aerospace

MDPI

Article

Guidance Design for Escape Flight Vehicle against Multiple Pursuit Flight Vehicles Using the RNN-Based Proximal Policy Optimization Algorithm

Xiao Hu [1,2], Hongbo Wang [2], Min Gong [2] and Tianshu Wang [1,*]

[1] School of Aerospace Engineering, Tsinghua University, Beijing 100084, China; huxiao18@mails.tsinghua.edu.cn
[2] China Academy of Launch Vehicle Technology, Beijing 100076, China
* Correspondence: tswang@mail.tsinghua.edu.cn

Abstract: Guidance commands of flight vehicles can be regarded as a series of data sets having fixed time intervals; thus, guidance design constitutes a typical sequential decision problem and satisfies the basic conditions for using the deep reinforcement learning (DRL) technique. In this paper, we consider the scenario where the escape flight vehicle (EFV) generates guidance commands based on the DRL technique, while the pursuit flight vehicles (PFVs) derive their guidance commands employing the proportional navigation method. For every PFV, the evasion distance is described as the minimum distance between the EFV and the PFV during the escape-and-pursuit process. For the EFV, the objective of the guidance design entails progressively maximizing the residual velocity, which is described as the EFV's velocity when the last evasion distance is attained, subject to the constraint imposed by the given evasion distance threshold. In the outlined problem, three dimensionalities of uncertainty emerge: (1) the number of PFVs requiring evasion at each time instant; (2) the precise time instant at which each of the evasion distances can be attained; (3) whether each attained evasion distance exceeds the given threshold or not. To solve the challenging problem, we propose an innovative solution that integrates the recurrent neural network (RNN) with the proximal policy optimization (PPO) algorithm, engineered to generate the guidance commands of the EFV. Initially, the model, trained by the RNN-based PPO algorithm, demonstrates effectiveness in evading a single PFV. Subsequently, the aforementioned model is deployed to evade additional PFVs, thereby systematically augmenting the model's capabilities. Comprehensive simulation outcomes substantiate that the guidance design method based on the proposed RNN-based PPO algorithm is highly effective.

Keywords: escape flight vehicle; multiple pursuit flight vehicles; guidance design; recurrent neural network (RNN); proximal policy optimization (PPO)

Citation: Hu, X.; Wang, H.; Gong, M.; Wang, T. Guidance Design for Escape Flight Vehicle against Multiple Pursuit Flight Vehicles Using the RNN-Based Proximal Policy Optimization Algorithm. *Aerospace* **2024**, *11*, 361. https://doi.org/10.3390/aerospace11050361

Academic Editor: Konstantinos Kontis

Received: 13 March 2024
Revised: 18 April 2024
Accepted: 28 April 2024
Published: 30 April 2024

1. Introduction

In modern escape-and-pursuit scenarios involving escape flight vehicles (EFVs) and pursuit flight vehicles (PFVs), the guidance design method has garnered significant scholarly attention as a pivotal strategy for addressing the intricate dynamics of escape and pursuit. In the scenario entailing a single EFV and a single PFV, numerous efficacious evasion guidance methods have been meticulously proposed, as documented in references [1–4]. Nonetheless, the aforementioned research outcomes are not directly applicable to scenarios involving a single EFV and multiple PFVs, which constitutes the primary focus of this paper. In the concerned scenario, many scholars [5–8] have explored guidance design methods from the perspective of the pursuit side. However, these findings are inapplicable to the escape side, the primary concern of this paper, owing to the fundamental divergence in objectives between PFVs and EFVs. In order to improve the escape capability of EFVs, the following three research areas have been widely studied, namely, evading based on

the no-fly zones, evading based on the optimal control theory, and evading based on deep reinforcement learning (DRL).

In the domain of guidance design based on the no-fly zones, Liang et al. [9] articulated a guidance method grounded in a feedback mechanism tailored for irregularly shaped no-fly zones, subsequently substantiating the algorithm's effectiveness. Liang et al. [10] introduced an innovative design method predicated on dynamic pressure for yaw angle control, facilitating continuous circumnavigation of multiple no-fly zones. Zhao et al. [11] delineated a design approach leveraging multi-stage convex optimization to adeptly navigate around multiple no-fly zones, with the algorithm's performance rigorously validated. Zhou et al. [12] proposed an approach by redefining a maneuverable design dilemma as an obstacle evasion challenge, achieving a flight vehicle's penetration trajectory with exemplary performance. Yu et al. [13] proposed a method to derive the high-precision analytical solution of flight time for a group of coordinated flight vehicles, and designed a guidance law to achieve the goal of simultaneous arrival in the presence of multiple no-fly zones. The aforementioned research work did not encompass a comprehensive consideration of the PFVs' pursuit capabilities. Consequently, assuring EFV evasion effectiveness becomes challenging in scenarios marked by intense confrontation, wherein PFVs possess the capability to directly approach the EFV.

In the domain of guidance design based on the optimal control theory, Yan et al. [14] delved into the escape dynamics of flight vehicles amidst a scenario characterized by the coordinated pursuit from multiple PFVs, proposing an evasion guidance strategy framed in a constrained optimal control paradigm. Wang et al. [15] studied the escape-and-pursuit game problem and demonstrated that the escape flight vehicle employed differential game theory can evade much easier compared to traditional strategies. Shen et al. [16] tackled the trajectory optimization challenge for glide vehicles confronted by two PFVs, crafting an evasion strategy predicated solely on the initial line-of-sight (LOS) angles derived from the interceptors. Nath et al. [17] investigated a game theoretical conundrum featuring two PFVs and a single EFV, devising a bifurcated evasion strategy for the latter, ingeniously integrating the path planning method with adept maneuvering. In contrast to the methods derived from the no-fly zones research, maneuvers based on optimal control exhibit markedly enhanced adaptability to dynamic conditions. However, the solution to the escape-and-pursuit problem invariably entails the online management of intricate equations, which requires a large amount of computing resources. Hence, this method is inappropriate for EFVs, which have limited computing resources.

In the domain of guidance design based on the DRL technique, He et al. [18] proposed a guidance method based on the deep deterministic policy gradient (DDPG) algorithm, which adeptly balances guidance accuracy, energy efficiency, and pursuit timing, thereby yielding superior performance relative to conventional guidance paradigms. Jiang et al. [19] proposed an evasion guidance law utilizing the actor–critic (AC) algorithm, potentially apt for scenarios involving multiple PFVs. Shen et al. [20] concentrated on formulating evasion trajectories for EFVs, aimed at evading two PFVs. The proposed method, augmented by deep neural networks, showcased effectiveness in scenarios with a fixed number of PFVs. Guo et al. [21] introduced an intelligent maneuvering strategy founded on the twin delayed deep deterministic policy gradient (TD3) algorithm, with its performance validated through simulation results. Hui et al. [22] proposed a novel training method for bank angle optimization utilizing the proximal policy optimization (PPO) algorithm. They demonstrated that the PPO algorithm could engender the performance of a "new quality", surpassing the capabilities traditionally achieved by conventional guidance design methods. Pham et al. [23,24] introduced an innovative Takagi–Sugeno–Kang elliptic type-2 fuzzy brain-imitated neural network (TSKET2FBINN), which they integrated into the guidance design of the flight vehicles. The effectiveness of the proposed method was substantiated through simulation results obtained from scenarios involving three flight vehicles. The aforementioned studies collectively underscore the effectiveness of the DRL technique in addressing escape-and-pursuit challenges. Nevertheless, their focus is pre-

dominantly on evasion strategies tailored for scenarios with a predetermined number of PFVs, thereby limiting their applicability in scenarios where the number of PFVs is variable.

Against this backdrop, this paper aims to employ a recurrent neural network (RNN)-based proximal policy optimization (PPO) algorithm to generate real-time guidance commands for the EFV, particularly in scenarios characterized by a varying number of PFVs, which rely on traditional methods for their guidance commands. Our novel contributions are summarized as follows.

1. The agent, employing the fully connected neural network (FCNN), is restricted to addressing problems with a fixed dimensionality of the input state s_t due to the fixed number of nodes in the FCNN's input layer. Given that the dimensionality of the input state s_t is a positive correlation with the number of PFVs requiring evasion (denoted as i_P), an FCNN-based agent alone is not capable of addressing the problem characterized by a varying number of PFVs in the escape-and-pursuit scenario. To address this challenging problem, we design a composite architecture integrating both the RNN and the FCNN. The proposed architecture employs the RNN to effectively handle the varying number of PFVs. Specifically, (1) the input of the RNN consists of a series of data sets, each comprising six elements representing the vector of relative position and velocity between the EFV and the ith PFV; (2) the number of data sets corresponds to the number of PFVs requiring evasion (i_P); (3) the input state s_t, with the dimensionality of $6 \times i_P$, undergoes processing i_P times, with each processing step involving six elements; (4) The output of the RNN is defined as the last hidden state of the RNN. Regarding the FCNN, the number of nodes in its input layer matches the dimensionality of the RNN's hidden state. Consequently, the RNN and FCNN can be interconnected, enabling the FCNN to generate guidance commands for the EFV.
2. The hidden state of the RNN is crucial for generating a reasonable output based on the integration of both previous and current input states; thus, it is essential to utilize the hidden state in training the agent of the EFV. In the conventional DRL techniques, the training data in the form of (s_t, a_t, r_t, s_{t+1}), with each element having fixed dimensionality, are produced through the ongoing interactions between the agent and its environment, and then stored in the replay buffer. Subsequently, a batch of training data are randomly selected from the replay buffer to facilitate the training of the agent. To address the challenge of variable dimensionality in s_t, we have developed a two-step strategy. In the first step, we incorporate both the current hidden state h_t and the next hidden state h_{t+1} into each training data, transforming its structure from (s_t, a_t, r_t, s_{t+1}) to $(s_t, a_t, r_t, s_{t+1}, h_t, h_{t+1})$. In the second step, we introduce an innovative dual-layer batch training approach. Specifically, the outer layer batch is constructed by segmenting the replay buffer based on the number of PFVs, thus ensuring that all training data in the same outer layer batch possess consistent dimensionality. Regarding the inner layer batch, it is generated by randomly selecting training data from the corresponding outer layer batch. These data, characterized by consistent dimensionality, are then utilized to train the agent in the EFV using the method employed in conventional DRL techniques.
3. The purpose of this paper is to generate optimal guidance commands that enable the EFV to effectively evade the PFVs while maximizing the residual velocity. To address the problem, a novel reward function is designed, by taking into account the prospective states (i.e., evasion distance and residual velocity) derived from a virtual scenario where the guidance commands of the EFV are predefined, facilitating rapid acquisition of feasible evasion distances and residual velocities. Given that this design uses future information for current decision-making, the agent of the EFV invoked for continuously generating the guidance commands according to the various real-time situations of the EFV and the PFVs can be trained in a more efficient manner.

