

Article

# Multi-Constrained Geometric Guidance Law with a Data-Driven Method

Xinghui Yan <sup>1</sup>, Yuzhong Tang <sup>1</sup> , Yulei Xu <sup>1</sup>, Heng Shi <sup>2,\*</sup>  and Jihong Zhu <sup>2</sup> 

<sup>1</sup> School of Power and Energy, Northwestern Polytechnical University, Xi'an 710129, China; yanxh@nwpu.edu.cn (X.Y.); tangyz@mail.nwpu.edu.cn (Y.T.)

<sup>2</sup> Department of Precision Instrument, Tsinghua University, Beijing 100084, China

\* Correspondence: shiheng@tsinghua.edu.cn

**Abstract:** A data-driven geometric guidance method is proposed for the multi-constrained guidance problem of variable-velocity unmanned aerial vehicles (UAVs). Firstly, a two-phase flight trajectory based on a log-aesthetic space curve (LASC) is designed. The impact angle is satisfied by a specified straight-line segment. The impact time is controlled by adjusting the phase switching point. Secondly, a deep neural network is trained offline to establish the mapping relationship between the initial conditions and desired trajectory parameters. Based on this mapping network, the desired flight trajectory can be generated rapidly and precisely. Finally, the pure pursuit and line-of-sight (PLOS) algorithm is employed to generate guidance commands. The numerical simulation results validate the effectiveness and superiority of the proposed method in terms of impact time and angle control under time-varying velocity.

**Keywords:** data-driven; multi-constrained guidance; impact time control; impact angle control; time-varying velocity



**Citation:** Yan, X.; Tang, Y.; Xu, Y.; Shi, H.; Zhu, J. Multi-Constrained Geometric Guidance Law with a Data-Driven Method. *Drones* **2023**, *7*, 639. <https://doi.org/10.3390/drones7100639>

Academic Editor: Sanjay Sharma

Received: 14 September 2023

Revised: 8 October 2023

Accepted: 13 October 2023

Published: 18 October 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Classical guidance laws, such as proportional navigation, do not give enough consideration to impact time and angle, which results in poor cooperative performance. For example, an impact angle constraint should be imposed for the UAVs to approach the target from a specified direction in some cases [1–4]. When multiple UAVs need to be gathered in a formation, the impact time should be coordinated [5–8]. Three performance indicators, namely miss distance, impact time, and impact angle, should be satisfied simultaneously during cluster flight. Therefore, impact time and angle control guidance (ITACG) laws have attracted extensive attention in both the academic and industrial worlds [9–18].

Scholars have conducted a series of studies based on different theories and methods, including optimal control theory, sliding mode control theory, computational geometry, and those methods based on proportional navigation. Lee et al. [9] proposed the first ITACG law based on optimal control theory, which consists of a feedback loop and an additional control command. In [10], a two-phase optimal ITACG law based on the virtual target method is proposed, which is suitable for all-around control in planar formations under admissible initial conditions and predetermined terminal constraints. The guidance problem is described as an optimal control model with discontinuities in [11], and an iterative guidance method is used to realize multi-constrained control. In [12], a new robust second-order sliding mode control law was developed by incorporating backstepping control, the line-of-sight rate shaping method, and second-order sliding mode control. Hou et al. [13] designed two different terminal sliding surfaces based on nonsingular terminal sliding mode control theory. This sliding mode control-based guidance law enables the UAV to satisfy both time and angle constraints while reaching the target. Reference [14], which combined the impact angle control guidance law based on backstepping control and the impact time control guidance law based on proportional navigation, achieved simultaneous

control of both impact angle and time. A new ITACG law with adjustable coefficients was proposed in [15]. The bias proportional guidance with impact angle constraint is extended with feedback control of impact time error. However, the UAVs' flight trajectories under the aforementioned guidance laws are implicit, making it difficult to accurately estimate the time-to-go in advance. Zhang et al. [16] proposed a trajectory planning method based on three circular arcs to control impact time and angle. An explicit two-phase flight trajectory based on second-order Bezier curves was proposed in [17], allowing for precise impact time and angle requirements. In [18], the geometric ITACG law was extended to three-dimensional space using log-aesthetic space curves (LASCs). However, the offline trajectory design methods in [16–18] cannot be computed quickly enough to achieve an optimal solution and do not adequately consider time-varying velocity.

Besides the classical theories and methods mentioned above, data-driven methods [19–21] have gained significant attention in the field of guidance law design in recent years. Data-driven methods focus on the relationships between input and output data in a system and approximate the behavior of the system without analytical modeling. After collecting large amounts of input and output data, a data-driven model can be established using machine learning techniques. In [22], a deep neural network (DNN) was trained to learn the mapping relationship between flight states and distances. Based on this DNN, a multi-constrained prediction–correction guidance algorithm was proposed. Guo et al. [23] proposed a guidance law that combines data-driven methods and proportional navigation. The data-driven methods were used to compensate for the impact time error of proportional navigation guidance. Based on this, a biased proportional navigation guidance law was designed to control the impact time. In [24], another impact time control guidance law was designed based on proportional navigation and data-driven methods, which do not rely on time-to-go information. Reference [25] designed a two-phase guidance law based on data-driven methods to control both impact time and angle.

However, the aforementioned guidance laws are developed based on the assumption of constant velocity and two-dimensional kinematics. In practical applications, the UAV's velocity varies with engine thrust and air drag in three dimensions. As a result, it is challenging to control impact time and angle precisely under time-varying velocity. This manuscript addresses the guidance problem of impact time and angle control in three dimensions. A guidance law based on data-driven methods and computational geometry is proposed. The main contributions of this manuscript are summarized as follows:

1. The proposed three-dimensional, multi-constrained guidance method develops a geometric trajectory planning framework. It uses data-driven methods to solve the trajectory parameters, taking impact time, impact angle, maximum overload, and time-varying velocity into account.
2. Different from references [9–15], the proposed guidance method overcomes the reliance on accurate time-to-go estimation, which is quite difficult under time-varying velocities. The proposed method achieves better performance in terms of impact time and angle under time-varying velocity.
3. Compared to references [16–18], the proposed guidance law adopts data-driven methods to solve trajectory parameters, which greatly speeds up the trajectory generation process. This improvement benefits the update of trajectories due to different mission requirements.

The remainder of this manuscript is organized as follows: Section 2 introduces foundational concepts, including the LASC and guidance model. In Section 3, a novel ITACG law is proposed based on geometric planning and data-driven methods. Numeric simulations and results are given in Section 4. Section 5 provides the conclusions.

## 2. Preliminaries

### 2.1. Log-Aesthetic Space Curves

LASCs are widely applied in many fields, such as industrial design [26,27] and artistic aesthetics [28,29]. Each LASC has a linear, logarithmic curvature graph (LCG) and a linear, logarithmic torsion graph (LTG), as represented by Equations (1) and (2) [30].

$$\log\left(\rho \frac{ds}{d\rho}\right) = \alpha \log \rho + c_1 \tag{1}$$

$$\log\left(\mu \frac{ds}{d\mu}\right) = \beta \log \mu + c_2 \tag{2}$$

where  $s$ ,  $\rho$ , and  $\mu$  are the curve length, radius of curvature, and radius of torsion, respectively;  $\alpha$  and  $\beta$  are the slopes of the segments in LCG and LTG, respectively;  $c_1$  and  $c_2$  are constants. Setting  $\Lambda = e^{-c_1}$  and  $\Omega = e^{-c_2}$ , we obtain

$$\frac{ds}{d\rho} = \frac{\rho^{\alpha-1}}{\Lambda} \tag{3}$$

$$\frac{ds}{d\mu} = \frac{\mu^{\beta-1}}{\Omega} \tag{4}$$

Setting  $\rho = 1$  and  $\mu = v$  at  $s = 0$ , and integrating Equations (3) and (4), results in

$$\rho = \begin{cases} e^{\Lambda s} & , \text{if } \alpha = 0 \\ (\Lambda \alpha s + 1)^{\frac{1}{\alpha}} & , \text{otherwise} \end{cases} \tag{5}$$

$$\mu = \begin{cases} e^{(\Omega s + \log v)} & , \text{if } \beta = 0 \\ (\Omega \beta s + v^\beta)^{\frac{1}{\beta}} & , \text{otherwise} \end{cases} \tag{6}$$

Hence, the curvature and torsion of a curve can be represented as

$$\kappa(s) = \begin{cases} e^{-\Lambda s} & , \text{if } \alpha = 0 \\ (\Lambda \alpha s + 1)^{-\frac{1}{\alpha}} & , \text{otherwise} \end{cases} \tag{7}$$

$$\tau(s) = \begin{cases} e^{-(\Omega s + \log v)} & , \text{if } \beta = 0 \\ (\Omega \beta s + v^\beta)^{-\frac{1}{\beta}} & , \text{otherwise} \end{cases} \tag{8}$$

Then, the LASC can be established by integrating the Frenet–Serret formula as follows [30,31]:

$$\begin{bmatrix} \frac{dT(s)}{ds} \\ \frac{dN(s)}{ds} \\ \frac{dB(s)}{ds} \end{bmatrix} = \begin{bmatrix} 0 & \kappa(s) & 0 \\ -\kappa(s) & 0 & \tau(s) \\ 0 & -\tau(s) & 0 \end{bmatrix} \begin{bmatrix} T(s) \\ N(s) \\ B(s) \end{bmatrix} \tag{9}$$

where  $T(s)$ ,  $N(s)$ , and  $B(s)$  are unit tangent vector, unit normal vector, and unit binormal vector, respectively.

