmark B with longitude of 120° W, latitude of 30° S, and importance I of 0. We set the weight of each intention inference criterion to $\alpha_1 = 0.85$, $\alpha_2 = 0.05$, $\alpha_3 = 0.1$, respectively. The decay rate of forgetting mechanism is set to 0.7. The probability of attacking these 26 landmarks is predicted and ranked in real time. The first prediction is at 290 s. The prediction interval is 10 s.

Table 6. The flying CAV information (unknown to the defender).

Initial Coordinates of the CAV	Coordinates of the Original Target	Time When the Target Is Changed	Coordinates of the Changed Target	Guidance Law
(180°, 0°)	(100° W, 40° S)	500 s	(20° W, 30° S)	RTG

For ease of observation, the light blue dotted line in the graph represents the original trajectory if the CAV has not changed its attack target during the flight.

After four predictions (at 320 s), it is predicted that the target of the CAV is A, as shown in Figure 15. During 320∼500 s, results show that the CAV intended to attack A.

As shown in Figure 16, the prediction results begin to change at 510 s, which means the system quickly recognizes that the enemy is maneuvering to change its attack target. The probability P_B of B became smaller, and the rank moved forward; P_A of A increased, and the rank went backward. This shows that the system gradually reduces the impact of the initial trajectory on the predicted results through the forgetting mechanism and gradually forgets that the CAV is going to attack landmark A at the beginning. At 520 s, it is predicted that the CAV changes to attack landmark B, which means that the forgetting mechanism of this system is responsive, forward-looking, and accurate. The prediction results fluctuate with small amplitude between 560 s and ∼580 s and stabilize after 590 s, showing that B is most likely to be attacked, as shown in Figure 17. Consequently, the forgetting mechanism can quickly identify that the enemy has changed its intention and quickly re-predict the new attack target correctly. The prediction results are accurate and stable, and fewer predictions are required to re-predict the new target after maneuvering.

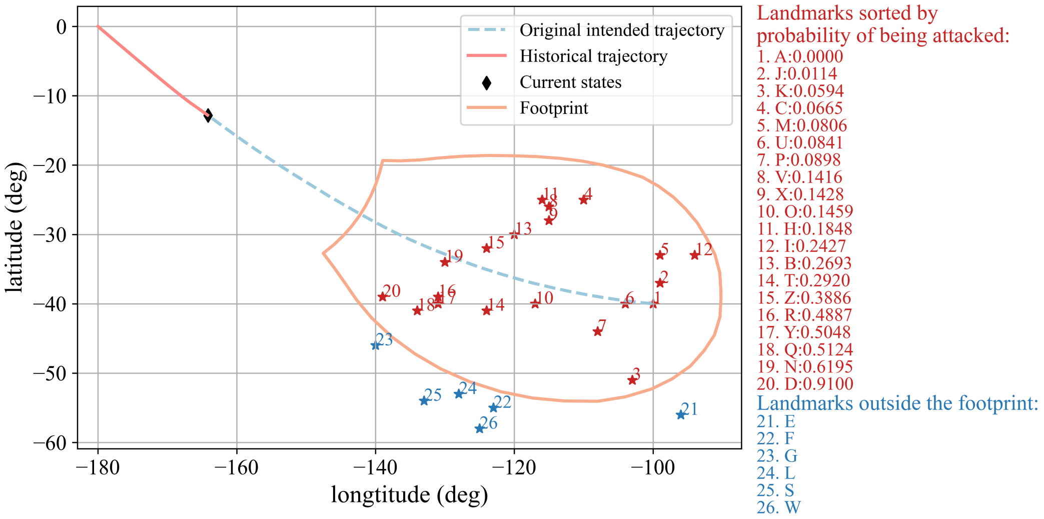


Figure 15. Footprint and intention inference at 320 s.