The rest of this paper is organized as follows. In Section 2, we first present the system model, and then formulate the problem and analyze its complexity. In Section 3, we propose the evasion guidance design method based on the RNN-based PPO algorithm. In Section 4,

the simulation results of guidance design are presented and discussed. Our conclusions are drawn in Section 5.

2. System Model and Problem Formulation

2.1. System Model

We consider a escape-and-pursuit scenario composed of a single EFV and up to three PFVs, in which the EFV's guidance commands are generated with the aid of the DRL technique, and each PFV's guidance commands are generated based on the traditional method. For clarity, the combat scenario is illustrated in the geocentric coordinate system, as shown in Figure 1.

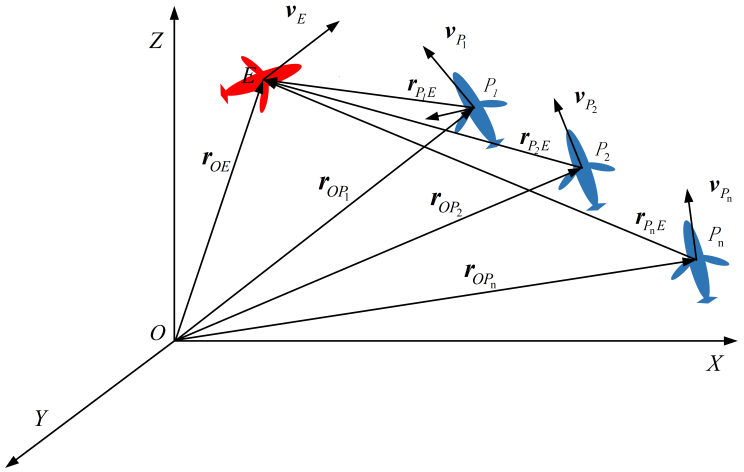


Figure 1. A escape-and-pursuit scenario composed of a single EFV and up to three PFVs in the geocentric coordinate system.

Specifically, O represents the center of the Earth and is also the origin of the geocentric coordinate system, while OX, OY, and OZ are the three axes of the geocentric coordinate system. E denotes the center of the mass of the EFV; $\boldsymbol{r}_{OE} = [r_{x,E}, r_{y,E}, r_{z,E}]$ is the position vector from the center of mass of the EFV to the Earth's center, and $\boldsymbol{v}_E = [v_{x,E}, v_{y,E}, v_{z,E}]$ is the velocity vector of the EFV. Additionally, P_i denote the center of the mass of the ith PFV; $\boldsymbol{r}_{OP_i} = [r_{x,P_i}, r_{y,P_i}, r_{z,P_i}]$ is the position vector from the center of mass of the ith PFV to the Earth's center, and $\boldsymbol{v}_{OP_i} = [v_{x,P_i}, v_{y,P_i}, v_{z,P_i}]$ is the velocity vector of the ith PFV.

Furthermore, the following assumptions are made:

1. Both the EFV and PFV can accurately observe the present and historical position and velocity of each other, and can use the information to generate its own guidance commands. Nonetheless, the future position and velocity of them are hard to predict due to the interacting behavior of the EFV and each PFV.
2. The EFV is characterized by its plane-symmetrical structure, with its guidance commands formulated through the DRL technique. These commands are primarily composed of the composite angle of attack (The composite angle of attack is mathematically defined as $\underline{\alpha}_{\text{cx}} = \arccos(\cos\alpha_{\text{cx}} \cdot \cos\beta_{\text{cx}})$.) (denoted by $\underline{\alpha}_{\text{cx}}$) and the angle of heel (denoted by γ_{cx}). The range of $\underline{\alpha}_{\text{cx}}$ is $[-16.0°, 16.0°]$, and the range of γ_{cx} is $[-90.0°, 90.0°]$. Conversely, each PFV demonstrates axial symmetry with its guidance commands derived using the proportional navigation technique [25]. These commands incorporate the angle of attack (represented by α_{cx}) and the angle of sideslip (denoted by β_{cx}). The range of both α_{cx} and β_{cx} is $[-20.0°, 20.0°]$.
3. The EFV is capable of detecting PFV when the distance between them is less than 2000.0 m, and the PFV possesses the capability to capture the EFV if the distance between them is less than 20.0 m. More precisely, the exact number of PFVs requiring evasion is determined by two factors: firstly, the total number of PFVs present in the scenario; secondly, the number of instances where the distance between the EFV and

each PFV is less than 2000.0 m. Furthermore, to successfully evade the PFVs, the EFV is required to maintain a distance larger than 20.0 m for each PFV during the entire escape-and-pursuit process.

4. For both the EFV and the PFVs, the time interval between the generation of successive guidance commands is maintained at a fixed value. Specifically, the step time $\Delta t = t_{i+1} - t_i$ remains constant throughout the escape-and-pursuit simulation.

The purpose of this paper is to use the DRL technique to generate the guidance commands of the EFV so that the EFV's residual velocity satisfying evasion distance constraint can be maximized. Since the PFVs can be regarded as a rival to facilitate the evaluation of the performance of the guidance command generating method designed for the EFV, it is sufficient to assume that the PFVs use the conventional proportional navigation method to generate their guidance commands.

The vector form of the kinematics model of flight vehicles in the geocentric coordinate system [25] is expressed as

$$\begin{cases} m_E \dfrac{\mathrm{d}^2 \boldsymbol{r}_{OE}}{dt^2} = \boldsymbol{F}_E + \boldsymbol{R}_E + m_E \boldsymbol{g_E}, \\ m_{P_i} \dfrac{\mathrm{d}^2 \boldsymbol{r}_{OP_i}}{dt^2} = \boldsymbol{F}_{P_i} + \boldsymbol{R}_{P_i} + m_{P_i} \boldsymbol{g_{P_i}}, \end{cases} \tag{1}$$

where m_E is the mass of the EFV, $\boldsymbol{F}_E = [F_{x,E}, F_{y,E}, F_{z,E}]$ is the EFV's control force vector with each element being a function of $\underline{\alpha}_{\mathrm{cx}}$ and γ_{cx}, $\boldsymbol{R}_E = [R_{x,E}, R_{y,E}, R_{z,E}]$ is the EFV's aerodynamic force vector with each element also being a function of $\underline{\alpha}_{\mathrm{cx}}$ and γ_{cx}, and $\boldsymbol{g}_E = [g_{x,E}, g_{y,E}, g_{z,E}]$ is the EFV's acceleration vector of gravity, whose elements are the functions of $\boldsymbol{r}_{OE}$. In addition, m_{P_i} is the mass of the ith PFV, $\boldsymbol{F}_{P_i} = [F_{x,P_i}, F_{y,P_i}, F_{z,P_i}]$ is the ith PFV's control force vector, $\boldsymbol{R}_{P_i} = [R_{x,P_i}, R_{y,P_i}, R_{z,P_i}]$ is the ith PFV's aerodynamic force vector, and $\boldsymbol{g}_{P_i} = [g_{x,P_i}, g_{y,P_i}, g_{z,P_i}]$ is the acceleration vector of gravity. Each element of $\boldsymbol{F}_{P_i}$ and $\boldsymbol{R}_{P_i}$ is a function of α_{cx_i} and β_{cx_i}, while each element of $\boldsymbol{g}_{P_i}$ is a function of $\boldsymbol{r}_{OP_i}$.

2.2. Evasion Distance and Residual Velocity

The time-varying relative distance vector between the EFV and the ith PFV is $\boldsymbol{d_i}(t)$, which is constituted by $d_{i,x}(t)$, $d_{i,y}(t)$, and $d_{i,z}(t)$. The value of the relative distance $d_i(t)$ is given by

$$\begin{cases} d_{i,x}(t) = r_{x,E}(t) - r_{x,P_i}(t), \\ d_{i,y}(t) = r_{y,E}(t) - r_{y,P_i}(t), \\ d_{i,z}(t) = r_{z,E}(t) - r_{z,P_i}(t), \\ d_i(t) = \sqrt{d_{i,x}(t)^2 + d_{i,y}(t)^2 + d_{i,z}(t)^2}. \end{cases} \tag{2}$$

The evasion distance is the minimum relative distance between the EFV and the ith PFV during the escape-and-pursuit process, and the residual velocity is the velocity of the EFV when the evasion distance is attained. The calculation method of the discrete-time evasion distance $d_i(t_j)$ and residual velocity $v_E(t_j)$ is shown in Figure 2, where $d_i(t)$ is the relative distance between the EFV and ith PFV, and $v_E(t_j)$ is the EFV's velocity at the time instant i_k at which the conditions $d_i(t_j) \geq d_i(t_{j-1})$ and $d_i(t_{j-1}) > 0$ are satisfied.