### 2.2. Guidance Problem Statement

To facilitate analytical analysis, the following assumptions are made [10,14,18]: (1) The UAV and target are assumed to be mass points. (2) Compared with the UAV, the target’s velocity is low and can be considered static. (3) Since the autopilot dynamics are much faster than the UAV dynamics, the autopilot lag is neglected in guidance law design.

The guidance scenario in three-dimensional space is shown in Figure 1. The position coordinates in the inertial frame of the UAV and target are  $I(x_m, y_m, z_m)$  and  $T(x_t, y_t, z_t)$ ,

respectively. The inertial frame is denoted as  $O - X - Y - Z$ . The subscripts 0 and  $f$  indicate the initial and final moments, respectively. The relative kinematic equations are:

$$\begin{aligned} \dot{x}_m &= V \cos \theta_m \cos \psi_m \\ \dot{y}_m &= V \cos \theta_m \sin \psi_m \\ \dot{z}_m &= V \sin \theta_m \\ \dot{\theta}_m &= \frac{a_{zm}}{V} \\ \dot{\psi}_m &= \frac{a_{ym}}{V \cos \theta_m} \end{aligned} \tag{10}$$

where  $V$  is the UAV velocity.  $a_{zm}$  and  $a_{ym}$  are the corresponding pitch and yaw acceleration.  $\theta_m$  and  $\psi_m$  represent the pitch angle and the yaw angle, respectively.  $R$  is the range between  $I$  and  $T$ .

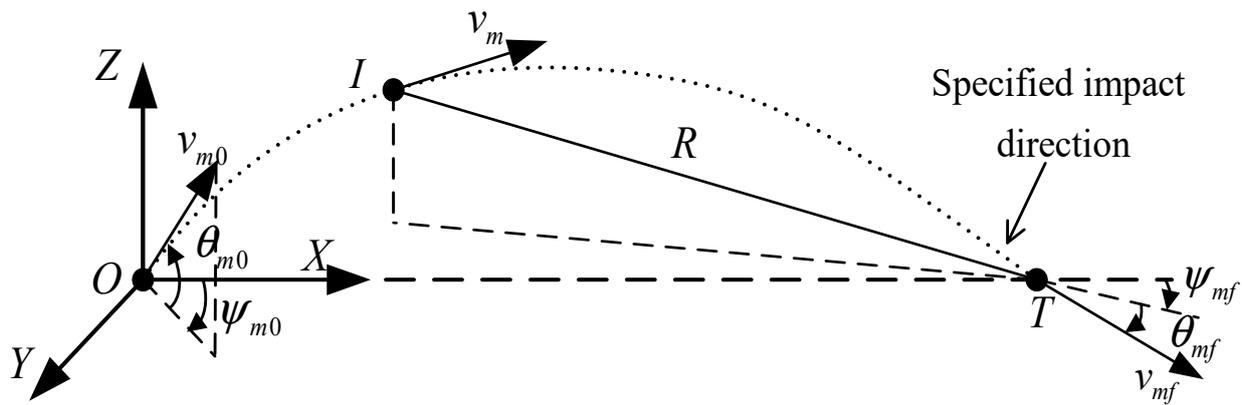


Figure 1. Three-dimensional guidance geometry.

Since air drag has a great influence on the UAV velocity, the drag model in [32,33] is adopted. The velocity change in the UAV can be expressed as

$$\dot{V} = \frac{1}{m} (T - \frac{1}{2} \rho V^2 S C_D - k_D a^2) \tag{11}$$

where  $m$ ,  $T$ , and  $S$  represent the mass, thrust, and reference area of the UAV, respectively.  $C_D$  is the parasite drag coefficient, and its relationship with the UAV velocity  $V$  is presented in Figure 2 [17].  $k_D$  is the induced drag coefficient.  $m$  and  $T$  are functions of time  $t$ , and can be represented as

$$T(t) = \begin{cases} T_{en}, & t \leq t_{en} \\ 0, & t > t_{en} \end{cases} \tag{12}$$

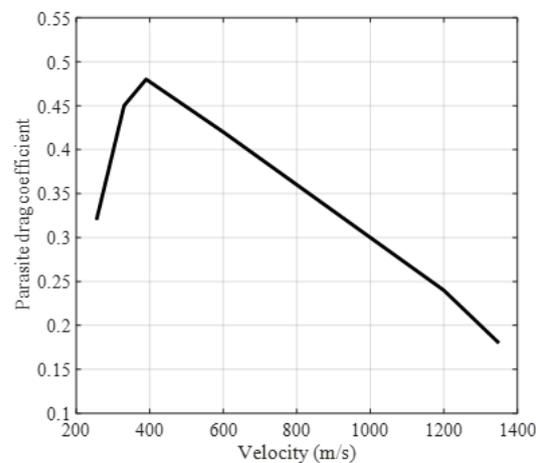
$$m(t) = \begin{cases} m_0 - \int_0^t \mu dt, & t \leq t_{en} \\ m_0 - m_{fu}, & t > t_{en} \end{cases} \tag{13}$$

where  $T_{en}$ ,  $\mu$ ,  $m_{fu}$ , and  $t_{en}$  are the thrust, fuel consumption rate, fuel mass, and maximum working time of the UAV engine, respectively.

Considering the constraints of miss distance, acceleration limitation, impact time, and angle, the design objective of the guidance law can be mathematically defined as

$$\begin{cases} \lim_{t \rightarrow t_f} x_m \rightarrow x_t, \lim_{t \rightarrow t_f} y_m \rightarrow y_t, \lim_{t \rightarrow t_f} z_m \rightarrow z_t \\ \lim_{t \rightarrow t_f} \theta_m \rightarrow \theta_{mf}, \lim_{t \rightarrow t_f} \psi_m \rightarrow \psi_{mf} \\ \sqrt{a_{zm}^2(t) + a_{ym}^2(t)} \leq a_{\max} \end{cases} \tag{14}$$

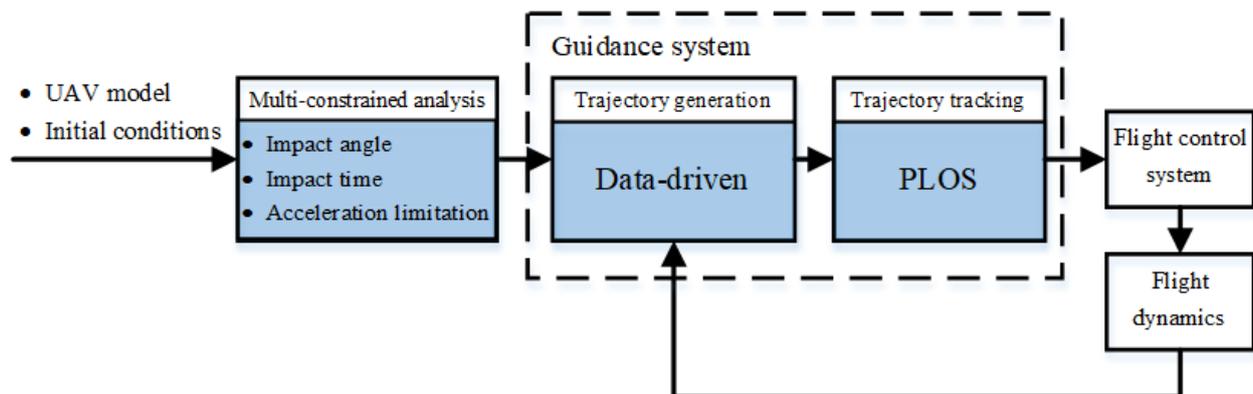
For the sake of simplicity, the launch position of the UAV is set to the origin of the inertial coordinate system  $O(0,0,0)$ . The target position is placed at  $T(x_t, 0, 0)$  on the X-axis.



**Figure 2.** Profile of the parasitic drag coefficient of the UAV with respect to velocity.

### 3. Guidance Law Design

In this section, the design concept of a two-phase trajectory is initially introduced (Section 3.1). Then, the impact angle, impact time, and acceleration limitation are analyzed (Section 3.2). Afterward, a trajectory generation data-driven method is proposed to satisfy multiple constraints (Section 3.3). Lastly, a pure pursuit and line-of-sight (PLOS) tracking algorithm is employed to generate the guidance commands (Section 3.4). Figure 3 shows the block diagram of the guidance method.



**Figure 3.** Guidance method design block diagram.

#### 3.1. Trajectory Design Concept

The impact time and angle control are realized by adjusting the LASC trajectory's length, endpoint position, and tangent vector. Denote the initial launch angle of the UAV as  $(\theta_{m0}, \psi_{m0})$ , and the desired angle at impact as  $(\theta_{mf}, \psi_{mf})$ . Near the moment that the UAV approaches the target, the terminal-phase flight trajectory can be approximated as a straight line. If the angle formed by this straight line with respect to the target is equal to the desired impact angle, the impact angle constraint is satisfied. To ensure a smooth transition between the initial and terminal trajectories, these two trajectories must be tangent at the phase switching point. Moreover, by adjusting the position of this phase switching point along the straight-line trajectory, the length of the two-phase trajectory can be adjusted along with the impact time.

