*electronics*

MDPI

*Article*

# Accelerating Fuzzy Actor–Critic Learning via Suboptimal Knowledge for a Multi-Agent Tracking Problem

**Xiao Wang [1,†], Zhe Ma [2,3,†], Lei Mao [2,3], Kewu Sun [2,3], Xuhui Huang [2,3], Changchao Fan [4] and Jiake Li [2,3,5,*]**

[1] College of Information Science and Technology, Beijing University of Chemical Technology, Beijing 100029, China
[2] Intelligent Science & Technology Academy Limited of CASIC, Beijing 100043, China
[3] Key Lab of Aerospace Defense Intelligent System and Technology, Beijing 100043, China
[4] The Second Academy of CASIC, Beijing 100854, China
[5] National Innovation Institute of Defense Technology, Academy of Military Sciences, Beijing 100071, China
* Correspondence: lijiake1223@163.com
† These authors contributed equally to this work.

**Abstract:** Multi-agent differential games usually include tracking policies and escaping policies. To obtain the proper policies in unknown environments, agents can learn through reinforcement learning. This typically requires a large amount of interaction with the environment, which is time-consuming and inefficient. However, if one can obtain an estimated model based on some prior knowledge, the control policy can be obtained based on suboptimal knowledge. Although there exists an error between the estimated model and the environment, the suboptimal guided policy will avoid unnecessary exploration; thus, the learning process can be significantly accelerated. Facing the problem of tracking policy optimization for multiple pursuers, this study proposed a new form of fuzzy actor–critic learning algorithm based on suboptimal knowledge (SK-FACL). In the SK-FACL, the information about the environment that can be obtained is abstracted as an estimated model, and the suboptimal guided policy is calculated based on the Apollonius circle. The guided policy is combined with the fuzzy actor–critic learning algorithm, improving the learning efficiency. Considering the ground game of two pursuers and one evader, the experimental results verified the advantages of the SK-FACL in reducing tracking error, adapting model error and adapting to sudden changes made by the evader compared with pure knowledge control and the pure fuzzy actor–critic learning algorithm.

**Keywords:** suboptimal knowledge; fuzzy system; actor–critic; Apollonius circle



**Citation:** Wang, X.; Ma, Z.; Mao, L.; Sun, K.; Huang, X.; Fan, C.; Li, J. Accelerating Fuzzy Actor–Critic Learning via Suboptimal Knowledge for a Multi-Agent Tracking Problem. *Electronics* **2023**, *12*, 1852. https://doi.org/10.3390/electronics12081852

Academic Editor: Alberto Fernandez Hilario

Received: 14 February 2023
Revised: 3 April 2023
Accepted: 11 April 2023
Published: 13 April 2023



## 1. Introduction

In the real world, it is a widespread phenomenon that predators have to hunt larger or faster prey. This hunting phenomenon can be naturally generalized to the field of robotics and control, where multiple slower robots (pursuers) try to capture one faster target (evader) who, conversely, attempts to escape. Theoretically, this is known as multi-player pursuit–evasion games with one superior evader [1]. Here, the superior evader signifies that the evader has comparatively more advantageous control resources than the pursuers [2]. The multi-player pursuit–evasion game is a common model in differential games that has been studied by many researchers during recent decades [3]. Facing the pursuit–evasion problem with multiple guided missiles, an optimal-damage-effectiveness cooperative–control strategy was proposed [4]. Considering a fixed duration differential game problem with Grönwall-type constraints, the players' attainability domain and optimal strategies were constructed [5]. Aiming at the problem of high dimensionality and high dynamics in the cluster confrontation game, an evolution strategies optimization method was proposed [6]. To deal with the major threat to public safety, as well as critical

infrastructure security caused by unmanned aircraft vehicles, a multi-agent jamming system was presented [7].