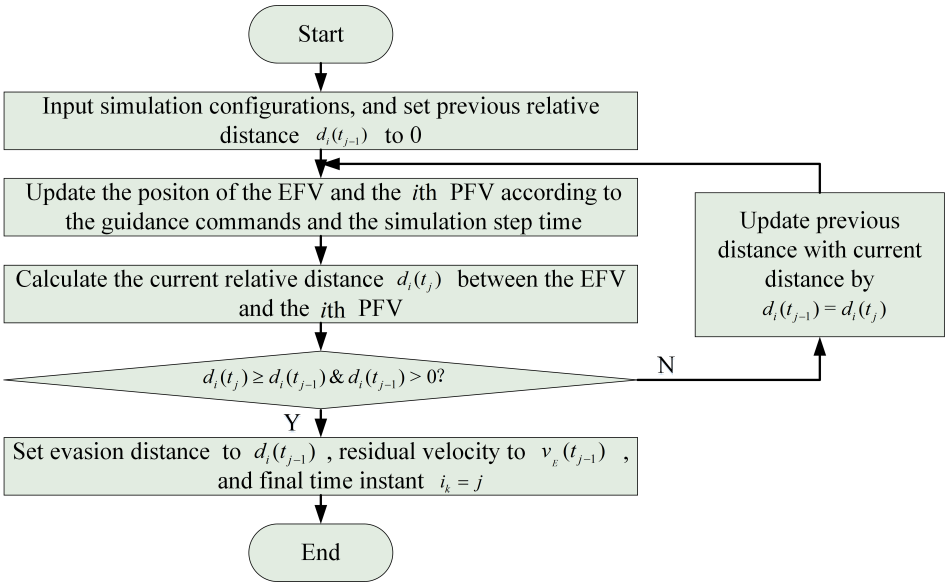


Figure 2. The calculation method of the evasion distance and the residual velocity.

2.3. Problem Formulation and Analysis

Each single step of simulating the escape-and-pursuit scenario composed of the EFV and up to three PFVs is described in Figure 3, and it consists of four major stages, i.e., Stage (1): The EFV generates its guidance commands based on its observation of the position and velocity of the PFVs at previous time instant, and it is imperative to note that the number of PFVs is variable, contingent upon whether the relative distance between the EFV and the *i*th PFV falls below the predefined threshold of 2000.0 m; Stage (2): According to the guidance commands generated by itself and its kinematics model, the EFV updates its own position and velocity at current time instant; Stage (3): The *i*th PFV generates its guidance commands based on its observation of the position and velocity of the EFV at previous time instant; Stage (4): Similarly, the *i*th PFV updates its position and velocity at current time instant according to the guidance commands generated by itself and its kinematics model.

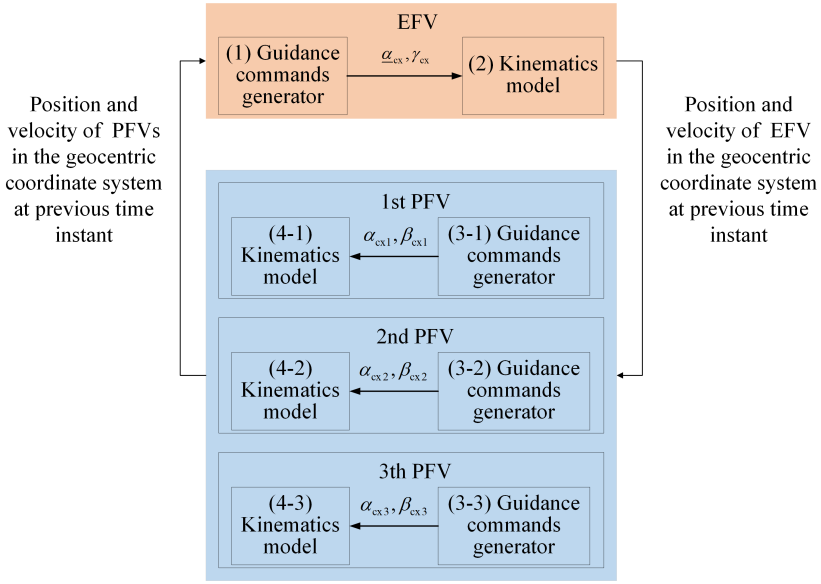
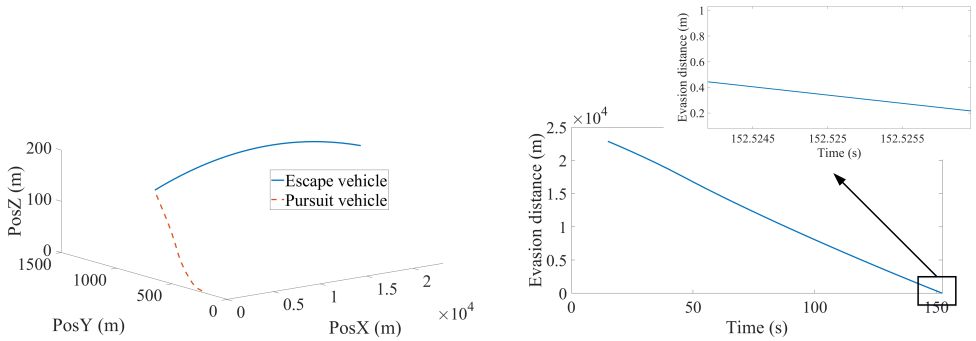


Figure 3. The iterative process of updating the positions and velocities of the EFV and the PFVs.

Based on the above description, we can obtain the following insights:

1. The kinematics model of the EFV is expressed by (1), which means that the position and velocity of the EFV are readily available if the output of Stage (1) has been determined.
2. According to its own position and velocity, the guidance commands of the ith PFV are readily available if the position and velocity of the EFV have been determined by executing the assumed proportional navigation method.
3. The kinematics model of the PFV is also expressed by (1). Therefore, the position and velocity of the ith PFV are readily available if the output of Stage (3) has been determined.
4. The only "independent variable" that can vary actively in every single step of simulating the escape-and-pursuit scenario, as illustrated by Figure 3, is the guidance command of the EFV, namely, $\underline{\alpha}_{cx}$ and γ_{cx}, which constitute the output of Stage (1).

Following the above analysis, we conduct a simulation study, where the EFV ignores the three PFVs and takes no measure to evade them. The simulation result between the EFV and the first PFV is shown in Figure 4. It can be observed from Figure 4a that the first PFV flies to the EFV directly, and from Figure 4b that the evasion distance is as small as 0.2 m, far less than 20.0 m, which means the EFV is captured by the first PFV completely. The parallel predicaments are observed in the interactions between the EFV and the other two PFVs (i.e., second and third PFV).



(**a**) (**b**)

Figure 4. The simulation result without the maneuvering of the EFV: (**a**) The flight trajectories of the vehicles; (**b**) The relative distance of the vehicles.

Therefore, we conclude that it is necessary for the EFV to take advantage of its maneuverability proactively to evade the PFVs, and it is essential to study the guidance design method of the EFV, in order to obtain the maximum residual velocity satisfying the evasion distance constraint. Based on the previous discussions and derivations, this problem can be formulated as

$$\begin{aligned} \max_{\underline{\alpha}_{cx}(t_i),\gamma_{cx}(t_i),n} \quad & v_E(t_{n-1}), \\ \text{s.t.} \quad & n = max(1_k, 2_k, 3_k), \\ & d_1(t_{1_k}) > 20.0 \text{ m}, \\ & d_2(t_{2_k}) > 20.0 \text{ m}, \\ & d_3(t_{3_k}) > 20.0 \text{ m}, \\ & \underline{\alpha}_{cx}(t_i) \in [-16.0^\circ, 16.0^\circ], \\ & \gamma_{cx}(t_i) \in [-90.0^\circ, 90.0^\circ], \\ & i \in [1, n]. \end{aligned} \tag{3}$$

In Poblem (3), 1_k, 2_k, 3_k are the time instant at which the evasion distance between the EFV and the corresponding PFV is attained according to Figure 2, and n is the maximum

value in the set $[1_k, 2_k, 3_k]$. It should be noted that if the ith PFV is not present in the scenario, its corresponding evasion distance time instant i_k defaults to 0, and its corresponding evasion distance $d_i(t_{i_k})$ defaults to 2000.0 m. As depicted in Figure 2, the specific time instant 1_k that satisfies $d_1(t_{1_k}) \geq d_1(t_{1_k-1})$ and $d_1(t_{1_k-1}) > 0$ is uncertain, because it is influenced by the guidance commands generated at the previous time instants by the EFV. Similarly, this uncertainty applies to 2_k and 3_k if the corresponding PFV exists in the scenario. Therefore, the process of analytically deriving an optimal solution for the problem (3) poses an enormous challenge.

3. The Proposed RNN-Based PPO Algorithm

In the traditional flight vehicle guidance designs, the input information is the state of the target, and the purpose of the generated guidance commands is to take the flight vehicle to the target continuously based on the state of the flight vehicle itself. For the proposed evasion guidance design, the input information of the EFV consists of the position and velocity of the PFVs, but the objective is to maximize the EFV's residual velocity, satisfying the constraint of the evasion distance. Therefore, it is difficult to adapt the traditional guidance design method to the escape-and-pursuit problem considered in this paper.