The parameters of the multi-constrained trajectory remain to be determined, including LASC parameters and phase switching point position. However, the complexity of the trajectory generation process makes it very difficult and even impractical to find an analytical solution. Consequently, a data-driven method is used to design the two-phase trajectory. The design framework is illustrated in Figure 4. The initial phase trajectory is a LASC

that coincides with the launch angle at the origin, while the terminal phase trajectory is a straight-line segment that fulfills the impact angle constraint. The phase switching point is denoted as  $S$ .  $O - X_L - Y_L - Z_L$  is the local coordinate system established for solving the LASC parameters under given conditions.

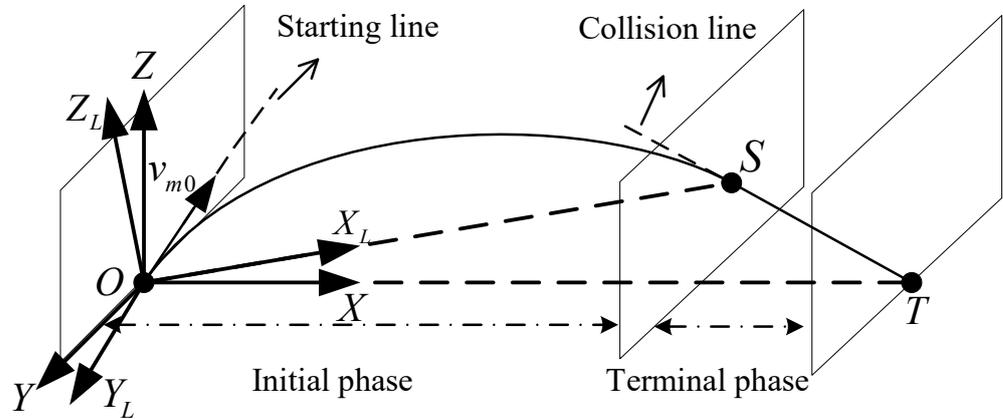


Figure 4. Two-phase guidance trajectory and geometric framework.

3.2. Multi-Constrained Analysis

In order to satisfy the impact angle constraint, it is necessary to make the initial phase trajectory tangent to the starting line at the origin point  $O$  and tangent to the collision line at the phase switching point  $S$ . Equations (5) and (6) provide insight that the LASC involves five trajectory parameters, namely  $\alpha, \beta, \Lambda, \Omega,$  and  $v$ . With the values of  $\alpha, \beta,$  and  $\Omega$  fixed, adjusting the values of  $\Lambda$  and  $v$  is adequate to satisfy the terminal positional and tangential constraints [30]. Thus, we set  $\alpha = \beta = \Omega = 0$  in this manuscript.

For simplicity, the LASC can be normalized into a standard form by scaling and rotating transformations [18,30]. The coordinates of the two endpoints are transformed to  $O(0,0,0)$  and  $S'(2,0,0)$ , with the scaling factor of  $|OS|/2$ . The initial condition is determined by the starting position  $O(0,0,0)$  and the launch angle  $(\theta_{m0}, \psi_{m0})$ . The terminal condition is determined by the endpoint position  $S'(2,0,0)$  and the impact angle  $(\theta_{mf}, \psi_{mf})$ . Thus, the LASC in the standard form can be obtained by integrating Equation (9). The generated LASC conforms to the following mapping relation  $f_1$ :

$$(L_{S_0}, \theta_{mf}, \psi_{mf}) = f_1(\theta_{m0}, \psi_{m0}, \Lambda, v, |OS|) \tag{15}$$

where  $L_{S_0}$  is the length of the LASC in the standard form.

To apply the standard-form trajectory to the inertial frame, the size of the LASC needs to be scaled as

$$L_S = L_{S_0} \cdot |OS|/2 \tag{16}$$

The length  $L$  of the entire two-phase trajectory is

$$L = L_S + |ST| \tag{17}$$

To satisfy the impact time constraint, the trajectory length needs to be adjusted by tuning the position of the phase switching point. The position of the phase switching point is determined by the length of the line segment  $|ST|$ . Therefore, the desired two-phase trajectory satisfies the following mapping relation  $f_2$ :

$$(L, \theta_{mf}, \psi_{mf}) = f_2(\theta_{m0}, \psi_{m0}, \Lambda, v, |OS|, |ST|) \tag{18}$$

The following geometric relationship can be obtained from Figure 4.

$$|OS|^2 = |ST|^2 + |OT|^2 - 2|ST| \cdot |OT|^2 \cos \angle OTS \tag{19}$$

Then, Equation (18) can be rewritten as

$$(L, \theta_{mf}, \psi_{mf}) = f_3(\theta_{m0}, \psi_{m0}, \Lambda, v, |ST|, |OT|) \tag{20}$$

Although the trajectory length and the trajectory parameters have been obtained, the impact time still needs to be calculated due to time-varying velocity. The following relationship between the trajectory length and the impact time needs to be satisfied.

$$\int_0^{t_f} V(t)dt - L = 0 \tag{21}$$

After obtaining the two-phase trajectory that satisfies the constraints of impact time and angle, it is necessary to evaluate the maximum acceleration demand during trajectory tracking. Because the acceleration demand imposes a constraint on the curvature of the trajectory as follows:

$$V^2 \cdot |\kappa(s)|_{\max} \leq a_{\max} \tag{22}$$

According to Equation (7), the curvature of the LASC is monotonically decreasing. Hence, we have

$$|\kappa(s)|_{\max} = |\kappa(0)| = |\kappa_0(0)| \cdot 2/|OS| \tag{23}$$

where  $\kappa(s)$  and  $\kappa_0(s)$  are the curvature in the inertial frame and the standard form, respectively.

Equation (20) can be supplemented as

$$(L, \theta_{mf}, \psi_{mf}, |\kappa(s)|_{\max}) = f_3(\theta_{m0}, \psi_{m0}, \Lambda, v, |ST|, |OT|) \tag{24}$$

While the UAV velocity varies with time, it is important to determine the exact maximum acceleration during trajectory tracking. To ensure the feasibility of flying along the trajectory, the maximum acceleration is evaluated using the maximum velocity and maximum curvature as

$$V_{\max} \cdot |\kappa(s)|_{\max} \leq a_{\max} \tag{25}$$

### 3.3. Trajectory Generation

#### 3.3.1. Analysis of Mapping Relationship

The LASC plays a crucial role in ensuring a smooth transition between the two phases. To satisfy the impact angle constraint while maintaining the adjustability of the trajectory length, a predefined collision line is set. The trajectory length is adjusted by tuning the phase switching point, which further realizes control over the impact time. Consequently, the trajectory parameters that need to be determined include LASC parameters  $\Lambda$  and  $v$ , initial relative distance  $|OT|$ , and straight-line trajectory length  $|ST|$ .

If the trajectory scaling factor is  $\sigma$ , then the initial relative distance  $|OT|$ , straight-line trajectory length  $|ST|$ , and trajectory length  $L$  are all scaled in the same proportion. Moreover, the maximum curvature becomes  $|\kappa(s)|_{\max}/\sigma$ , while the LASC parameters  $\Lambda$  and  $v$  remain unchanged. For the sake of simplicity, we assume the initial distance between the objects is a definite value  $R_0$  for further analysis. Thus, the Equation (24) can be rewritten as

$$(L, \theta_{mf}, \psi_{mf}, |\kappa(s)|_{\max}) = f_0(\theta_{m0}, \psi_{m0}, \Lambda, v, |ST|) \tag{26}$$

It can be inferred from Equation (26) that the trajectory parameters are determined by the initial angle, impact angle, and trajectory length. There is also a maximum curvature associated with these trajectory parameters. The corresponding mapping relation  $F_0$  can be expressed as

$$(\Lambda, v, |ST|, |\kappa(s)|_{\max}) = F_0(\theta_{m0}, \psi_{m0}, \theta_{mf}, \psi_{mf}, L) \tag{27}$$

It is worth pointing out that when a curve rotates around the X-axis, its geometric features, such as relative distances and angles, remain unchanged. As shown in Figures 5 and 6,  $v'_{m0}$  and  $v'_{mf}$  are the projection vectors of the initial unit vector  $v_{m0} = (x_0, y_0, z_0)$  and termi-

nal unit vector  $v_{mf} = (x_f, y_f, z_f)$  in the YOZ plane, respectively.  $\theta_0$  is the angle between  $v_{m0}$  and the YOZ plane, while  $\theta_f$  is the angle between  $v_{mf}$  and the YOZ plane.  $\psi_0$  is the angle between  $v'_{m0}$  and the Z-axis, while  $\psi_f$  is the angle between  $v'_{mf}$  and the Z-axis. The related relation expressions are as follows:

$$\begin{aligned} \theta_0 &= \arcsin x_0 \\ \theta_f &= \arcsin x_f \\ \psi_0 &= \arctan(y_0, z_0) \\ \psi_f &= \arctan(y_f, z_f) \end{aligned} \tag{28}$$

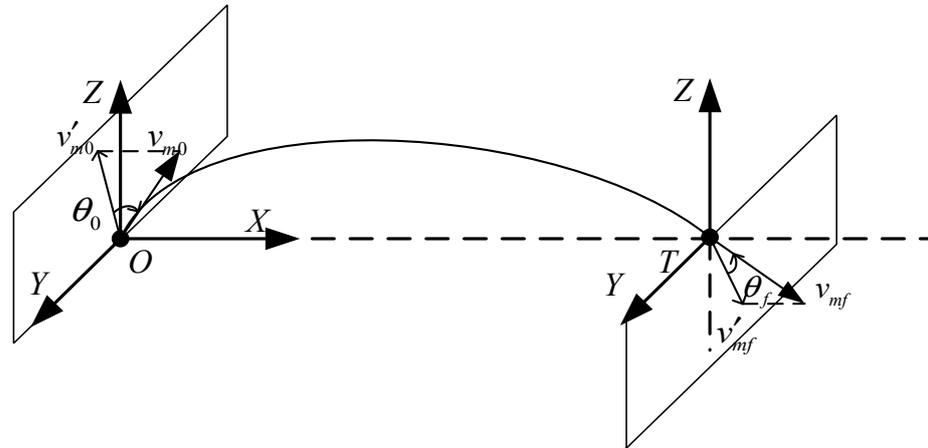


Figure 5. Illustration of the initial and terminal vectors of the curve.