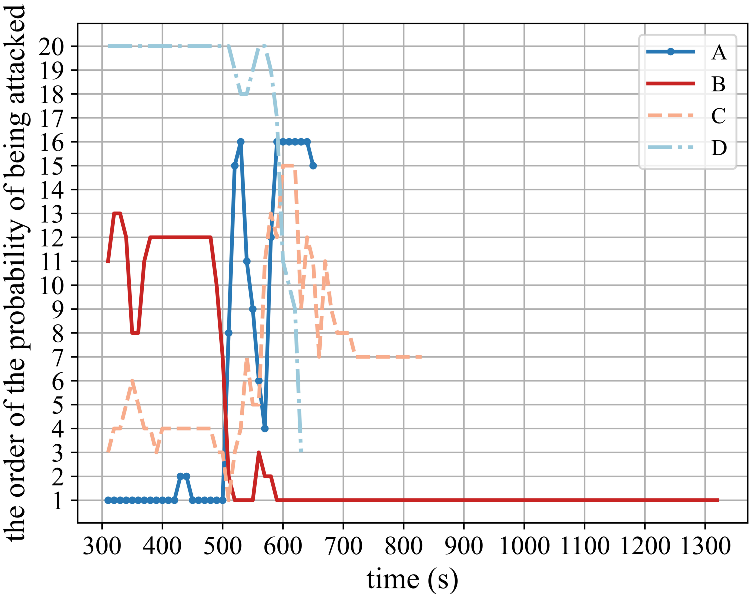


Figure 16. Ranking of probability of being attacked over time.

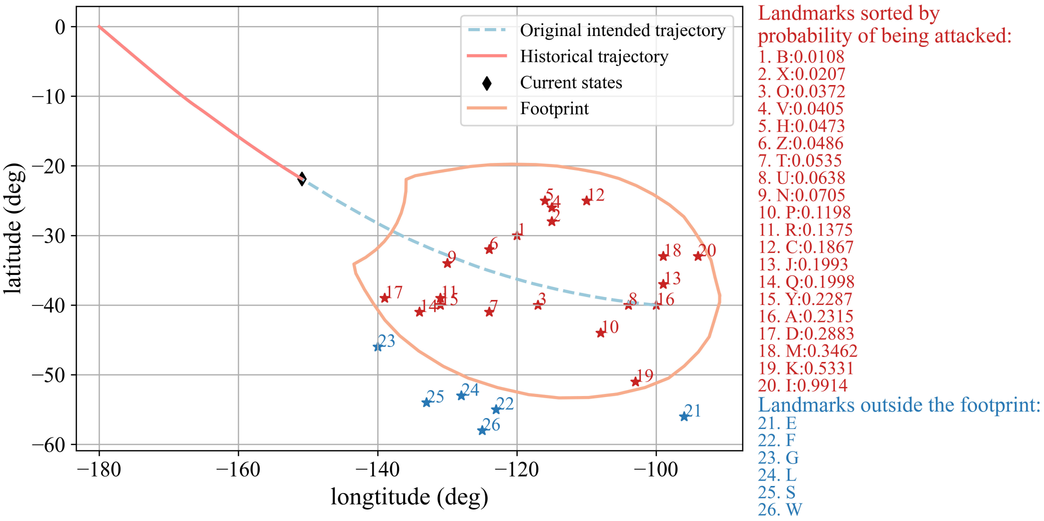


Figure 17. Footprint and intention inference at 590 s.

5.4. Advantages and Disadvantages

Combined with the above three typical simulations, the advantages of footprint prediction and intention inference system are summarized below.