Guidance commands of flight vehicles can be regarded as a series of data sets having fixed time intervals, according to the assumptions in Section 2.1; thus, guidance design constitutes a typical sequential decision problem and satisfies the basic conditions for using the DRL technique. As discussed in Section 2.3, the only independent variable in every single step of simulating the escape-and-pursuit scenario is the output of Stage (1) in Figure 3, namely, the guidance commands of the EFV ($\underline{\alpha}_{\mathrm{cx}}$ and γ_{cx}). Because these guidance commands take their legitimate values from multi-dimensional continuous spaces, in principle, the family of policy gradient algorithms can be the appropriate candidates for the solving method. Policy gradient algorithms are generally divided into two categories, namely, the on-policy and the off-policy algorithms. The on-policy algorithms use a policy neural network to interact with the environment so that the training data can be generated, which is then utilized to update the policy neural network itself immediately. Therefore, in on-policy algorithms, the obtained training data can only be used once. As a result, typically on-policy algorithms require a longer training time than off-policy algorithms. On the other hand, during the initial period of the training process, the policy neural network of both the on-policy and off-policy algorithms may be updated dramatically, because significant differences can exist between the training data obtained at neighboring episodes (The significant difference is due to exploration in a huge action space with a policy neural network yet to be optimized.). In this case, it becomes difficult for the policy neural network to quickly find a good solution. To address this issue, the researchers from OpenAI proposed the PPO algorithm [26], which imposes constraints on the magnitude of the update carried out by the policy neural network. The PPO algorithm has been demonstrated effective in solving problems that are featured with multi-dimensional continuous action space, such as path planning [27,28], and swarm robots control [29]. In what follows, we will employ the PPO algorithm to solve the problem considered. Aiming to solve the proposed problem (3), the research design is structured as follows:

1. We design the framework of the proposed RNN-based PPO algorithm, utilizing the RNN to dynamically manage the dimensionality of the environment state that varies due to the different number of PFVs requiring evasion at different time instants in a single escape-and-pursuit simulation.
2. We design the structure of the training data by putting the hidden state of the RNN into it. Furthermore, we engineered a dual-layer batch method to adeptly manage the dimensional variances between environment states, enhancing both the stability and the efficiency of the training task.
3. We design the architecture of both the actor and critic networks by integrating RNN and FCNN. Furthermore, we propose two distinct training strategies depending on

whether the model is initialized using pre-trained weights from scenarios involving a smaller number of PFVs.

4. We design an elaborate reward function by creating a virtual escape-and-pursuit scenario, enabling rapid calculations of future evasion distance and residual velocity for generating current guidance commands of the EFV.

3.1. Design of the Interaction between the Agent and the Environment

Upon selecting the PPO algorithm, the crucial work is to design the interaction structure between the environment and the agent, which can generate the guidance commands of the EFV based on the information obtained from the environment. It is worth noting that the explicit inputs of Stage (1) in Figure 3 are the position and velocity of the PFVs, while the outputs of Stage (1) also rely on the position and velocity of the EFV itself implicitly. As shown in Section 2.1, both the position and velocity of the EFV and the ith PFV are described by six variables, namely, $r_{x,E}$, $r_{y,E}$, $r_{z,E}$, $v_{x,E}$, $v_{y,E}$, $v_{z,E}$, and r_{x,P_i}, r_{y,P_i}, r_{z,P_i}, v_{x,P_i}, v_{y,P_i}, v_{z,P_i}. An intuitive idea is to set the above 12 variables as the inputs of the agent directly. However, the absolute values of the position and velocity of the EFV and ith PFV are not really meaningful for the escape-and-pursuit problem considered. Using these absolute values may cause the agent to treat the absolute values as the feature of the problem mistakenly; thus, the generalization capability of the agent trained may be degraded. Therefore, it is a better alternative to set the relative position and relative velocity, totally six variables (i.e., $r_{x,E} - r_{x,P_i}$, $r_{y,E} - r_{z,P_i}$, $r_{z,E} - r_{z,P_i}$, $v_{x,E} - v_{x,P_i}$, $v_{y,E} - v_{y,P_i}$, $v_{z,E} - v_{z,P_i}$), as the inputs of the agent. As a result, the computational complexity can be reduced while improving the adaptability of the agent.

As discussed in Section 2.3, the significant feature in the proposed problem is the varying number of PFVs, attributed to (1) the variable total number of PFVs in the scenario; (2) the number of detected PFVs, whose relative distance to the EFV is less than 2000.0 m, is uncertain. Consequently, the standard PPO algorithm, training the agent with the FCNN alone, encounters significant obstacles owing to the prerequisite of a predetermined input layer dimensionality (i.e., the fixed dimensionality of the input data). Specifically, supposing the scenario's PFVs number up to three, the FCNN input layer's dimensionality is accordingly posited to be 3×6. When the detected number of PFVs is m ($m < 3$), the input layer's initial $m \times 6$ elements are fed with the relative state between the EFV and the respective PFVs, while the subsequent $(3 - m) \times 6$ elements can only be filled with 0 (or other predetermined values), bring more difficult in the training task.

The preceding analysis definitively clarifies that the core of solving the escape-and-pursuit problem, characterized by a varying number of PFVs, depends on the creation of a network architecture that is adaptable to dynamically varying data dimensionality. RNN, with the intrinsic ability to process variable-length data sequences, stands out as a feasible solution capable of meeting this challenge. Therefore, this paper introduces a novel RNN-based PPO algorithm to solve the problem as delineated in Figure 5. In this context, $\boldsymbol{X}_1$ is composed of six elements, specifically: $r_{x,E} - r_{x,P_1}$, $r_{y,E} - r_{z,P_1}$, $r_{z,E} - r_{z,P_1}$, $v_{x,E} - v_{x,P_1}$, $v_{y,E} - v_{y,P_1}$, $v_{z,E} - v_{z,P_1}$. Analogously, $\boldsymbol{X}_2$ and $\boldsymbol{X}_3$ share this structure and implication.

Based on Figure 5, we can obtain the following insights:

1. The environment is composed of a single EFV and up to three PFVs, with the agent integrated in the EFV capable of generating guidance commands (i.e., $\underline{\alpha}_{\text{cx}}$ and γ_{cx}) to evade the PFVs unremittingly.
2. At every interaction step, the state of the environment is represented by a sequence of nodes, where the number of nodes corresponds to the number of PFVs detected by the EFV. Each node is composed of six elements, denoting the relative position and velocity between the EFV and the corresponding PFV. Moreover, the states corresponding to PFVs detected earlier are prioritized and fed to the agent accordingly. Specifically, the state illustrated in Figure 5 reveals that the first PFV is detected initially, followed by the second, and finally, the third PFV.

3. In the proposed RNN-base PPO algorithm, both the actor and critic networks are constituted by a combination of the RNN and the FCNN. The actor network receives the state of the environment as input and generates the EFV's guidance commands as output. The critic network's input encapsulates the environment state, the actor network's action, and the reward feedback from the environment, resulting in an output that is the Q-value corresponding to the above information. The Q-value emerges as a pivotal metric instrumental in the parameter updating process in both the actor and critic networks.

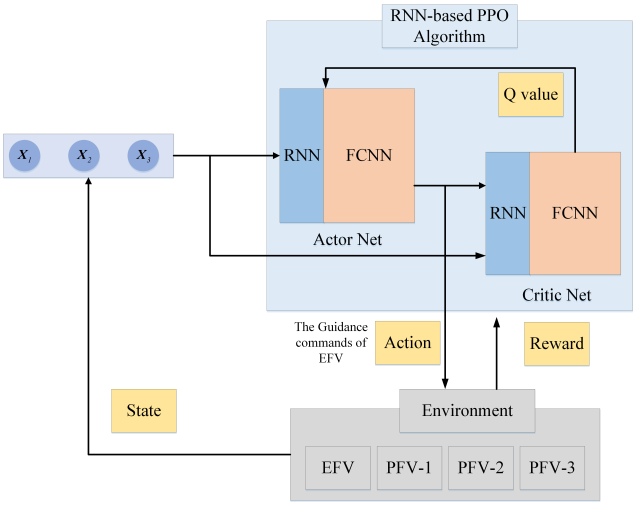


Figure 5. The RNN-based PPO algorithm structure.

3.2. Design of the Replay Buffer

As previously mentioned, the PPO algorithm stands as a quintessential example in the off-policy algorithm family, enabling the reuse of collected data for multiple training iterations of the agent. It is noteworthy that the training data are stored in a specialized memory structure, known as the replay buffer. In the conventional PPO algorithm, the training data's structure is delineated as (s_t, a_t, r_t, s_{t+1}), where s_t is the environment state at the current time instant, a_t is the action generated by the agent at the current time instant, r_t is the immediate reward obtained from the environment at the current time instant, and s_{t+1} is the environment state at next time instant following the execution of the action a_t.

It is widely recognized that the RNN excels in processing sequences of variable lengths, a skill fundamentally rooted in the management of the hidden state (denoted as h_t), which encapsulates the entirety of previous input information. With the introduction of the RNN to manage the varying number of PFVs requiring evasion, incorporating the hidden state into the training data becomes an evident necessity. Consequently, the training data's structure is augmented to $(s_t, a_t, r_t, s_{t+1}, h_t, h_{t+1})$, where h_t represents the hidden state at the current time instant, and h_{t+1} corresponds to the hidden state at the next time instant, as illustrated in Figure 6.