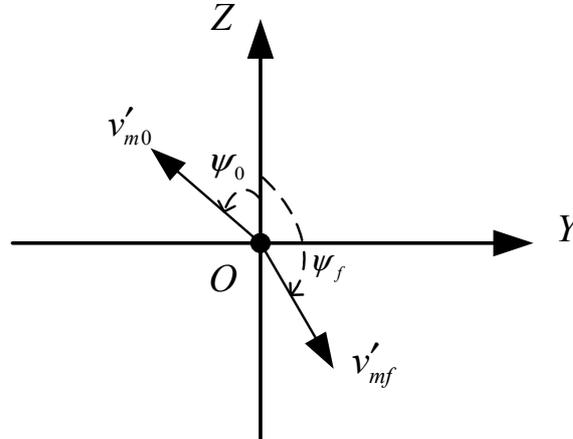


Figure 6. Illustration of the projection of the initial and terminal vectors onto the YOZ plane.

$\theta_0$ ,  $\theta_f$ , and  $\Delta\psi$  represent the relative geometry of the initial vector and terminal vector in space. The corresponding values remain unchanged in spite of rotating around the X-axis. Therefore, the Equation (27) can be further simplified as follows:

$$(\Lambda, v, |ST|, |\kappa(s)|_{\max}) = F(\theta_0, \theta_f, \Delta\psi, L) \tag{29}$$

where  $\Delta\psi = |\psi_0 - \psi_f|$ .

Compared with Equation (27), Equation (29) reduces the dimensions of the input variables from 5 to 4, which significantly reduces the computational burden of constructing the database. Additionally, the proposed trajectory design method gains better flexibility by using fewer parameters in trajectory generation.

### 3.3.2. Acquisition of Trajectory Parameters

The LASC that satisfies the terminal constraints can be obtained by solving the LASC parameters  $\Lambda$  and  $v$  with the improved simplex method [30,34]. However, the existing trajectory generation problem involves time-varying velocity, which makes it difficult to solve analytically and rapidly. Local optimization algorithms like the improved simplex method would struggle to obtain the global optimal solution, while global optimization algorithms would suffer from time-consuming computation. Hence, a data-driven method is introduced here to solve the trajectory parameters efficiently. Instead of relying on complex analytical equations, the data-driven method focuses on establishing the relationship between input and output variables, which benefits solving the trajectory parameters accurately and efficiently.

Due to the strong capability of nonlinear fitting, DNN-based modeling is a widely adopted data-driven method and has gained more and more popularity in various fields [22,35,36]. DNN is composed of multiple layers and massive neurons, which allows it to model complex relationships and patterns through training data. Through learning from large datasets, DNN can perform nonlinear mappings between input and output variables, which makes it highly effective in handling complex and high-dimensional problems.

The general training process has three main steps [37]: forward propagation, loss function calculation, and backpropagation. In the step of forward propagation, input data are passed through the network's input layer to the output layer:

$$Y^m = f^m(W^m Y^{m-1} + b^m) \quad (30)$$

where  $W^m$  and  $b^m$  are the weight coefficients and biases of the  $m$ th layer, respectively.  $Y^m$  represents the output value of the  $m$ th layer. To achieve better performance in nonlinear mapping, the activation function  $f^m$  is applied.

The predicted output values are obtained through forward propagation. Then, these predicted values are compared with the actual output values to further assess the DNN's mapping performance. The prediction error is obtained by the loss function, while the weights and biases of each neuron are tuned according to the gradient of the loss function. The commonly used loss function is the mean squared error (MSE), which is defined as follows:

$$L = \frac{1}{N} \sum_{i=1}^N (Y_i^{pred} - Y_i^{actual})^2 \quad (31)$$

where  $N$  is the number of samples.  $Y_i^{pred}$  and  $Y_i^{actual}$  represent the predicted output and actual output values of the  $i$ th sample, respectively.

In the step of backpropagation, the updates for weights and biases are performed as follows:

$$W_t^m = W_{t-1}^m - \eta \frac{\partial L}{\partial W_{t-1}^m} \quad (32)$$

$$b_t^m = b_{t-1}^m - \eta \frac{\partial L}{\partial b_{t-1}^m} \quad (33)$$

where  $W_t^m$  represents the weights of the  $m$ th layer at time  $t$ .  $b_t^m$  denotes the biases of the  $m$ th layer at time  $t$ .  $\eta$  is the learning rate that determines the step size for searching weight values.  $\partial$  represents the gradient sign. The “−” sign indicates that the weight updates are made to reduce the loss function.

Therefore, a deep neural network, denoted as  $NET_{DNN}$  and shown in Figure 7, can be trained to establish the mapping in Equation (29).  $NET_{DNN}$  is used to solve the trajectory parameters  $\Lambda$ ,  $v$ ,  $|ST|$  and the maximum curvature  $|\kappa(s)|_{\max}$ . Then, the multiple constraints can be checked according to Equations (9) and (25).

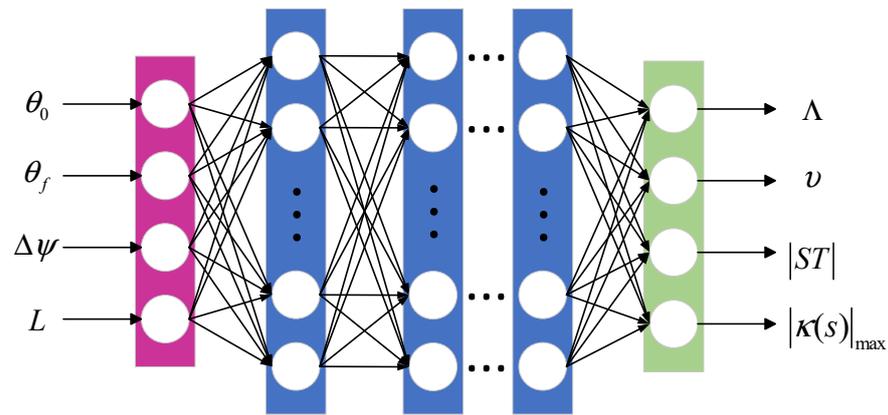


Figure 7. Structure diagram of a neural network.

### 3.4. Trajectory Tracking

It is crucial to guide the UAV along the designed trajectory in the presence of disturbances. Otherwise, the actual flight trajectory may deviate from the designed trajectory. Therefore, a closed-loop trajectory tracking method is needed. Four commonly used three-dimensional trajectory tracking algorithms are compared in [38], namely lookahead, nonlinear guidance law (NLGL), pure pursuit and line-of-sight (PLOS), and vector field. Among these algorithms, the PLOS algorithm performs the best in both accuracy and computational efficiency. Therefore, the PLOS trajectory tracking method depicted as Algorithm 1 is used to track the designed multi-constrained trajectory.