1. The system has good real-time performance.
   A maximum of 4.74 s is required for one footprint prediction and intention inference (predictions are made for 26 landmarks) due to the substitution of DNN for trajectory integration. Therefore, as long as each prediction period is more than 5 s, it can reconcile the requirements for real-time performance. Table 7 exhibits the comparison of the time consumption by trajectory integrator and DNN. We can see from the Table 7 that a high-precision trajectory integration is time-consuming due to the long flight time. For this reason, it is difficult to meet the demand of real-time performance for footprint prediction and intention inference. In contrast, the DNN shows obvious improvement of the real-time performance while ensuring an acceptable prediction accuracy.
2. This system can infer the intention of HGVs that change attack target during flight.
   This system adopts EWMA and improves EWMV. Mean and variance with forgetting properties makes the system more affected by the recent flight states. Therefore, the forgetting mechanism can infer in a timely manner that the flying HGV has maneuvered and predict its new attack target.
3. The system is forward-looking.
   The first few predictions after the initial descent segment can basically determine which landmark is most likely to be attacked. The flight time of a reentry is about 30 min, and the flight time of the initial descent segment is about five minutes. Forty seconds after the initial descent segment, the most likely landmark to be attacked can be predicted, leaving as much time as possible for the defender to intercept the enemy. In addition, when the HGV changes its attack target during its reentry, the system can also analyze that the HGV has changed its attack intention and predict the new target within 20 s.
4. The DNN-based footprint prediction can not only provide decision-making reference for the defender, but also be utilized for the attacker to generate the current footprint of the own HGV.
   The current literature on footprint generation of HGVs is only from the perspective of the attacker. The footprint of HGVs provides critical information for trajectory planning, such as providing guidance for entering into the TAEM. It takes an average time of 13.42 s to generate a footprint in [13]. Compared with the footprint generation based on the Gaussian pseudospectral method, the calculation speed of [13] has been greatly improved, but it still has room for improvement. If the DNN is utilized to replace the traditional integrator to generate the footprint from the perspective of the attacker, the calculation time of each generation can be less than 1 s, which greatly improves the real-time performance of battlefield mission planning. In addition, since the model, aerodynamic parameters, guidance law, and constraints are completely known, the accuracy of generating the footprint of the friendly HGV will be greatly improved compared with the accuracy of predicting the footprint of the enemy in this study.

Table 7. The comparison of the time consumption by trajectory integrator and DNN.

	One Range Prediction		One Footprint Prediction		One Intention Inference	
	Integrator	**DNN**	**Integrator**	**DNN**	**Integrator**	**DNN**
Time consumption	1.53 s	0.6 ms	52.71 s	0.85 s	189.95 s	3.66 s

The system still has the following shortcomings that need to be improved:

1. This system can only infer the probability of attacking the set landmarks. If the set landmarks do not include the enemy's real attack target, the system may only speculate that a set landmark close to the real target is most likely to be attacked.
2. Because the research object of the intention inference in this study is non-cooperative vehicles, the overall design parameters and aerodynamic parameters of them are not completely known, and many of them are estimated based on reverse engineering estimation methods [33]. Therefore, the ranges approximated by DNN in this study are not accurate when the enemy model deviates greatly from the hypothetical model in this study, and the predicted footprint will also not be so accurate. In particular, landmarks near the boundary of the footprint are also likely to be attacked.
   In the future work, we will introduce parameter identification into the footprint prediction and intention inference system to improve the inaccuracy of range prediction and footprint prediction. One solution is to identify lift, drag, aircraft mass, reference area, and bank angle through aerodynamic parameter identification, as suggested in [15]. Another solution is to refer to the idea in [34], that is, to identify the disturbance of the standard aerodynamic data through Kalman filtering. In future work, the aerodynamic parameters will be identified according to the trajectory information of the flying HGVs in real time, and the aerodynamic model fitted by the existing public information or public literature will be continuously corrected. Using the revised aerodynamic model, DNNs can predict more accurate ranges, footprint, and intention in real time.

6. Conclusions

In this paper, a DNN-based footprint prediction and intention inference of HGVs is proposed. First, a baseline multi-constrained entry guidance algorithm is developed based on a compound bank angle corridor, and then a dataset containing enough trajectories for the following DNN learning is generated offline by traversing different initial states and control commands. Second, DNNs replace traditional integrator to approximate the relationship between the flight state/command and the ranges. On this basis, an online footprint prediction algorithm is developed by traversing the ranges meeting the real-time requirements and ensuring high accuracy. Third, the intention inference system performs online intelligent prediction of the target probability to be attacked. Target reachability, importance, and historical data including the orientation and the control strategy of the HGVs are taken into consideration. A forgetting mechanism is proposed to help the intention inference algorithm to be effective when the HGVs change their attack target during the flight. On these bases, DNN-based footprint prediction and intention inference of HGVs is proposed.

Simulations are given to substantiate the effectiveness and the real-time capability of the proposed techniques. The results show that the calculation time of each footprint generation and intention inference is about 4.51 s, reconciling the requirements for real-time performance while ensuring high accuracy. The system can infer the final attack target correctly in the early stage of reentry flight and can instantly identify the enemy's change of attack intention and re-predict the new attack target.