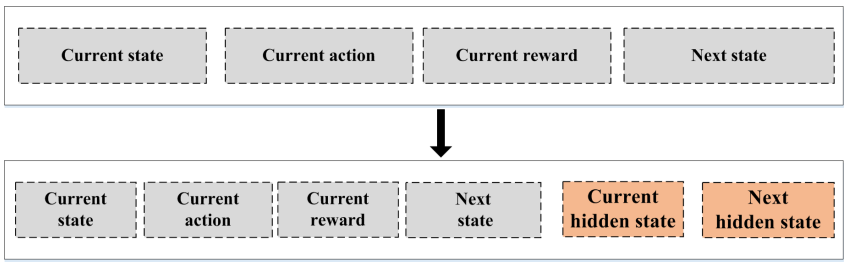


Figure 6. The structure of the training data.

Given the varying number of PFVs, the dimensionality of the environment state (s_t and s_{t+1}), which has a positive correlation with the exact number of PFVs, stored in the replay buffer changes accordingly. This variability hampers the ability to sample the replay buffer conventionally due to the different dimensionality of the data and can not be used to update the network's weights in a single operation. Although sampling the replay buffer for a single piece of data at a time and training the network's weights appears to be an alternative, the efficiency of this approach is notably suboptimal. To address this problem, we propose a dual-layer batch sampling approach, structured in two sequential stages. Initially, training data entries with the same dimensionality in the replay buffer are aggregated to form the outer layer batch. Subsequently, the inner layer batch undergoes random sampling, in line with the traditional DRL technique, as illustrated in Figure 7.

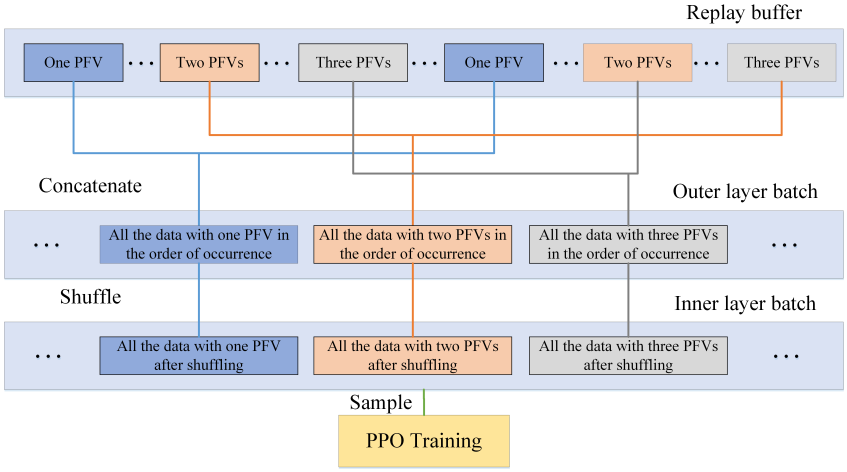


Figure 7. The dual-layer replay buffer structure.

3.3. Design of the Actor and Critic Networks

Following the proposed dual-layer batch approach, the training data obtained by the random sampling method can be directly employed for updating the weights in both the actor and critic networks, adhering to the conventional workflow of the PPO algorithm. The computational procedure of the actor network is delineated in Figure 8.

According to Figure 8, we can obtain the following insights:

1. The lower section of the figure illustrates the various environment states, where $\boldsymbol{X}_1$, $\boldsymbol{X}_2$, and $\boldsymbol{X}_3$ represent the relative state between the EFV and the first, second, and third PFVs, respectively. In the depicted scenario, the EFV initially evades the first PFV, subsequently evading with the second and third PFVs. Upon successfully evading the first PFV, only the relevant input information, namely, the relative state between the EFV and the second and third PFVs, is fed to the RNN for processing.
2. Drawing upon the step delineated in the figure, in the RNN computation phase, state nodes (i.e., $\boldsymbol{X}_1$, $\boldsymbol{X}_2$, $\boldsymbol{X}_3$) are sequentially fed into the RNN's input layer. This procedure yields a sequence of hidden states (i.e., h_1, h_2, and h_3), from which the hidden state located at the last position (h_3) is selected as the output of the RNN. Regarding the FCNN computation phase, the initial input constitutes the output derived from the RNN computation. This phase culminates in the generation of guidance commands for the EFV, serving as its output.

Similar to the actor network, the computational workflow of the critic network is organized into two discrete phases, each corresponding to the RNN and the FCNN, respectively. Given the RNN's capacity to handle varying numbers of PFVs utilizing the same structure, the model that has been initially trained in scenarios with fewer PFVs can proficiently act as an initial configuration for addressing more complex scenarios involving more PFVs. Consequently, we have delineated two distinct strategic approaches:

1. Begin the training task by randomly initializing the weights in both the actor and critic networks. This approach signifies the model's initiation without prior knowledge, enabling it to autonomously develop the evasion guidance strategy.
2. Implement an incremental learning strategy by loading the model's weights from previous scenarios. This approach enhances the model's proficiency in mastering the evasion guidance strategy, building upon the pre-learned similar knowledge.

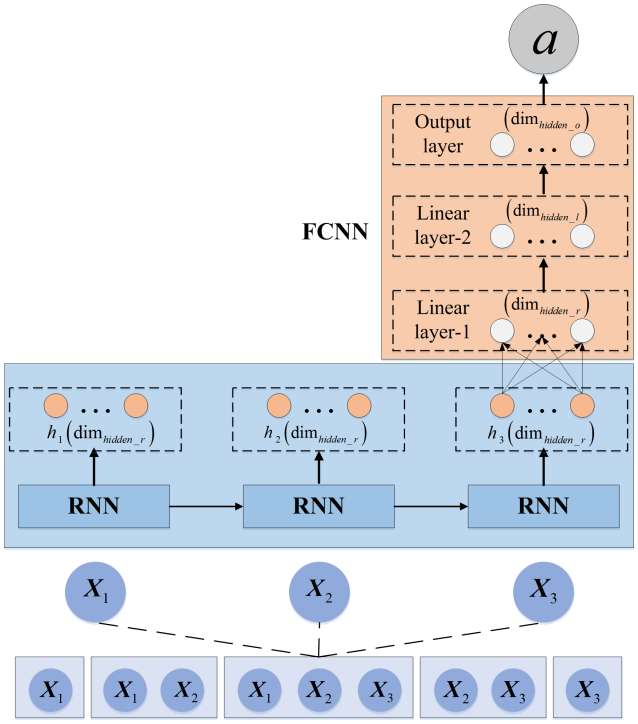


Figure 8. The computational procedure of the actor network.

3.4. Design of the Reward Function

In the DRL technique, the reward value obtained from the environment determines the direction of the optimization of the agent, and the training purpose is to make the agent steadily obtain the highest possible reward. Therefore, the reward function that can accurately characterize the effect of the action currently taken subject to the current state is very important. We propose a domain-knowledge-aided reward function expressed as

$$\begin{cases} R = \sum R_i, \\ R_i = \sum R_{p_i} + R_{f_i}, \\ i \in [1, 2, 3]. \end{cases} \tag{4}$$

where R is the total reward of a single training episode, R_i is the total reward corresponding to the ith PFV, R_{p_i} is the immediate reward of every single step about the ith PFV, and R_{f_i} is the reward of the final step about ith PFV, when the evasion distance between the EFV and the ith PFV is attained. It should be noted that if the ith PFV does not exist in the scenario, the corresponding reward R_i is set to 0. Since the goal of the optimization is to maximize the residual velocity satisfying the evasion distance constraint, the final step reward R_{f_i} is expressed as

$$R_{f_i} = K_{v_i} \times v_E(t_{i_k-1}) + K_{d_i} \times 20.0, \tag{5}$$

where K_{v_i} equals 10.0 when the evasion distance $d(t_{i_k-1})$ exceeds 20.0 m (i.e., the assumed safe evasion distance), otherwise it is set to 0, and $v_E(t_{i_k-1})$ represents the residual velocity

of the EFV at the time instant of t_{i_k-1} when the conditions (i.e., $d(t_{i_k}) \geq d(t_{i_k-1})$ and $d(t_{i_k-1}) > 0$) are satisfied in Figure 2. In addition, we have

$$K_{d_i} = \text{clip}(d(t_{i_k-1})/20.0, 0, 1), \tag{6}$$

where $\text{clip}(x, \min, \max)$ outputs x if $\min \leq x \leq \max$, outputs min if $x < \min$, and outputs max if $x > \max$. Hence K_{d_i} is limited to the range of $[0, 1]$.

It is obvious that it is very hard to learn from the sparse reward (i.e., the total reward R_i becomes R_{f_i} when setting R_{p_i} to 0; then the agent is trained with R_{f_i} alone). Therefore, we proposed the calculation method of R_{p_i}, which embodies a function reflecting both the prospective residual velocity and the prospective evasion distance. Both of them can be achieved by keeping the current guidance commands generated by the agent of the EFV fixed, until the end of the *virtual escape-and-pursuit scenario* created, as described in Figure 9.