---

**Algorithm 1** Pure Pursuit and Line-of-Sight algorithm.

---

- 1: **for** each time step **do**
  - 2:     Obtain the current waypoint  $W_i = (x_1, y_1, z_1)$ , next waypoint  $W_{i+1} = (x_2, y_2, z_2)$ , current UAV position  $I = (x, y, z)$  and corresponding pitch and yaw angles  $\theta, \psi$  in the inertial frame.
  - 3:     Calculate  $\alpha_y \leftarrow \arctan\left(\frac{W_{i+1}(2) - W_i(2)}{W_{i+1}(1) - W_i(1)}\right)$ .
  - 4:     Calculate  $\psi_d \leftarrow \arctan(W_{i+1}(2) - I(2), W_{i+1}(1) - I(1))$ .
  - 5:     Calculate  $W_i^R \leftarrow R_Z(-\alpha_y)W_i, W_{i+1}^R \leftarrow R_Z(-\alpha_y)W_{i+1}, I^R \leftarrow R_Z(-\alpha_y)I$ .
  - 6:     Calculate  $\alpha_z \leftarrow \arctan\left(\frac{W_{i+1}^R(3) - W_i^R(3)}{W_{i+1}^R(1) - W_i^R(1)}\right)$ .
  - 7:     Calculate  $\theta_d \leftarrow \arctan\left(W_{i+1}^R(3) - I^R(3), W_{i+1}^R(2) - I^R(2)\right)$ .
  - 8:     Calculate  $e = R_Y(-\alpha_z)R_Z(-\alpha_y)(I - W_i)$ .
  - 9:     Calculate  $\dot{\psi}_m \leftarrow k_{1,\psi}(\psi_d - \psi) + k_{2,\psi}(-e_y)$   
 $\dot{\theta}_m \leftarrow k_{1,\theta}(\theta_d - \theta) + k_{2,\theta}(e_z)$
  - 10:     Calculate  $a_{zm}$  and  $a_{ym}$  commands from Equation (10)  
 $a_{zm} \leftarrow (k_{1,\theta}(\theta_d - \theta) + k_{2,\theta}(e_z))V$       $a_{ym} \leftarrow (k_{1,\psi}(\psi_d - \psi) + k_{2,\psi}(-e_y))V \cos \theta$
  - 11: **end for**
- 

Where  $a_{zm}$  and  $a_{ym}$  are the pitch and yaw acceleration commands, respectively.  $\alpha_z$  and  $\alpha_y$  are the rotation angles used to transform the inertial frame into the local frame with respect to UAV velocity. These rotations can be represented by rotation matrices  $R_Z$  and  $R_Y$ .  $e_z$  and  $e_y$  are the vertical and cross-track errors, respectively.  $k_{1,\theta}$  and  $k_{1,\psi}$  are the proportional gains of the pure pursuit strategy.  $k_{2,\theta}$  and  $k_{2,\psi}$  are the proportional gains of the LOS strategy.

## 4. Numerical Simulation

### 4.1. Simulation Setup

To assess the performance of the proposed guidance method, it is compared with the guidance method in [14]. The simulation experiment involves four UAVs:  $I_1, I_2, I_3,$  and  $I_4$ .

Their simulation settings are depicted in Table 1. The UAV engine thrust  $T_{en} = 10,000$  N, maximum working time of the engine  $t_{en} = 10$  s, UAV's initial mass  $m_0 = 200$  kg, fuel mass  $m_{fu} = 50$  kg, fuel consumption rate  $\mu = 5$  kg/s, induced drag coefficient  $k_D = 0.05$ , and maximum acceleration limit  $a_{max} = 100$  m/s<sup>2</sup>. The four proportional gains of the PLOS algorithm are set as  $k_{1,\theta} = k_{1,\psi} = 20$  and  $k_{2,\theta} = k_{2,\psi} = 0$ . The position of the target is (15,000, 0, 0) m. The autopilot is assumed to have a first-order lag with a time constant of 0.2 s. Thus, the flight velocity, distance, and time can be obtained through simulation.

**Table 1.** Comparison of simulation settings for UAVs.

UAV	Guidance	Launch Time	Initial Heading Direction	Desired Impact Direction
$I_1$	Proposed	$t = 0$ s	(0.4082, 0.4082, 0.8165)	(0.6247, -0.6247, -0.4685)
$I_2$	Proposed	$t = 2$ s	(0.6667, -0.3333, 0.6667)	(0.4682, 0.8109, -0.3511)
$I_3$	Ref. [14]	$t = 0$ s	(0.4082, 0.4082, 0.8165)	(0.6247, -0.6247, -0.4685)
$I_4$	Ref. [14]	$t = 2$ s	(0.6667, -0.3333, 0.6667)	(0.4682, 0.8109, -0.3511)

The simulation was performed on a hardware platform with a 2.5 GHz six-core processor and 16 GB RAM. It takes approximately 50 ms to generate a flight trajectory that satisfies the constraints of impact time, impact angle, and maximum acceleration limit.

#### 4.2. Establishment of Mapping Network

Before training network  $NET_{DNN}$ , trajectory dataset samples need to be obtained first. Here, the optimal Latin hypercube method is employed to sample the trajectory data within the design space [39]. This sampling technique ensures that the samples are evenly distributed across the entire design space. The value ranges of the trajectory parameters are set to  $\Lambda \in [0, 1]$ ,  $v \in (1.5, 15]$  and  $|ST| \in [0, 6000]$  m. Other trajectory parameters are set as  $\alpha = \beta = \Omega = 0$  and  $R_0 = 15,000$  m. Based on the trajectory parameters, the flight trajectories are obtained by integrating the Frenet–Serret Equation (9). Then the corresponding parameters  $\theta_0, \theta_f, \Delta\psi, L$  and  $|\kappa(s)|_{max}$  can be obtained from the generated trajectories. During the generation process of LASC, an undesired condition may occur in which the cumulative angular variation in tangent or binormal vectors is greater than  $2\pi$ . Such conditions should be excluded because of the excessive loss of UAV velocity in practical applications. Then, a dataset of 1000 samples is obtained, as shown in Table 2. The input data  $S_1$  are composed of parameters  $\theta_0, \theta_f, \Delta\psi$ , and  $L$ . The output data  $S_2$  are composed of parameters  $\Lambda, v, |ST|$ , and  $|\kappa(s)|_{max}$ . The mapping network  $NET_{DNN}$  is then trained using  $S_1$  and  $S_2$ . All the data are normalized before training and split into two parts. A total of 70% of the data is used for training, while the remaining 30% is used for testing.

**Table 2.** Sample data for DNN mapping network training and testing.

NO.	Input Parameter				Output Parameter			
	$\theta_0$	$\theta_f$	$\Delta\psi$	$L$	$\Lambda$	$v$	$ ST $	$ \kappa(s) _{max}$
1	24.7554	42.2519	149.8320	18,234.11	0.1860	2.1069	1284	$1.5675 \times 10^{-4}$
2	47.3558	35.4730	167.8687	18,980.46	0.5441	5.7322	4577.62	$3.7876 \times 10^{-4}$
3	36.0618	30.7972	174.3580	19,442.37	0.4036	13.3226	3539.06	$2.5077 \times 10^{-4}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
1000	35.5963	23.7348	170.6936	19,876.64	0.3774	9.4083	2949	$2.5318 \times 10^{-4}$

Then, a feed-forward fully connected DNN is developed to learn the nonlinear relationship between  $S_1$  and  $S_2$ . The appropriate selection of the hyperparameters is critical to achieving optimal performance. Here, the optimized hyperparameters are given in Table 3. The corresponding optimization process is similar to that in [36]. The training loss curve in Figure 8 shows that the DNN model keeps approaching the real model during the training

process. The coefficient of determination  $R^2$  is adopted to assess the approximation error [40]. Figure 9 illustrates the learning effect, which visualizes the prediction performance of DNN. The results demonstrate that the coefficients of determination of all four output parameters are above 0.99. Therefore, the prediction accuracy of the mapping network  $NET_{DNN}$  is high.

4.3. Case 1: UAV Guidance under Constant Velocity

In Case 1, the velocity is set to be constant and equal to  $V = 300$  m/s. The desired impact time is set to  $t_f = 65$  s. According to Equation (28), we have  $\theta_{0,1} = 24.0948^\circ$ ,  $\theta_{f,1} = 38.6598^\circ$ ,  $\Delta\psi_1 = 153.4349^\circ$ ,  $L_1 = 19,500$  m, and  $\theta_{0,2} = 41.8103^\circ$ ,  $\theta_{f,2} = 27.9152^\circ$ ,  $\Delta\psi_2 = 139.9783^\circ$ ,  $L_2 = 18,900$  m. Based on mapping network  $NET_{DNN}$ , we obtain  $\Lambda_1 = 0.1553$ ,  $v_1 = 2.2468$ ,  $|ST|_1 = 3857.56$  m,  $|\kappa(s)|_{\max,1} = 1.6785 \times 10^{-4}$ , and  $\Lambda_2 = 0.3257$ ,  $v_2 = 2.1061$ ,  $|ST|_2 = 1931.00$  m,  $|\kappa(s)|_{\max,2} = 2.2483 \times 10^{-4}$ . Both trajectories of the UAVs  $I_1$  and  $I_2$  satisfy the maximum acceleration constraint according to Equation (25).

The simulation results are shown in Figure 10 and Table 4. It can be observed that all the UAVs have successfully reached the target position. The impact time errors for UAVs  $I_1$  and  $I_2$  are both within 0.08 s, while the impact angle errors are both within  $0.02^\circ$ . Obviously, compared with UAVs  $I_3$  and  $I_4$ , UAVs  $I_1$  and  $I_2$  achieve better accuracy in both impact time and impact angle. This is because the guidance method in [14] adopts a linear estimation method to calculate the time-to-go, which would degenerate under nonlinear kinematics.

4.4. Case 2: UAV Guidance under Time-Varying Velocity

In this case, the UAV velocities are time-varying according to Equation (11). The desired impact time is set to  $t_f = 40$  s. All other simulation conditions are the same as in Case 1. According to Equation (28), we have  $\theta_{0,1} = 24.0948^\circ$ ,  $\theta_{f,1} = 38.6598^\circ$ ,  $\Delta\psi_1 = 153.4349^\circ$ ,  $L_1 = 19,433.92$  m, and  $\theta_{0,2} = 41.8103^\circ$ ,  $\theta_{f,2} = 27.9152^\circ$ ,  $\Delta\psi_2 = 139.9783^\circ$ ,  $L_2 = 18,786.81$  m. Based on mapping network  $NET_{DNN}$ , we obtain  $\Lambda_1 = 0.1413$ ,  $v_1 = 2.2106$ ,  $|ST|_1 = 3495.75$  m,  $|\kappa(s)|_{\max,1} = 1.6338 \times 10^{-4}$ , and  $\Lambda_2 = 0.3193$ ,  $v_2 = 2.0755$ ,  $|ST|_2 = 1746.41$  m,  $|\kappa(s)|_{\max,2} = 2.1801 \times 10^{-4}$ . Both trajectories of the UAVs  $I_1$  and  $I_2$  satisfy the maximum acceleration constraint according to Equation (25).