Author Contributions: Conceptualization, L.C., C.D. and J.X.; methodology, L.C. and J.X.; software, J.X.; validation, C.D.; investigation, L.C. and J.X.; resources, C.D. and L.C.; data curation, C.D. and L.C.; writing—original draft preparation, L.C. and J.X.; writing—review and editing, L.C. and J.X.; supervision, C.D. and L.C.; project administration, C.D. and L.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research is supported by the National Natural Science Foundation of China (No. 11902174).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available in Sections 2.1, 3.1 and 5.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Fahroo, F.; Doman, D.B.; Ngo, A.D. Footprint generation for reusable launch vehicles using a direct pseudospectral method. In Proceedings of the IEEE 2003 American Control Conference, Denver, CO, USA, 4–6 June 2003; Volume 3, pp. 2163–2168.
2. Fahroo, F.; Doman, D. A direct method for approach and landing trajectory reshaping with failure effect estimation. In Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit, Providence, RI, USA, 16–19 August 2004; p. 4772.
3. Tian, B.; Fan, W.; Su, R.; Zong, Q. Real-time trajectory and attitude coordination control for reusable launch vehicle in reentry phase. *IEEE Trans. Ind. Electron.* **2014**, *62*, 1639–1650. [CrossRef]
4. Benedikter, B.; Zavoli, A.; Colasurdo, G.; Pizzurro, S.; Cavallini, E. Convex optimization of launch vehicle ascent trajectory with heat-flux and splash-down constraints. *J. Spacecr. Rocket.* **2022**, *59*, 900–915. [CrossRef]
5. Benedikter, B.; Zavoli, A.; Colasurdo, G.; Pizzurro, S.; Cavallini, E. Autonomous Upper Stage Guidance Using Convex Optimization and Model Predictive Control. In Proceedings of the AIAA ASCEND Forum, Online, 16–18 November 2020; p. 4268. [CrossRef]
6. Benedikter, B.; Zavoli, A.; Colasurdo, G.; Pizzurro, S.; Cavallini, E. Autonomous Upper Stage Guidance with Robust Splash-Down Constraint. *arXiv* **2022**, arXiv:2212.06518.
7. Lu, P. Asymptotic analysis of quasi-equilibrium glide in lifting entry flight. *J. Guid. Control Dyn.* **2006**, *29*, 662–670. [CrossRef]
8. Lu, P.; Xue, S. Rapid generation of accurate entry landing footprints. *J. Guid. Control Dyn.* **2010**, *33*, 756–767. [CrossRef]
9. Li, H.; Zhang, R.; Zhaoying, L.; Zhang, R. Footprint problem with angle of attack optimization for high lifting reentry vehicle. *Chin. J. Aeronaut.* **2012**, *25*, 243–251. (In Chinese) [CrossRef]
10. Liu, X.; Shen, Z.; Lu, P. Solving the maximum-crossrange problem via successive second-order cone programming with a line search. *Aerosp. Sci. Technol.* **2015**, *47*, 10–20. [CrossRef]
11. Ngo, A.; Doman, D. Footprint determination for reusable launch vehicles experiencing control effector failures. In Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit, Monterey, CA, USA, 5–8 August 2002; p. 4775.
12. Saraf, A.; Leavitt, J.; Ferch, M.; Mease, K. Landing footprint computation for entry vehicles. In Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit, Providence, RI, USA, 16–19 August 2004; p. 4774.
13. Fu, S.; Lu, T.; Yin, J.; Xia, Q. Rapid algorithm for generating entry landing footprints satisfying the no-fly zone constraint. *Int. J. Aerosp. Eng.* **2021**, *2021*, 8827377. [CrossRef]
14. Zhang, K.; Xiong, J.; Li, F.; Fu, T. Bayesian Trajectory Prediction for a Hypersonic Gliding Reentry Vehicle Based on Intent Inference. *J. Astronaut.* **2018**, *39*, 1258–1265. (In Chinese)