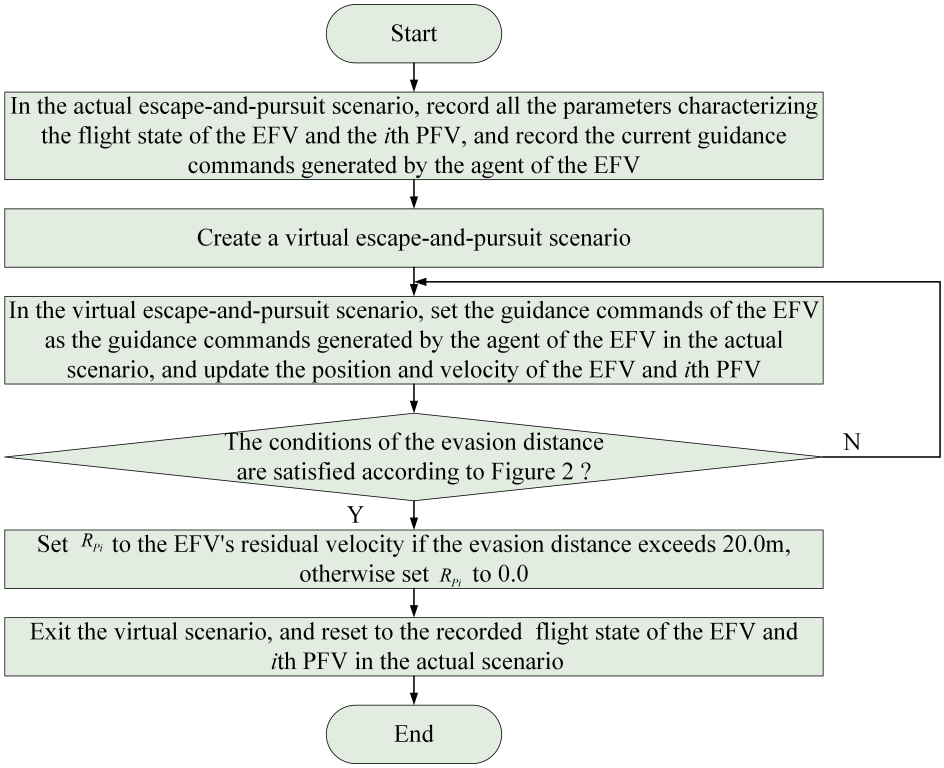


Figure 9. The procedure of calculating the prospective residual velocity and the prospective evasion distance in a virtual escape-and-pursuit scenario.

4. Simulation Results and Discussions

To meticulously evaluate the effectiveness of the RNN-based PPO algorithm in addressing the proposed escape-and-pursuit problem, characterized by a varying number of PFVs, we design two distinct kinds of simulation experiments. The first kind of simulation experiment comprises three distinct scenarios, with the number of PFVs sequentially set at one, two, and three. The second kind of simulation experiments are engineered to benchmark the performance of the RNN-based PPO algorithm against that of the FCNN-based PPO algorithm, providing a comparative analysis. The detailed results of these simulation experiments are outlined as follows.

4.1. Training Result of the RNN-Based PPO Algorithm in the Designed Three Scenarios

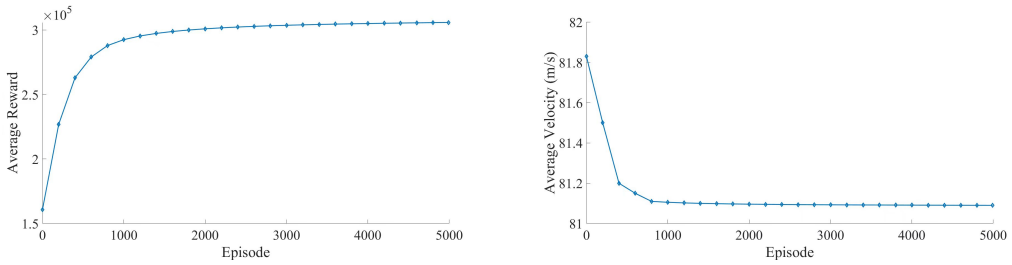
As previously outlined, the EFV initiates evasion maneuvers against the *i*th PFV when the relative distance decreases to below 2000.0 m, continuing until the evasion distance is attained according to Figure 2. It is necessary to emphasize that in scenarios where the number of PFVs exceeds one, a concurrent application of multiple evasion distance

judgment criteria is employed. Specifically, if any of the evasion distances relative to the PFVs fall below 20.0 m, the escape attempt is deemed unsuccessful, leading to the termination of the training episode. The training episode is only considered successful when the EFV manages to successfully evade all PFVs present in the scenario.

In the first phase of our research, we engaged in the training of the EFV's agent in a scenario characterized by the presence of one PFV. Given the limited computational resources of the EFV, our objective is to minimize the parameter scale in the designs of both the actor and critic networks. Compared to the FCNN, the RNN exhibits a larger parameter scale, which positively correlates with the number of RNN's layers. Consequently, we have opted to set the number of RNN layers to one to efficiently manage the computational demands. Regarding the additional hyperparameters, such as the dimension of the RNN's hidden state, the number of FCNN layers, and the learning rate, we employed the standard parameters that our research group utilizes for comparable escape-and-pursuit problems. The specific values of these hyperparameters are delineated in Table 1, and the corresponding outcomes are illustrated in Figure 10.

Table 1. The hyperparameters set used in the training process for the single PFV.

Parameter	Meaning	Value
l_R	The number of the layer in the RNN	1
R_D	The dimensionality of the hidden state	256
l_F	The number of the layer in the FCNN	3
F_D	The number of the nodes in each hidden layer in the FCNN	256
lr	The learning rate	1×10^{-4}



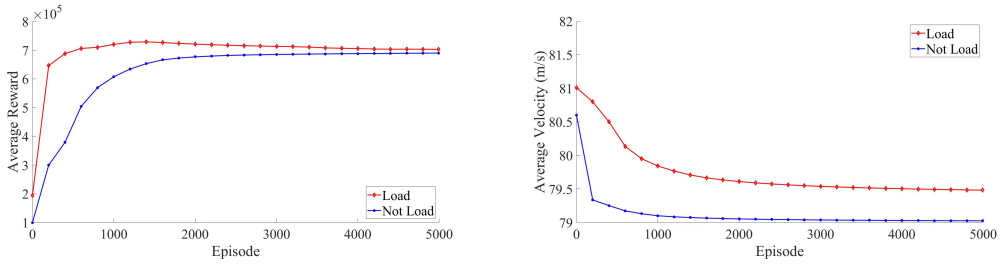
(**a**) (**b**)

Figure 10. The training result of the scenario characterized by one PFV: (**a**) The curve of the episode's reward; (**b**) The curve of the episode's residual velocity.

According to the results in Figure 10, the following observations are obtained:

1. As illustrated in Figure 10a, there is a consistent augmentation in the episode rewards, which serves as a testament to the EFV's adept evasion of the PFV, facilitated by the judicious application of the designated reward function and hyperparameters. This trend unequivocally underscores the successful execution of training tasks, as evaluated from the perspective of the DRL technique.
2. Figure 10b elucidates a discernible downward trend in residual velocity concomitant with the increment in episode number. It is imperative to acknowledge that the EFV's energy reserves are inherently limited, with a significant allocation dedicated to modifying its flight trajectory to evade the PFV. This strategic energy deployment engenders a consistent reduction in residual velocity as the evasion distance increases.

In the second phase of our research, we proceeded to train the EFV's agent in a scenario that entails the evasion from two PFVs, applying the specific hyperparameters as systematically outlined in Table 1. The training results based on the two initial strategies as described in Section 3.3 are depicted in Figure 11.

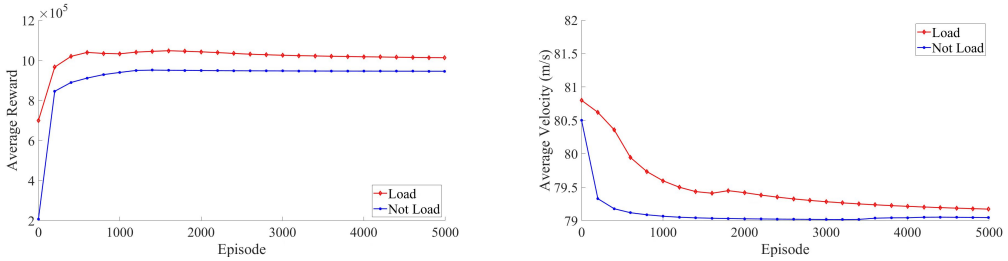


(**a**) (**b**)

Figure 11. The training result of the scenario characterized by two PFVs: (**a**) The curve of the episode's reward; (**b**) The curve of the episode's residual velocity.

From Figure 11, it is apparent that both training strategies converge to a similar solution, indicating that the RNN-based PPO algorithm can consistently train the agent in the EFV across diverse initial conditions. Furthermore, the loading strategy outperforms its counterpart in terms of episode reward and residual velocity, notably in convergence speed and terminal values. This demonstrates that utilizing pre-learned knowledge to enhance effectiveness in addressing the escape-and-pursuit problem is an effective strategy.

In the third phase of our research, we furthered the training of the EFV's agent in a scenario involving three PFVs, utilizing the hyperparameters detailed in Table 1. Consistent strategies were adopted in this scenario in the same way in the previous scenario with two PFVs, leading to similar insights, which are elucidated in Figure 12.



(**a**) (**b**)

Figure 12. The training result of the scenario characterized by three PFVs: (**a**) The curve of the episode's reward; (**b**) The curve of the episode's residual velocity.

Beyond the aforementioned analyses, we explore the correlation between the residual velocity of the EFV and the number of PFVs in the scenarios. As shown in Figure 13, a negative correlation is evident between the residual velocity and the number of PFVs requiring evasion. This trend is attributable to the increased energy consumption for evading a greater number of PFVs, leading to the reduced residual velocity. This phenomenon is further explained with reference to Figure 14. An increased number of PFVs correlates with longer evasion duration (i.e., t_1, t_2, and t_3 represent the precise time instants when the evasion distances for the first, second, and third PFVs are respectively attained), thereby leading to increased energy consumption. Given the limited total energy of the EFV, greater energy consumption on evasion inversely affects the residual velocity. Additionally, Figure 15 graphically illustrates the reduction in evasion distances between the EFV and the three PFVs, which decrease from 2000.0 m to their respective terminal values across various time instants.