The simulation results are shown in Figure 11 and Table 5. It can be revealed that although all the UAVs have reached the target point, the guidance performances of UAVs  $I_3$  and  $I_4$  have significantly decreased. Their impact time errors are close to 2 s and the impact pitch angle errors exceed  $10^\circ$ . This is mainly because the guidance law in [14] adopts a linear estimation method for time-to-go prediction, which significantly degenerates under time-varying velocity. By contrast, the impact time errors of UAVs  $I_1$  and  $I_2$  are both less than 0.1 s, and the impact angle errors are both less than  $0.05^\circ$ . It can be seen that the time-varying velocity has little influence on the performance of the proposed guidance method. This is because the proposed method has taken the influence of variable velocity into account during neural network training.

Table 3. Hyperparameters for the DNN model.

Hyperparameters	Preferences
Optimizer	Adam
Number of hidden layers	4
Number of neurons per layer	4; 40; 40; 40; 40; 4
Activation functions	Hidden: Relu
Regularization	L2
Learning rate	0.01
Mini-batch size	150
Number of epochs	100
Weight initialization	He
Loss function	MSE

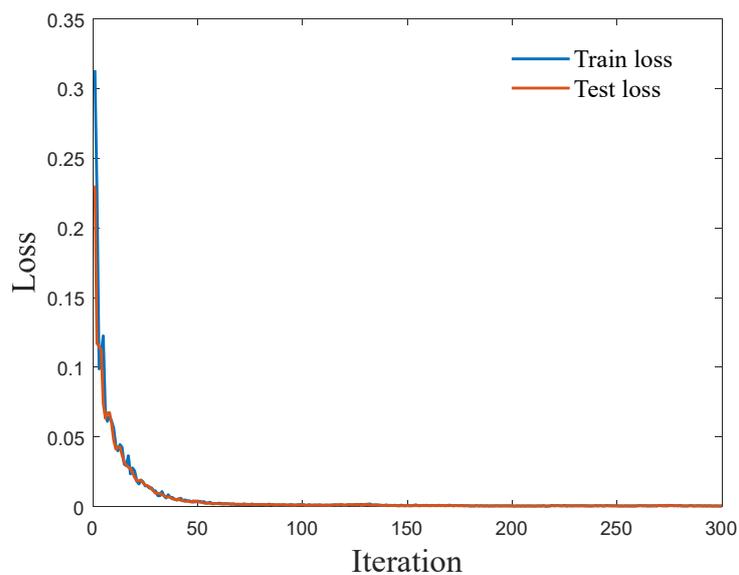


Figure 8. DNN training loss curve.

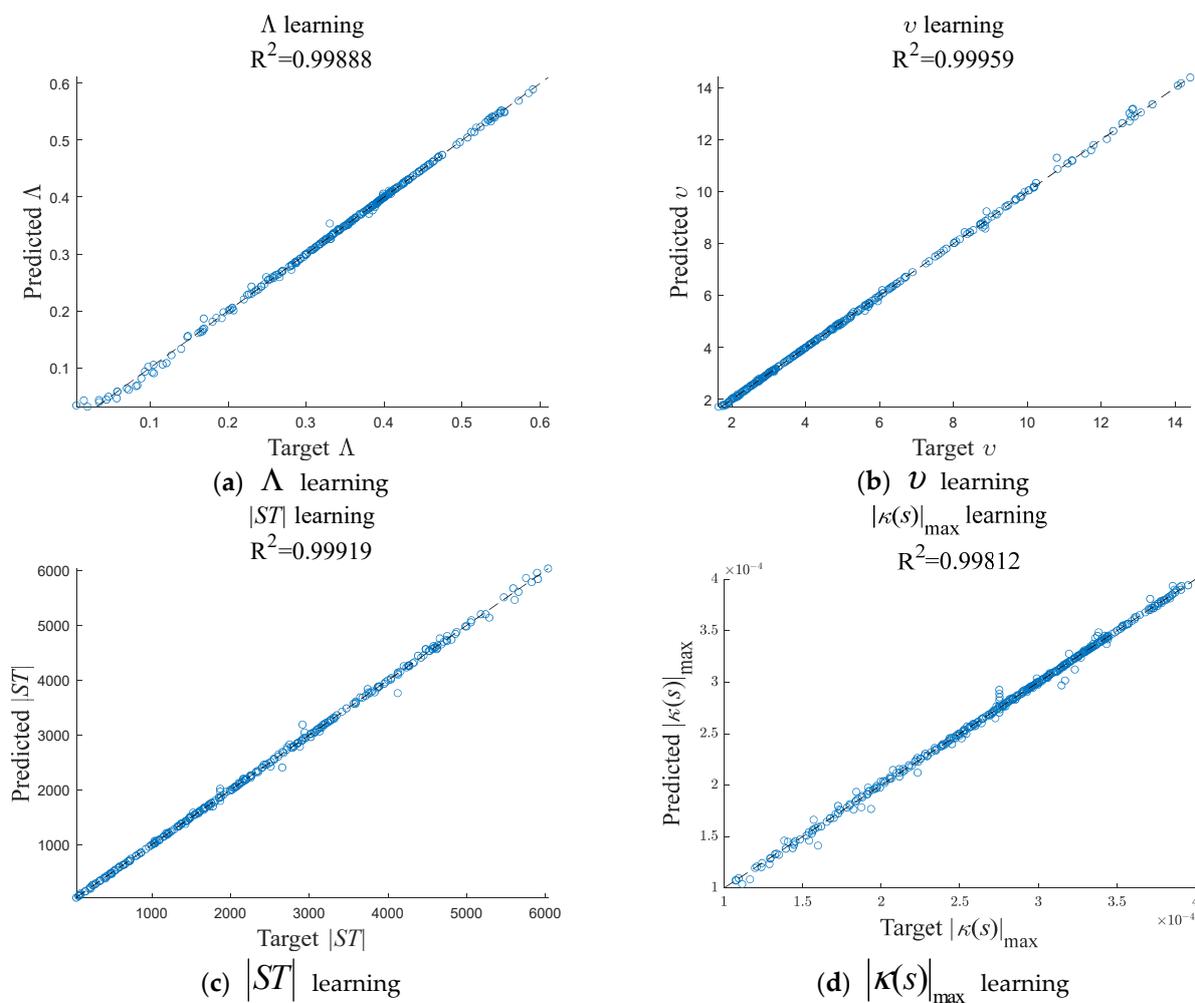
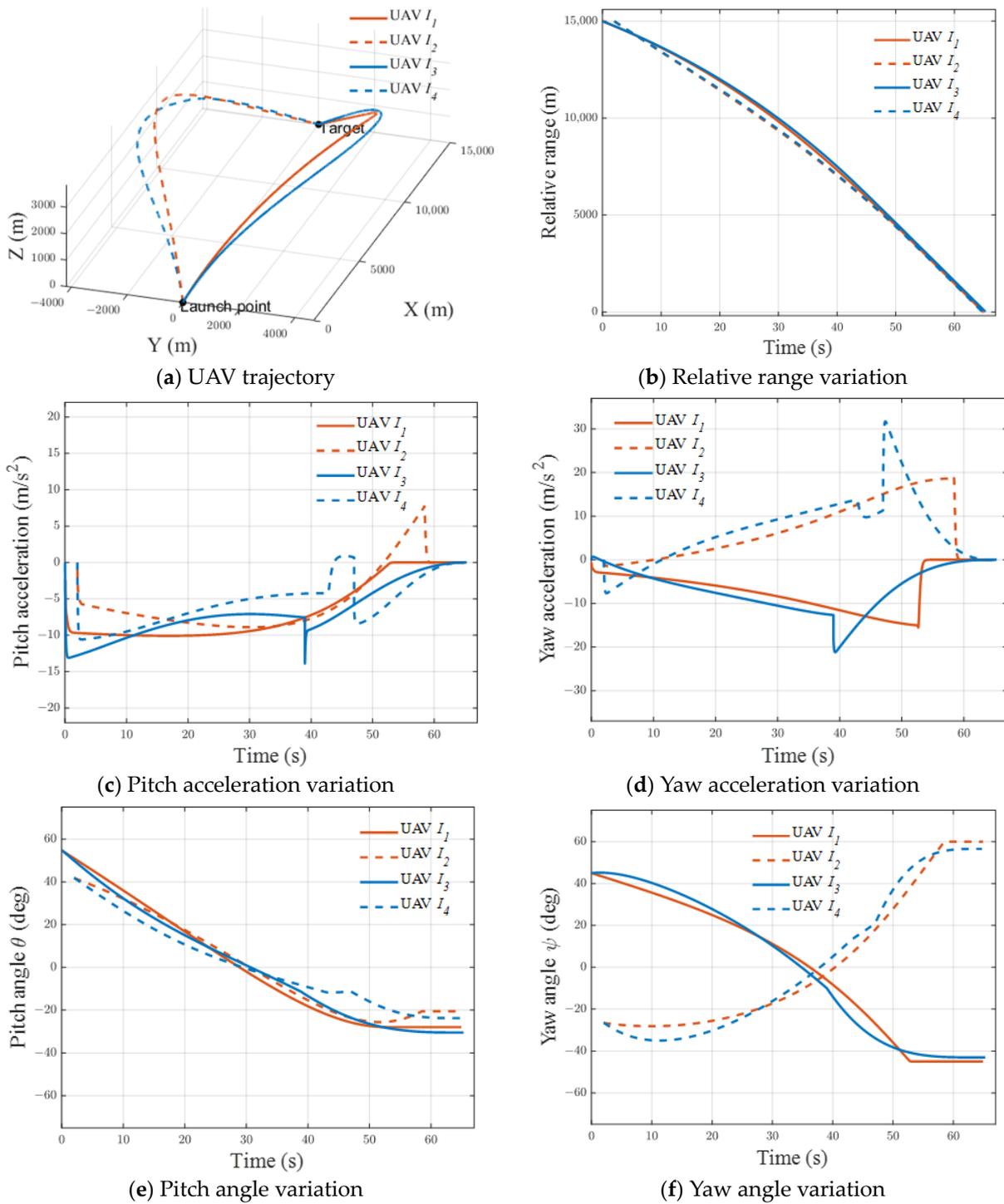