15. Hu, Y.; Gao, C.; Li, J.; Jing, W.; Li, Z. Novel trajectory prediction algorithms for hypersonic gliding vehicles based on maneuver mode on-line identification and intent inference. *Meas. Sci. Technol.* **2021**, *32*, 115012. [CrossRef]
16. Luo, Y.; Tan, X.; Wang, H.; Qu, Z. Method for predicting the attack intention of hypersonic vehicles. *J. Xidian Univ.* **2019**, *46*, 113–119. (In Chinese)
17. Zhou, W.; Yao, P.Y.; Zhang, J.; Wang, X.; Wei, S. Combat Intention Recognition for Aerial Targets Based on Deep Neural Network. *Acta Aeronaut. Et Astronaut. Sin.* **2018**, *39*, 322468–322476.
18. Izzo, D.; Märtens, M.; Pan, B. A survey on artificial intelligence trends in spacecraft guidance dynamics and control. *Astrodynamics* **2019**, *3*, 287–299. [CrossRef]
19. Li, Z.; Sun, X.; Hu, C.; Liu, G.; He, B. Neural network based online predictive guidance for high lifting vehicles. *Aerosp. Sci. Technol.* **2018**, *82*, 149–160. [CrossRef]
20. Horn, J.F.; Schmidt, E.M.; Geiger, B.R.; DeAngelo, M.P. Neural network-based trajectory optimization for unmanned aerial vehicles. *J. Guid. Control Dyn.* **2012**, *35*, 548–562. [CrossRef]
21. Schierman, J.; Hull, J.; Ward, D. Adaptive guidance with trajectory reshaping for reusable launch vehicles. In Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit, Monterey, CA, USA, 5–8 August 2002; p. 4458.
22. Cheng, L.; Jiang, F.; Wang, Z.; Li, J. Multiconstrained real-time entry guidance using deep neural networks. *IEEE Trans. Aerosp. Electron. Syst.* **2020**, *57*, 325–340. [CrossRef]
23. Lu, P. Entry guidance: A unified method. *J. Guid. Control Dyn.* **2014**, *37*, 713–728. [CrossRef]
24. Wen, T.; Zeng, X.; Circi, C.; Gao, Y. Hop reachable domain on irregularly shaped asteroids. *J. Guid. Control. Dyn.* **2020**, *43*, 1269–1283. [CrossRef]
25. Yang, H.; Li, S.; Bai, X. Fast homotopy method for asteroid landing trajectory optimization using approximate initial costates. *J. Guid. Control. Dyn.* **2019**, *42*, 585–597. [CrossRef]
26. Cheng, L.; Wang, Z.; Song, Y.; Jiang, F. Real-time optimal control for irregular asteroid landings using deep neural networks. *Acta Astronaut.* **2020**, *170*, 66–79. [CrossRef]
27. Shimrat, M. Algorithm 112: Position of point relative to polygon. *Commun. ACM* **1962**, *5*, 434. [CrossRef]
28. Wang, X.; Guo, J.; Tang, S.; Qi, S.; Wang, Z. Entry trajectory planning with terminal full states constraints and multiple geographic constraints. *Aerosp. Sci. Technol.* **2019**, *84*, 620–631. [CrossRef]

29. Zhang, D.; Liu, L.; Wang, Y. On-line reentry guidance algorithm with both path and no-fly zone constraints. *Acta Astronaut.* **2015**, *117*, 243–253. [CrossRef]
30. Li, Z.; Yang, X.; Sun, X.; Liu, G.; Hu, C. Improved artificial potential field based lateral entry guidance for waypoints passage and no-fly zones avoidance. *Aerosp. Sci. Technol.* **2019**, *86*, 119–131. [CrossRef]
31. Finch, T. Incremental calculation of weighted mean and variance. *Univ. Camb.* **2009**, *4*, 41–42.
32. Phillips, T.H. A common aero vehicle (CAV) model, description, and employment guide. *Schafer Corp. AFRL AFSPC* **2003**, *27*, 1–12.
33. LI, H.; Jin, X.; LIN, P. Modeling and Analyzing of Common Aero Vehicle with Parametric Configuration. *J. Astronantics* **2011**, *32*, 2305–2311.
34. Cui, N.; Lu, B.; Fu, Y.; Zhang, X. Aerodynamic Parameter Identification of a Reentry Vehicle Based on Kalman Filter Method. *J. Chin. Inert. Technol.* **2014**, *6*, 755–758. (In Chinese)

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.