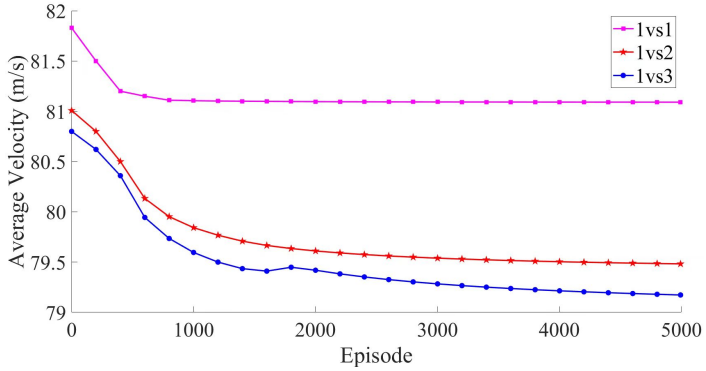
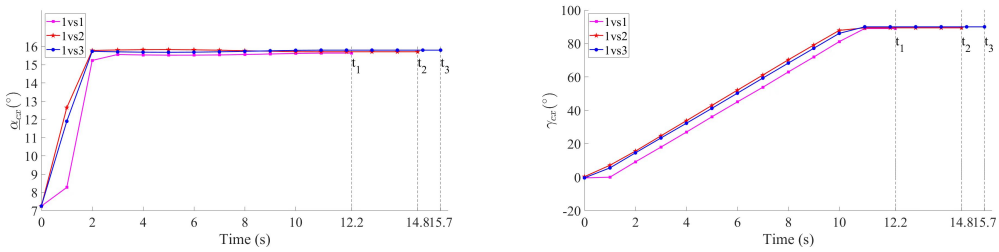


Figure 13. The comparison of the residual velocity among the three scenarios.



(a) **(b)**

Figure 14. The curve of the guidance commands: (**a**) The curve of α_{cx}; (**b**) The curve of γ_{cx}.

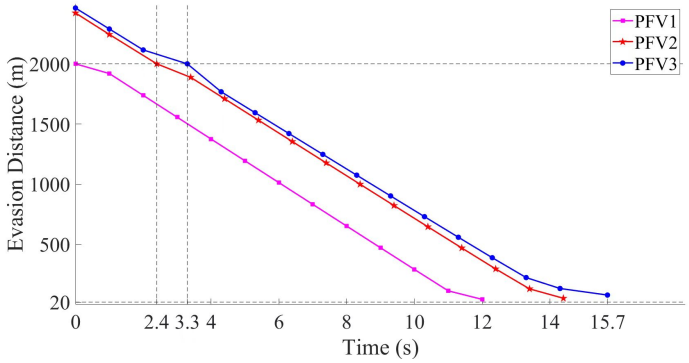


Figure 15. The evasion distances between the EFV and the three PFVs.

4.2. Comparative Analysis of the RNN-Based PPO Algorithm and The Conventional FCNN-Based PPO Algorithm

This section details the design of comparative simulation experiments between the RNN-based PPO algorithm and the conventional FCNN-based PPO algorithm. Considering the requirement to predefine the dimensionality of the FCNN's input layer, it is established as $6 \times i_P$, where 6 represents the relative state between the EFV and ith PFV, and i_P represents the total number of PFVs in the scenarios. As models trained for different numbers of PFVs possess distinct architectures, the training strategy that incorporates pre-learned knowledge, as detailed in Section 3.3, cannot be applied to agents exclusively using FCNN. To conduct a comparative evaluation of the effectiveness of the RNN-based PPO algorithm versus the FCNN-based PPO algorithm, the three scenarios described in Section 4.1 are replicated using the FCNN-based PPO algorithm to train the evasion models. The results of this comparative analysis are comprehensively depicted in Figure 16.

Comparative analysis across all scenarios indicates that although both the RNN-based and FCNN-based PPO algorithms achieve convergence, the RNN-based algorithm consistently surpasses its conventional counterpart in efficiency and terminal values. The

reason for this superior performance is delineated as follows. Given that FCNN are constrained to processing input data of fixed dimensionality, the RNN's input layer nodes are configured to $6 \times i_P$, where i_P represents the total number of PFVs. As analyzed in Section 3.3, in the same escape-and-pursuit simulation, the number of PFVs requiring evasion varies sequentially over time—specifically, $1, \ldots, (i_P - 1), i_P, (i_P - 1), \ldots, 1$. During the training of the RNN-based PPO algorithm, states corresponding to the exact number of PFVs requiring evasion are fed into the RNN. Conversely, in the FCNN-based PPO algorithm's training, a zero-padding method is employed to fill the input layer nodes if the PFVs are not detected by the EFV or have been successfully evaded. Consequently, the RNN-based PPO algorithm exhibits superior performance.

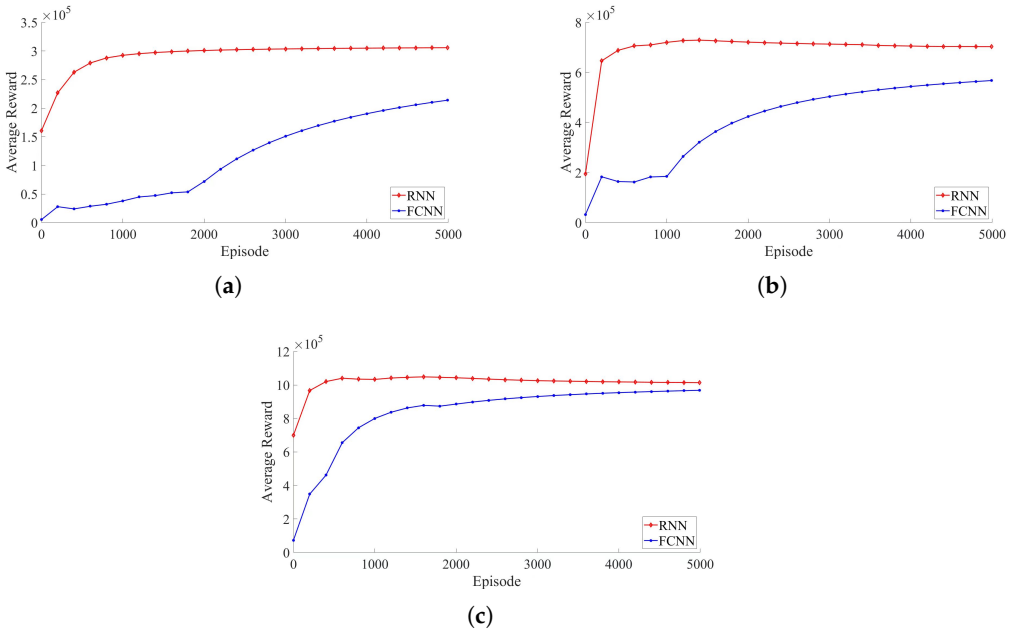


Figure 16. The comparison of the reward between the proposed RNN-based PPO algorithm and the conventional PPO algorithm: (**a**) The reward in one PFV scenario; (**b**) The reward in two PFV scenarios; (**c**) The reward in three PFV scenarios.

To further emphasize the superiority of the RNN-based PPO algorithm over the FCNN-based algorithm, we deployed the model trained in a scenario involving a single PFV and applied it to the scenarios featured by two and three PFVs. The results of this simulation are detailed in Table 2. According to Table 2, the model trained using the RNN-based algorithm successfully evades all three PFVs, whereas the model trained using the FCNN-based algorithm manages to evade only the first PFV. The primary reason is elucidated as follows. In the FCNN mode, the relative states between the EFV and the second and third PFV cannot be fed to the agent, as the FCNN's input layer dimensionality is fixed at 6, which is filled by the relative state of the EFV and the first PFV. Conversely, in the RNN mode, the relative states between the EFV and all three PFVs can be sequentially fed to the agent, enabling successful evasion. These simulation results demonstrate that the model trained with the proposed RNN-based PPO algorithm can handle scenarios not encountered during training, indicating excellent generalizability and the capability to manage scenarios involving more than three PFVs.

Table 2. The simulation result of evading the PFVS in an unknown scenario with pre-trained model.

The Total Number of the PFV in the Scenario	The Index of the PFV	The Evasion Distance of the RNN-Based PPO Algorithm (m)	The Evasion Distance of the FCNN-Based PPO Algorithm (m)
1	1	20.76 (>20.0, success)	25.41 (>20.0, success)
2	1	22.47 (>20.0, success)	25.41 (>20.0, success)
	2	34.22 (>20.0, success)	10.32 (<20.0, fail)
3	1	26.72 (>20.0, success)	25.41 (>20.0, success)
	2	37.27 (>20.0, success)	10.32 (<20.0, fail)
	3	42.22 (>20.0, success)	3.12 (<20.0, fail)

4.3. Improvement for Future Work

The effectiveness of the proposed RNN-based PPO algorithm is evidenced by the results of the simulation experiments outlined above. Two aspects of the algorithm can be further improved:

1. It is posited that the EFV and PFV can obtain each other's positions and velocities continuously, accurately, and instantaneously. This assumption simplifies the problem to a certain extent. Future efforts will concentrate on refining the algorithm through incremental training involving intermittent, erroneous, and delayed data to enhance the evasion model's adaptability.
2. Given the limited computational resources available on the EFV, the RNN was selected to facilitate the training of the algorithm, rather than the transformer model, which is prevalent in contemporary artificial intelligence research. Our future work will explore the compatibility of embedded intelligent processors with the transformer model and aim to replace the current RNN with the transformer to enhance the algorithm's adaptability.
3. This paper primarily investigates an intelligent evasion model designed for scenarios in which a single EFV evades multiple PFVs. There is a potential risk that the intelligent evasion model might underperform in complex scenarios where multiple EFVs collaboratively evade multiple PFVs. The primary reason for this is that in such scenarios, each EFV must effectively evade multiple PFVs while simultaneously avoiding collisions with fellow EFVs. This scenario constitutes a multi-agent joint reinforcement learning challenge, extending beyond the single-agent reinforcement learning framework addressed in this paper. Our future work will involve related research into these more interesting and challenging problems.