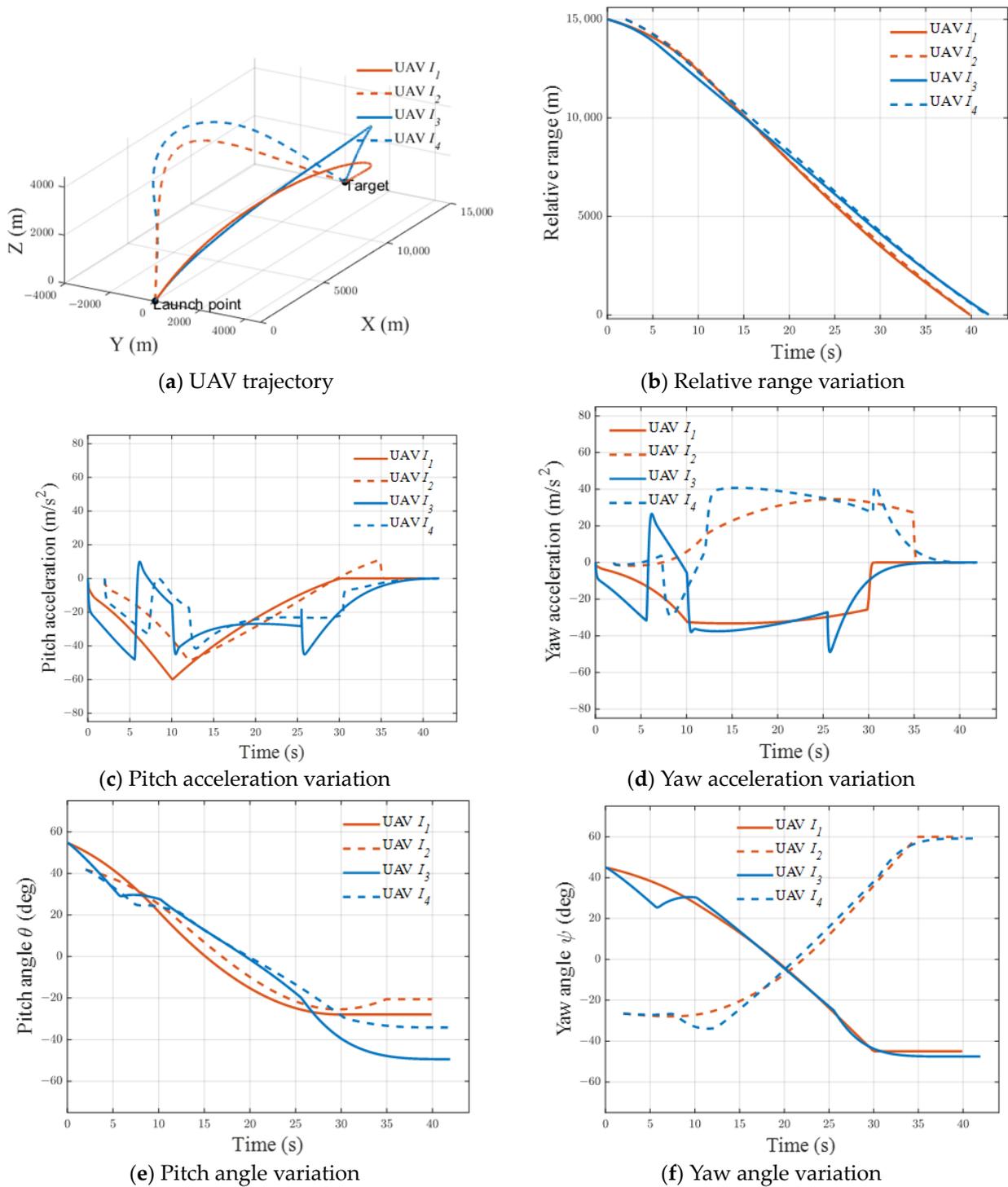
Figure 9. Learning effect of the network on parameter prediction.



**Figure 10.** Simulation results of Case 1 (UAVs  $I_1$  and  $I_2$  adopt the proposed guidance method; UAVs  $I_3$  and  $I_4$  adopt the guidance method in Reference [14]).

**Table 4.** Simulation results of Case 1.

UAV	$I_1$	$I_3$	$I_2$	$I_4$
Desired impact time			65 s	
Impact time	64.92 s	65.29 s	64.94 s	64.73 s
Desired impact pitch angle				$-20.57^\circ$
Impact pitch angle	$-27.95^\circ$	$-30.50^\circ$	$-20.55^\circ$	$-23.75^\circ$
Desired impact yaw angle				$60^\circ$
Impact yaw angle	$-45.00^\circ$	$-43.05^\circ$	$60.02^\circ$	$-56.55^\circ$



**Figure 11.** Simulation results of Case 2 (UAVs  $I_1$  and  $I_2$  adopt the proposed guidance method; UAVs  $I_3$  and  $I_4$  adopt the guidance method in Reference [14]).

**Table 5.** Simulation results of Case 2.

UAV	$I_1$	$I_3$	$I_2$	$I_4$
Desired impact time			40 s	
Impact time	39.92 s	41.93 s	39.91 s	41.71 s
Desired impact pitch angle		$-27.94^\circ$		$-20.57^\circ$
Impact pitch angle	$-27.94^\circ$	$-49.37^\circ$	$-20.56^\circ$	$-34.19^\circ$
Desired impact yaw angle		$-45^\circ$		$60^\circ$
Impact yaw angle	$-44.99^\circ$	$-47.53^\circ$	$59.97^\circ$	$-59.18^\circ$

## 5. Conclusions

This manuscript proposes a data-driven method based on a multi-constrained geometric guidance law under time-varying velocity in three dimensions. The two-phase trajectory consists of an LASC and a collision line segment. By adjusting the phase switching points, the UAV's flight time and impact angle can be precisely controlled. The characteristic parameters of the desired trajectory are calculated through DNN using a data-driven method. Finally, the guidance commands are generated using a trajectory-tracking algorithm.

The proposed guidance can generate an explicit flight trajectory, which does not rely on accurate time-to-go information. Moreover, the trajectory generation processes are greatly shortened by data-driven-based mapping. The simulation results demonstrate the high performance of the proposed guidance method in terms of impact time and angle control under both constant and time-varying velocities.

**Author Contributions:** Conceptualization, X.Y. and Y.T.; methodology, X.Y. and Y.T.; software, X.Y. and Y.T.; formal analysis, X.Y. and H.S.; writing—original draft preparation, X.Y. and Y.T.; writing—review and editing, X.Y., Y.X., H.S. and J.Z. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the National Natural Science Foundation of China (Grant No. 62203362).