In order to solve the problem of multi-agent pursuit and evasion, it is necessary to study the modeling and control method of a multi-agent system (MAS) [8,9]. At present, a multi-agent system generally refers to a system composed of a group of agents with certain autonomous abilities [10]. Each agent in the system has the abilities of perception, cognition, decision-making, execution, self-organization, learning and reasoning [11]. In MAS, one of the most popular research objects is multi-agent distributed formation control [12]. Formation means that by designing a communication topology network and distributed controller, the state of networked agents can follow the desired formation configuration and maintain or adjust the configuration over time to meet the needs of multi-agent practical tasks. Aiming at the problem of a formation control problem without collisions, a control strategy that consists of a bounded attractive component was proposed [13]. Based on algebraic graph theory and rigid graph theory, the interconnections between vehicles in formation and the inter-vehicle distance constraints of the desired formation can be described [14]. For the problem of the controllability of leader–follower multi-agent systems, the upper bound on the controllability index was discussed [15]. For the robust control problem of aerial-refueling UAV close formation systems, a distributed formation control method based on adaptive disturbance observers with the barrier function was attempted [16]. To provide a fixed-time tracking consensus, a consensus protocol based on the integral sliding mode surface was presented [17].

With the rise of artificial intelligence technology, reinforcement learning (RL) has shown unprecedented potential in the field of agent decision-making [18]. Reinforcement learning is a kind of feedback-based learning, that is, there is an agent that can perceive the environment, act according to the environment's state and receive feedback information from the environment to adjust its action policy. Nowadays, RL is widely used in the fields of industrial automation [19], competitive games, autonomous detection and so on [20,21]. At present, reinforcement learning algorithms mainly include TD learning, Q-learning, SARSA and actor–critic [22]. However, in the field of practical applications, RL is facing the challenges of low efficiency and hard convergence due to the large system state and decision-making state spaces [23]. In order to improve the practical application effect of reinforcement learning, there are factorial reinforcement learning (FRL) [24], hierarchical reinforcement learning (HRL) [25], inverse reinforcement learning (IRL) [26], deep reinforcement learning (DRL), etc. [27].

Benefitting from the rapid development of RL, it was gradually applied to solve multi-agent system problems. At first, game theory was introduced into MAS and the concept of learning was proposed, which was called the minimax-Q algorithm [28]. Based on this algorithm, many derivative types were studied, such as Nash Q-learning [29], which was mainly used to solve zero-sum differential games. However, if the knowledge structure of the problem is not clear enough or the number of agents is large, the problem is difficult. Therefore, since 2010, researchers have turned their attention to deep reinforcement learning technology, which contains the networked policy and state [30,31]. By expanding the algorithm of deep deterministic policy gradient (DDPG) to MAS, the multi-agent DDPG (MADDGP) was proposed [32]. This is a framework that adopted centralized training and decentralized execution. Then, a counterfactual benchmark was added to the actor–critic framework to solve the credit allocation problem in MAS [33]. Furthermore, to apply the attention mechanism of the shared parameters, the agent was enabled to learn more effectively in complex multi-agent environments [34]. Benefitting from the technique of deep reinforcement learning, policy optimizations can be conducted using a data-driven method [35]. To realize the cooperative UAV formation, a multi-agent reinforcement learning algorithm with heuristic functions was proposed. Through employing the policy of centralized training with decentralized execution, an improved MADDPG was proposed to evaluate the value function more accurately in a UAV cluster [36]. To allow for real applications of the multi-agent RL technique, a timing recovery loop for

PSK and QAM modulations based on swarm reinforcement learning were proposed for high-speed telecommunications systems [37,38].

Based on the review of the abovementioned literature, it was found that the solution of multi-agent policy optimization can be divided into knowledge-driven and data-driven methods. Both methods have their advantages and disadvantages. For the knowledge-driven method, the advantages lie in good stability and low computational complexity, while the disadvantages lie in the strong dependence on the model information. Meanwhile, for the data-driven method, the advantage lies in relaxing the dependence on the model information, while the disadvantage lies in the large amount of computation. Therefore, the focus of this study was to combine the advantages and avoid the disadvantages of these two methods. Facing the problem of tracking policy optimization for multiple pursuers, this study proposed a new form of fuzzy actor–critic learning algorithm based on suboptimal knowledge (SK-FACL). Specifically speaking, based on the utilization of suboptimal knowledge, the data-driven learning process could be sped up. Therefore, from the knowledge-driven perspective, the proposed SK-FACL introduces a policy iteration process, which makes the policy more adaptive to an inaccurate modeling environment. From the data-driven perspective, SK-FACL introduces a suboptimal guided policy, improving the learning efficiency of the original FACL. To sum up, the key contributions of this study are as follows:

(1) A new form of accelerating fuzzy actor–critic learning algorithm framework was proposed, where the represented prior knowledge can be continuously optimized together with the whole policy, and the combination of the knowledge controller and the fuzzy actor–critic learning algorithm helps the learning process start quickly and enables the agent to learn faster.