5. Conclusions

In this paper, we have considered the escape-and-pursuit scenario involving a single EFV and up to three PFVs, aiming to maximize the EFV's residual velocity under the constraint of the evasion distance threshold. We assume that the EFV generates guidance commands with the aid of the DRL technique, while each PFV uses the conventional proportional navigation method. We reveal that, in general, it is difficult to find the analytical solution to the residual velocity maximization problem because of the varying number of PFVs. The guidance design problem considered constitutes a typical sequential decision problem, and the results of the decision in each step are from a multi-dimensional continuous space, making the PPO algorithm an appropriate choice. However, the conventional PPO algorithm, when employing the FCNN alone, can not manage the varying number of PFVs requiring evasion. Consequently, we propose the RNN-based PPO algorithm and design the replay buffer, the actor and critic networks, and the reward function skillfully. Comprehensive simulation results robustly validate the effectiveness of the guidance design method supported by the innovative RNN-based PPO algorithm, particularly when utilizing pre-trained models developed with fewer PFVs. Additionally, comparative

analyses with the FCNN-based PPO algorithm further confirm the superior adaptability and enhanced performance of the RNN-based PPO algorithm, even in scenarios with a predefined number of PFVs.

Author Contributions: Conceptualization, X.H. and H.W.; methodology, X.H. and T.W.; software, X.H.; validation, M.G.; formal analysis, M.G.; investigation, X.H.; resources, M.G.; data curation, X.H.; writing—original draft preparation, X.H.; writing—review and editing, T.W. and M.G.; visualization, M.G.; supervision, H.W. and T.W.; project administration, X.H.; funding acquisition, M.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data are contained within the article.

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

References

1. Garcia, E.; Casbeer, W.D.; Pachter, M. Design and Analysis of State Feedback Optimal Strategies for the Differential Game of Active Defense. *IEEE Trans. Autom. Control* **2019**, *64*, 553–568. [CrossRef]
2. Sinha, A.; Kumar, S.R.; Mukherjee, D. Nonsingular Impact Time Guidance and Control Using Deviated Pursuit. *Aerosp. Sci. Technol.* **2021**, *115*, 106776. [CrossRef]
3. Cheng, L.; Jiang, F.H.; Wang, Z.B.; Li, J.F. Multiconstrained Real-time Entry Guidance Using Deep Neural Networks. *IEEE Trans. Aerosp. Electron. Syst.* **2021**, *57*, 325–340. [CrossRef]
4. Peng, C.; Zhang, H.W.; He, Y.X.; Ma, J.J. State-Following-Kernel-Based Online Reinforcement Learning Guidance Law against Maneuvering Target. *IEEE Trans. Aerosp. Electron. Syst.* **2022**, *58*, 5784–5797. [CrossRef]
5. Shalumov, V. Cooperative Online Guide-launch-guide Policy in a Target-missile-defender Engagement Using Deep Reinforcement Learning. *Aerosp. Sci. Technol.* **2020**, *104*, 105996. [CrossRef]
6. Liu, X.D.; Li, G.F. Adaptive Sliding Mode Guidance with Impact Time and Angle Constraints. *IEEE Access* **2020**, *8*, 26926–26932. [CrossRef]
7. Zhou, J.L.; Yang, J.Y. Distributed Guidance Law Design for Cooperative Simultaneous Attacks with Multiple Missiles. *J. Guid. Control Dyn.* **2016**, *39*, 2436–2444. [CrossRef]
8. Zhai, C.; He, F.H.; Hong, Y.G.; Wang, L.; Yao, Y. Coverage-based Interception Algorithm of Multiple Interceptors against the Target Involving Decoys. *J. Guid. Control Dyn.* **2016**, *39*, 1647–1653. [CrossRef]
9. Liang, Z.X.; Ren, Z. Tentacle-Based Guidance for Entry Flight with No-Fly Zone Constraint. *J. Guid. Control Dyn.* **2018**, *41*, 991–1000. [CrossRef]
10. Liang, Z.X.; Liu, S.Y.; Li, Q.D.; Ren, Z. Lateral Entry Guidance with No-Fly Zone Constraint. *Aerosp. Sci. Technol.* **2017**, *60*, 39–47. [CrossRef]
11. Zhao, D.J.; Song, Z.Y. Reentry Trajectory Optimization with Waypoint and No-Fly Zone Constraints Using Multiphase Convex Programming. *Acta Astronaut.* **2017**, *137*, 60–69. [CrossRef]
12. Zhou, Q.H.; Liu, Y.F.; Qi, N.M.; Yan, J.F. Anti-warning Based Anti-interception Avoiding Penetration Strategy in Midcourse. *Acta Aeronaut. Astronaut. Sin.* **2017**, *38*, 319922.
13. Yu, W.B.; Chen, W.C.; Jiang, Z.G.; Zhang, W.Q.; Zhao, P.L. Analytical Entry Guidance for Coordinated Flight with Multiple No-fly-zone Constraints. *Aerosp. Sci. Technol.* **2019**, *84*, 273–290. [CrossRef]
14. Yan, T.; Cai, Y.L.; Xu, B. Evasion Guidance Algorithms for Air-breathing Hypersonic Vehicles in Three-player Pursuit-evasion Games. *Chin. J. Aeronaut.* **2020**, *33*, 3423–3436. [CrossRef]
15. Wang, Y.Q.; Ning, G.D.; Wang, X.F. Maneuver Penetration Strategy of Near Space Vehicle Based on Differential Game. *Acta Aeronaut. Astronaut. Sin.* **2020**, *41*, 724276.
16. Shen, Z.P.; Yu, J.L.; Dong, X.W.; Hua, Y.Z.; Ren, Z. Penetration Trajectory Optimization for the Hypersonic Gliding Vehicle Encountering Two Interceptors. *Aerosp. Sci. Technol.* **2022**, *121*, 107363. [CrossRef]
17. Nath, S.; Ghose, D. Worst-Case Scenario Evasive Strategies in a Two-on-One Engagement Between Dubins' Vehicles With Partial Information. *IEEE Control Syst. Lett.* **2022**, *7*, 25–30. [CrossRef]
18. He, S.M.; Shin, H.S.; Tsourdos, A. Computational Missile Guidance: A Deep Reinforcement Learning Approach. *J. Aerosp. Inf. Syst.* **2021**, *18*, 571–582. [CrossRef]
19. Jiang, L.; Nan, Y.; Zhang, Y.; Li, Z. Anti-Interception Guidance for Hypersonic Glide Vehicle: A Deep Reinforcement Learning Approach. *Aerospace* **2022**, *9*, 424. [CrossRef]
20. Shen, Z.P.; Yu, J.L.; Dong, X.W.; Ren, Z. Deep Neural Network-based Penetration Trajectory Generation for Hypersonic Gliding Vehicles Encountering Two Interceptors. In Proceedings of the 2022 41st Chinese Control Conference (CCC), Hefei, China, 25–27 June 2022; pp. 3392–3397.
21. Guo, Y.H.; Jiang, Z.J.; Huang, H.Q.; Fan, H.J.; Weng, W.Y. Intelligent Maneuver Strategy for a Hypersonic Pursuit-Evasion Game Based on Deep Reinforcement Learning. *Aerospace* **2023**, *10*, 783. [CrossRef]

22. Hui, J.P.; Wang, R.; Yu, Q.D. Generating New Quality Flight Corridor for Reentry Aircraft Based on Reinforcement Learning. *Acta Aeronaut. Astronaut. Sin.* **2022**, *9*, 325960.
23. Pham, D.H.; Lin, C.M.; Giap, V.N.; Cho, H.Y. Design of Missile Guidance Law Using Takagi-Sugeno-Kang (TSK) Elliptic Type-2 Fuzzy Brain Imitated Neural Networks. *IEEE Access* **2023**, *11*, 53687–53702. [CrossRef]
24. Pham, D.H.; Lin, C.M.; Huynh, T.T.; Cho, H.Y. Wavelet Interval Type-2 Takagi-Kang-Sugeno Hybrid Controller for Time-series Prediction and Chaotic Synchronization. *IEEE Access* **2022**, *10*, 104313–104327. [CrossRef]
25. Qian, X.F. *Missile Flight Aerodynamics*; Beijing Institute of Technology Press: Beijing, China, 2014.
26. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal Policy Optimization Algorithms. Available online: http://arxiv.org/abs/1707.06347 (accessed on 28 August 2017).
27. Qi, C.Y.; Wu, C.F.; Lei, L.; Li, X.L.; Cong, P.Y. UAV Path Planning Based on the Improved PPO Algorithm. In Proceedings of the 2022 Asia Conference on Advanced Robotics, Automation, and Control Engineering (ARACE), Qingdao, China, 26–28 August 2022; pp. 193–199.
28. Xiao, Q.H.; Jiang, L.; Wang, M.M.; Zhang, X. An Improved Distributed Sampling PPO Algorithm Based on Beta Policy for Continuous Global Path Planning Scheme. *Sensors* **2023**, *23*, 6101. [CrossRef]
29. Tan, Z.Y.; Karaköse, M. Proximal Policy Based Deep Reinforcement Learning Approach for Swarm Robots. In Proceedings of the 2021 Zooming Innovation in Consumer Technologies Conference (ZINC), Novi Sad, Serbia, 26–27 May 2021; pp. 166–170.