**Data Availability Statement:** The authors do not have permission to share data.

**Acknowledgments:** The authors are thankful to the anonymous reviewers for their useful suggestions.

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

## References

- Kim, M.; Grider, K.V. Terminal Guidance for Impact Attitude Angle Constrained Flight Trajectories. *IEEE Trans. Aerosp. Electron. Syst.* **1973**, *6*, 852–859. [[CrossRef](#)]
- Ryoo, C.-K.; Cho, H.; Tahk, M.-J. Optimal Guidance Laws with Terminal Impact Angle Constraint. *J. Guid. Control Dyn.* **2005**, *28*, 724–732. [[CrossRef](#)]
- Ratnoo, A.; Ghose, D. Impact Angle Constrained Guidance Against Nonstationary Nonmaneuvering Targets. *J. Guid. Control Dyn.* **2010**, *33*, 269–275. [[CrossRef](#)]
- Song, K.-R.; Jeon, I.-S. Impact-Angle-Control Guidance Law with Terminal Constraints on Curvature of Trajectory. *Mathematics* **2023**, *11*, 974. [[CrossRef](#)]
- Jeon, I.-S.; Lee, J.-I.; Tahk, M.-J. Impact-Time-Control Guidance Law for Anti-Ship Missiles. *IEEE Trans. Control Syst. Technol.* **2006**, *14*, 260–266. [[CrossRef](#)]
- Cho, D.; Kim, H.J.; Tahk, M.-J. Nonsingular Sliding Mode Guidance for Impact Time Control. *J. Guid. Control Dyn.* **2016**, *39*, 61–68. [[CrossRef](#)]
- Tekin, R.; Erer, K.S.; Holzapfel, F. Polynomial Shaping of the Look Angle for Impact-Time Control. *J. Guid. Control Dyn.* **2017**, *40*, 2668–2673. [[CrossRef](#)]
- Shi, H.; Chen, Z.; Zhu, J.; Kuang, M. Model Predictive Guidance for Active Aircraft Protection from a Homing Missile. *IET Control Theory Appl.* **2022**, *16*, 208–218. [[CrossRef](#)]
- Lee, J.-I.; Jeon, I.-S.; Tahk, M.-J. Guidance Law to Control Impact Time and Angle. *IEEE Trans. Aerosp. Electron. Syst.* **2007**, *43*, 301–310. [[CrossRef](#)]
- Zhang, Z.; Ma, K.; Zhang, G.; Yan, L. Virtual Target Approach-Based Optimal Guidance Law with Both Impact Time and Terminal Angle Constraints. *Nonlinear Dyn.* **2022**, *107*, 3521–3541. [[CrossRef](#)]

11. Deng, Y.; Ren, J.; Wang, X.; Cai, Y. Midcourse Iterative Guidance Method for the Impact Time and Angle Control of Two-Pulse Interceptors. *Aerospace* **2022**, *9*, 323. [[CrossRef](#)]
12. Harl, N.; Balakrishnan, S.N. Impact Time and Angle Guidance With Sliding Mode Control. *IEEE Trans. Control Syst. Technol.* **2012**, *20*, 1436–1449. [[CrossRef](#)]
13. Hou, Z.; Yang, Y.; Liu, L.; Wang, Y. Terminal Sliding Mode Control Based Impact Time and Angle Constrained Guidance. *Aerosp. Sci. Technol.* **2019**, *93*, 105142. [[CrossRef](#)]
14. Jung, B.; Kim, Y. Guidance Laws for Anti-Ship Missiles Using Impact Angle and Impact Time. In Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit, Keystone, CO, USA, 21–24 August 2006; Guidance, Navigation, and Control and Co-located Conferences. American Institute of Aeronautics and Astronautics: Palo Alto, CA, USA, 2006; p. 6432.
15. Zhang, Y.; Ma, G.; Liu, A. Guidance Law with Impact Time and Impact Angle Constraints. *Chin. J. Aeronaut.* **2013**, *26*, 960–966. [[CrossRef](#)]
16. Zhang, Y.; Zhang, Y.; Li, H. Impact Time and Impact Angle Control Based on CCC Path Planning. In Proceedings of the 31st Chinese Control Conference, Hefei, China, 25–27 July 2012; pp. 4300–4305.
17. Yan, X.; Kuang, M.; Zhu, J. A Geometry-Based Guidance Law to Control Impact Time and Angle under Variable Speeds. *Mathematics* **2020**, *8*, 1029. [[CrossRef](#)]
18. Yan, X.; Zhu, J.; Kuang, M.; Yuan, X. A Computational-Geometry-Based 3-Dimensional Guidance Law to Control Impact Time and Angle. *Aerosp. Sci. Technol.* **2020**, *98*, 105672. [[CrossRef](#)]
19. Kirchdoerfer, T.; Ortiz, M. Data-Driven Computational Mechanics. *Comput. Methods Appl. Mech. Eng.* **2016**, *304*, 81–101. [[CrossRef](#)]
20. Hao, G.; Ni, W.; Tian, H.; Cao, L. Mobility-Aware Trajectory Design for Aerial Base Station Using Deep Reinforcement Learning. In Proceedings of the 2020 International Conference on Wireless Communications and Signal Processing (WCSP), Nanjing, China, 21–23 October 2020; pp. 1131–1136.
21. Arani, A.H.; Hu, P.; Zhu, Y. HAPS-UAV-Enabled Heterogeneous Networks: A Deep Reinforcement Learning Approach. *IEEE Open J. Commun. Soc.* **2023**, *4*, 1745–1760. [[CrossRef](#)]
22. Cheng, L.; Jiang, F.; Wang, Z.; Li, J. Multiconstrained Real-Time Entry Guidance Using Deep Neural Networks. *IEEE Trans. Aerosp. Electron. Syst.* **2021**, *57*, 325–340. [[CrossRef](#)]
23. Guo, Y.; Li, X.; Zhang, H.; Cai, M.; He, F. Data-Driven Method for Impact Time Control Based on Proportional Navigation Guidance. *J. Guid. Control Dyn.* **2020**, *43*, 955–966. [[CrossRef](#)]
24. Huang, J.; Chang, S.; Chen, Q.; Zhang, H. Data-Driven-Based Impact Time Control Guidance Law Independent of Time-to-Go. *Acta Armamentarii* **2022**, *44*, 2299. [[CrossRef](#)]
25. Huang, J.; Chang, S. Data-Driven Method Based Impact Time and Impact Angle Control Guidance Law. *Syst. Eng. Electron.* **2022**, *44*, 3213–3220. [[CrossRef](#)]
26. Ziatdinov, R.; Yoshida, N.; Kim, T. Analytic Parametric Equations of Log-Aesthetic Curves in Terms of Incomplete Gamma Functions. *Comput. Aided Geom. Des.* **2012**, *29*, 129–140. [[CrossRef](#)]
27. Gobithaasan, R.U.; Teh, Y.M.; Miura, K.T.; Ong, W.E. Lines of Curvature for Log Aesthetic Surfaces Characteristics Investigation. *Mathematics* **2021**, *9*, 2699. [[CrossRef](#)]
28. Miura, K.T.; Gobithaasan, R.U. Aesthetic Design with Log-Aesthetic Curves Log-Aesthetic Curve and Surfaces Log-Aesthetic Curve and Surface. In *Mathematical Progress in Expressive Image Synthesis III*; Dobashi, Y., Ochiai, H., Eds.; Springer: Singapore, 2016; pp. 107–119.
29. Luca, L.; Popescu, I.; Cherciu, M.; Ghimisi, S.; Cirtina, M.L.; Pasare, M.M. Synthesis of Two New Mechanisms Which Generate a Highly Aesthetic Design Image—The Flower of Life. *Appl. Sci.* **2020**, *10*, 1670. [[CrossRef](#)]
30. Yoshida, N.; Fukuda, R.; Saito, T. Log-Aesthetic Space Curve Segments. In Proceedings of the 2009 SIAM/ACM Joint Conference on Geometric and Physical Modeling, San Francisco, CA, USA, 5–8 October 2009; Association for Computing Machinery: New York, NY, USA, 2009; pp. 35–46.
31. Struik, D.J. *Lectures on Classical Differential Geometry*; Courier Corporation: North Chelmsford, MA, USA, 1961; ISBN 978-0-486-65609-0.
32. Davidovitz, A.; Shinar, J. Two-Target Game Model of an Air Combat with Fire-and-Forget All-Aspect Missiles. *J. Optim. Theory Appl.* **1989**, *63*, 133–165. [[CrossRef](#)]
33. Alkaher, D.; Moshaiov, A. Dynamic-Escape-Zone to Avoid Energy-Bleeding Coasting Missile. *J. Guid. Control Dyn.* **2015**, *38*, 1908–1921. [[CrossRef](#)]
34. Nelder, J.A.; Mead, R. A Simplex Method for Function Minimization. *Comput. J.* **1965**, *7*, 308–313. [[CrossRef](#)]
35. LeCun, Y.; Bengio, Y.; Hinton, G. Deep Learning. *Nature* **2015**, *521*, 436–444. [[CrossRef](#)]
36. Cheng, L.; Wang, Z.; Jiang, F.; Li, J. Fast Generation of Optimal Asteroid Landing Trajectories Using Deep Neural Networks. *IEEE Trans. Aerosp. Electron. Syst.* **2020**, *56*, 2642–2655. [[CrossRef](#)]
37. Yang, M.; Liang, S.; Yi, M.; Tian, Y.; Guo, M.; Le, J.; Zhang, H. Identification of Turbulence Eddy Viscosity Coefficient in Supersonic Isolation Section Based on Deep Neural Network. *J. Aerosp. Power* **2023**, *38*, 312–324. [[CrossRef](#)]

38. Pelizer, G.V.; da Silva, N.B.F.; Branco, K.R.L.J. Comparison of 3D Path-Following Algorithms for Unmanned Aerial Vehicles. In Proceedings of the 2017 International Conference on Unmanned Aircraft Systems (ICUAS), Miami, FL, USA, 13–16 June 2017; pp. 498–505.
39. Jin, R.; Chen, W.; Sudjianto, A. An Efficient Algorithm for Constructing Optimal Design of Computer Experiments. *J. Stat. Plan. Inference* **2005**, *134*, 268–287. [[CrossRef](#)]
40. Nagelkerke, N.J.D. A Note on a General Definition of the Coefficient of Determination. *Biometrika* **1991**, *78*, 691–692. [[CrossRef](#)]

**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.