(2) The combination of the knowledge controller and the fuzzy actor–critic learning algorithm makes the whole policy more robust and enables the agent to effectively modify the tracking policy under pure knowledge control and approach the ideal tracking policy.

(3) The proposed policy framework was constructed based on a fuzzy inference system, which increases the interpretability of the whole policy and enables researchers to analyze the logic of how agents operate more easily and clearly.

The structure of the rest of this paper is as follows: Section 2 presents the multi-agent tracking scenario and the fuzzy actor–critic learning algorithm; Section 3 shows the overall design of the proposed SK-FACL; Section 4 discusses the employment of the Apollonius circle and more details about the policy iteration; Section 5 simulates the proposed algorithm and several competitive other methods; and finally, Section 6 presents the conclusions.

## 2. Preliminaries

This study mainly focused on the ground differential game; therefore, the involved game scenarios are introduced in Section 2.1. In addition, to clearly express the proposed SK-FACL, which is based on FACL, the introduction of FACL is discussed in Section 2.2.

### *2.1. Description of the Ground Tracking Problem*

For the sake of simplicity, each of the pursuers and evader is supposed to have a constant speed, where the evader will be faster than the pursuers. In this study, we supposed there existed two pursuers, which were labeled $P_0$ and $P_1$, and one high-speed evader labeled $E_0$. The initial conditions of $P_0$, $P_1$ and $E_0$ are illustrated in Figure 1. It was defined that the symbol $v_p$ represents the constant speed of each pursuer, and the symbol $v_e$ represents that of the evader, satisfying the condition $v_e > v_p$. It is shown in Figure 1 that the initial positions of the pursuers were represented by $\{x_{P_i}, y_{P_i}\}(i = 0, 1)$ and that of the evader was $\{x_{E0}, y_{E0}\}$. For this game, there were two assumptions that needed to be satisfied:

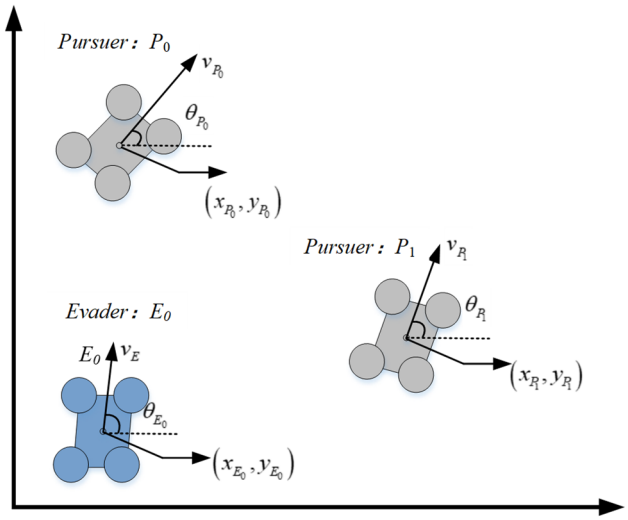



**Figure 1.** The initial condition of the two pursuers and the evader.

(1) Each pursuer was supposed to have the ability to know the immediate position of the evader at every time step $t$.

(2) The constant speed of the evader could be measured by each pursuer.

Based on these conditions, the capture of the evader occurs if the distance between the pursuer $P_i$ and the evader $E_0$ is less than or equal to the capture distance $d_c$, i.e., $\left\|(x_{P_i}, y_{P_i}) - (x_e, y_e)\right\| \leq d_c$. From the capture condition, it is seen that if one of the pursuers can meet the capture distance criterion, the evader will have failed to escape.

The pursuers and the evader are expected to have self-learning abilities; therefore, they can be seen as agents. Each agent here followed a model expressed as follows:

$$\begin{cases} P_0(k+1) = P_0(k) + \boldsymbol{v}_{P0}(k) \\ P_1(k+1) = P_1(k) + \boldsymbol{v}_{P1}(k) \\ E_0(k+1) = E_0(k) + \boldsymbol{v}_{E0}(k) \end{cases} \tag{1}$$

where $k$ represents the $k$th point of time and $k+1$ represents the next point of time after $k$. In addition, each pursuer and the evader have a constant speed $v_{Pi}(i = 0, 1)$ and $v_E$ with the heading angles $\theta_{Pi}(i = 0, 1)$ and $\theta_{E0}$, respectively.

### *2.2. The Fuzzy Actor–Critic Learning Algorithm*

Actor–critic learning algorithm (AC algorithm) is the most popular basic algorithm in RL, as it is able to deal with decision problems in continuous systems. Generally, a type of actor–critic learning system contains three parts: the actor part and two critic parts. The function of the actor part is to choose the optimal action for each state, generating the final policy. The critic parts are used to estimate the value functions of the agent in the current and the next step. The reason why AC can have its unique advantage is that the policy and value functions are composed of networks. For an arbitrary actor–critic reinforcement learning system, in order to complete the policy optimization under continuous-state space and continuous-action space, it is necessary to input the state $s_t$ into the actor network and critic network to obtain the action $a_t$ and value function $V_t$. Generally, such networks can be formed using neural networks. However, the fuzzy actor–critic reinforcement learning algorithm involved in this study was based on fuzzy inference systems instead of neural networks. Using fuzzy inference systems does not affect the mapping connection of states to actions and value functions, but it enhances the interpretability of network parameters, making the mapping structure easier to trace and understand. Compared with an ANN, a fuzzy inference system (FIS) is more feasible to be explained, and the necessary human knowledge can be considered to build the inference rules. In other words, how agents operate in the learning process under FIS can be more easily and clearly analyzed.

Therefore, by introducing the FIS into the AC algorithm, the fuzzy actor–critic learning algorithm (FACL) can be obtained.

For the FACL, the basic architecture is illustrated in Figure 1. From the figure, it is seen that the environment can provide the state information $s_t$ and $s_{t+1}$ at the $t$ and $t+1$ time steps to the agent, meanwhile giving out the reward $r_t$. The actor provides the operation instruction $a_t$ of the agent via the $s_t$, and then a white Gaussian noise $\varepsilon$ is added as an exploration mechanism. Finally, the input to the agent $a_t{}'$ is obtained. The two critic parts can estimate the value functions $\hat{V}(s_t)$ and $\hat{V}(s_{t+1})$ at $s_t$ and $s_{t+1}$, respectively. Based on the time difference error $\Delta_t$, the networks of the actor and critic can be updated.

In the FACL, the actor is represented by an adaptive fuzzy logic controller (FLC), which is composed of FIS. From Figure 2, it can be seen the inference parts of actor and critic are the same; this is because that the FIS employed in the actor shares the same basic structure as that in the critic. With the triangular membership functions, eight rules are activated for three inputs at one time. Further, the output of actor $a_t$ and the output of critic $\hat{V}(s_t)$ are calculated according to the consequent sets $\{\omega^1, \omega^2, \ldots, \omega^8\}$ and $\{\vartheta^1, \vartheta^2, \ldots, \vartheta^8\}$, respectively.

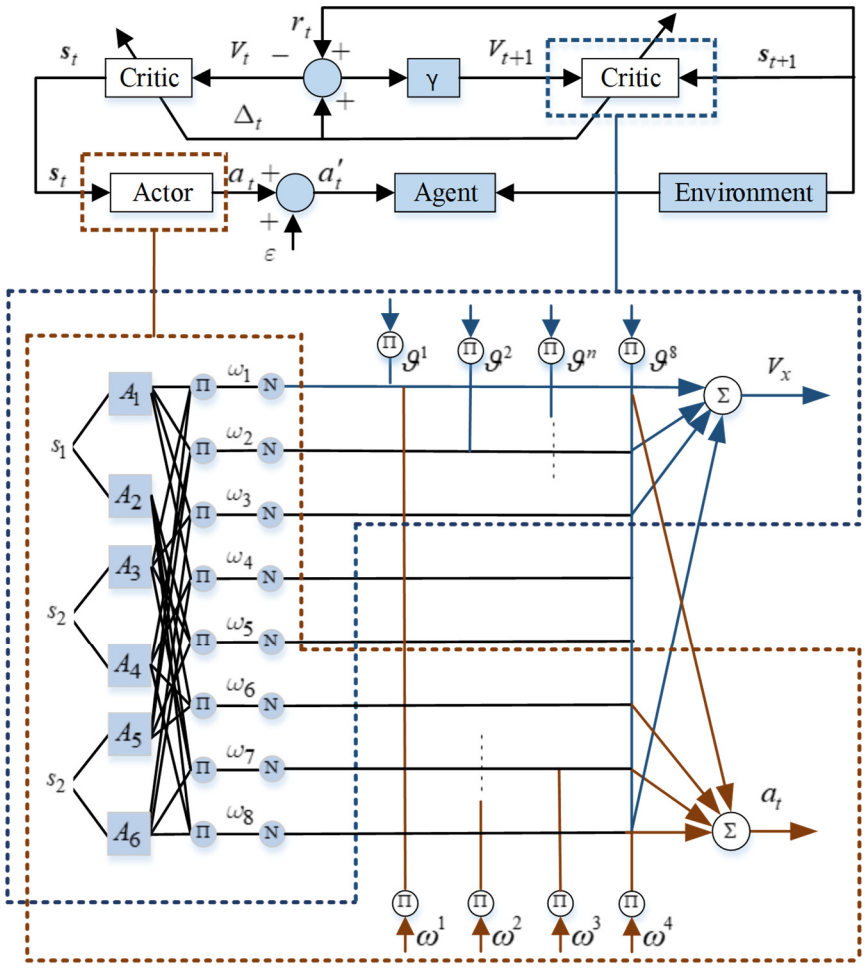



**Figure 2.** The architecture of the fuzzy actor–critic learning algorithm.

For a reinforcement learning process, the main goal of the agent is to maximize the long-run discounted return $R_t$, which is given as

$$R_t = \sum_{k=0}^{T} \gamma^k r_{t+k} \tag{2}$$

where $\gamma$ represents the discount factor and satisfies $0 \leq \gamma \leq 1$, $t$ is the current time step and $r_{t+k}$ is the immediate reward at the time step $t+k$. To evaluate the performance of the running policy, the value function is needed. The value function at the current state is defined as the expected sum of the discounted rewards, as shown in Equation (3):

$$V(s_t) = E\left\{\sum_{k=0}^{\infty} \gamma^k r_{t+k}\right\} = r_t + \gamma V(s_{t+1}) \tag{3}$$

For a temporal difference (TD)-based method, the agent uses an estimated value function $\hat{V}(s_t)$ instead of the real $V(s_t)$. Based on $\hat{V}(s_t)$ and $\hat{V}(s_{t+1})$, the TD error $\delta_t$ can be obtained:

$$\delta_t = r_t + \gamma \hat{V}(s_{t+1}) - \hat{V}(s_t) \tag{4}$$

Since the actor is represented by an adaptive FLC, the output is expressed as

$$a(t) = \sum_{l=1}^{M} \varphi^l \omega^l(t) \tag{5}$$

where $\omega^l(t)$ represents the consequent parameter in the FLC, $M$ is the number of fuzzy rules and the $\varphi^l$ is the firing strength of the rule $l$. The firing strength for the rule $l$ is

$$\varphi^l = \frac{\prod_{i=1}^{n} \mu^{F_i^l}(x_i)}{\sum_{l=1}^{M} \prod_{i=1}^{n} \mu^{F_i^l}(x_i)} \tag{6}$$

where $n$ is the number of inputs and $\mu^{F_i^l}$ is the membership degree of input $x_i$ in fuzzy rule $F_i^l$. When selecting membership functions, one can choose continuously varying membership functions, such as Gaussian curves, or choose triangular curves. Here, the triangular membership functions were mainly considered to lower the computational cost. For example, for a fuzzy inference system with two inputs and three membership functions for each input, nine fuzzy rules need to be activated to obtain the output. Furthermore, the more membership functions each input is equipped with, the number of rules that need to be activated will increase exponentially. For a triangular membership function, no matter how many fuzzy rules are equipped with an input, only two functions will be activated at one time. Therefore, this will greatly save computational consumption. Moreover, for any given input, the sum of the firing strengths is always equal to 1, which avoids ambiguity. Taking the distance input as an example, Figure 3 shows the sets of membership functions, namely, "negative far", "close" and "positive far".

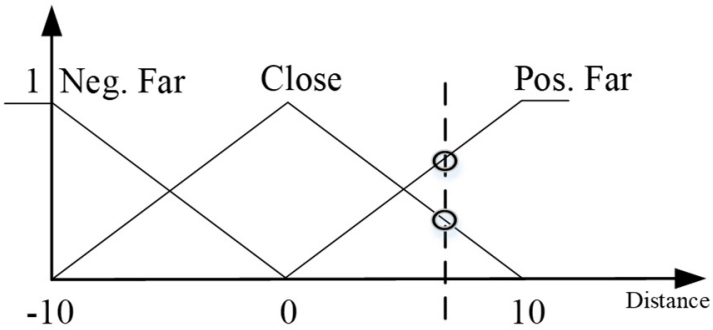



**Figure 3.** Triangular membership functions for the inputs.

For triangular membership functions, the firing strength expressed in Equation (6) can be rewritten as below:

$$\varphi^l = \prod_{i=1}^{n} \mu^{F_i^l}(x_i) \tag{7}$$

In order to promote exploration of the action space, a random white noise $\varepsilon$ chosen from a Gaussian distribution by $N(0, \sigma^2)$ was added to the output of the FLC of $a_t$ to generate the real control signal $a_t{}'$. The relationship is shown below:

$$a_t{}' = a_t + N(0, \sigma^2) \tag{8}$$

Based on the TD error $\delta_t$, the consequent parameter $\omega^l(t)$ can be adapted to

$$\omega^l(t+1) = \omega^l(t) + \beta_L \delta_t (a_t' - a_t) \frac{\partial a}{\partial \omega^l} \tag{9}$$

where $\beta_L \in (0, 1)$ is the learning rate for the FLC and the term $\frac{\partial a}{\partial \omega^l}$ equals the firing strength $\varphi^l$.

In the FACL, the agent can select its action according to the FLC, and then it comes to the interaction with the environment. Before and after the interaction, the two critic parts are expected to evaluate the value function at $s_t$ and $s_{t+1}$. The evaluation is used to determine whether the new policy is going to be better or worse than expected. For the state $s_t$, the output of the critic $\hat{V}(s_t)$ is an approximation to $V(s_t)$ given by

$$\hat{V}(s_t) = \sum_{l=1}^{M} \varphi^l \vartheta^l(t) \tag{10}$$

where $\vartheta^l(t)$ is the consequent parameter of the critic and $\varphi^l$ is the same firing strength as in the FLC. Similar to the update rule in Equation (9), the parameter $\vartheta_l^t$ can be adapted to

$$\vartheta^l(t+1) = \vartheta^l(t) + \alpha_L \delta_t \frac{\partial \hat{V}(s_t)}{\partial \vartheta^l} \tag{11}$$

where $\alpha_L \in (0, 1)$ is the learning rate for the critic and the partial derivative term $\frac{\partial \hat{V}(s_t)}{\partial \vartheta^l}$ is calculated as $\varphi^l$. In order to prevent instability in the actor, we set $\beta_L < \alpha_L$, making sure that the actor will converge slower than the critic.

To sum up, the updating laws for the consequent parameters $\omega^l$ and $\vartheta^l$ are implemented as below:

$$\begin{cases} \omega^l(t+1) = \omega^l(t) + \beta_L \delta_t \varepsilon \varphi^l \\ \vartheta^l(t+1) = \vartheta^l(t) + \alpha_L \delta_t \varphi^l \end{cases} \tag{12}$$

The basic composition of the FACL is expressed above. Based on the architecture of the FACL, this study added the knowledge control part, which combined the suboptimal policy, speeding up the learning process of the FACL while correcting the suboptimal policy.

## 3. Overall Design of the Proposed SK-FACL

For the ground tracking problem described in Section 2.1, although the FACL can be employed to deal with it, there still exist several disadvantages:

(1) The efficiency of pure data-driven learning is low, and the prior knowledge that can be obtained from the environment will be wasted.

(2) It is difficult to quickly adjust the tracking policy only using an iteration process when the target suddenly turns.

Therefore, in view of the above shortcomings, this study introduced the Apollonius circle to give the guided policy from the perspective of geometric planning on the basis of the original FACL [39]. By integrating the guided policy into the network of reinforcement learning, the prior information is utilized, and the policy that copes with the escaping target, which may suddenly turn, can be adjusted quickly. The overall algorithm design is shown in Figure 4.

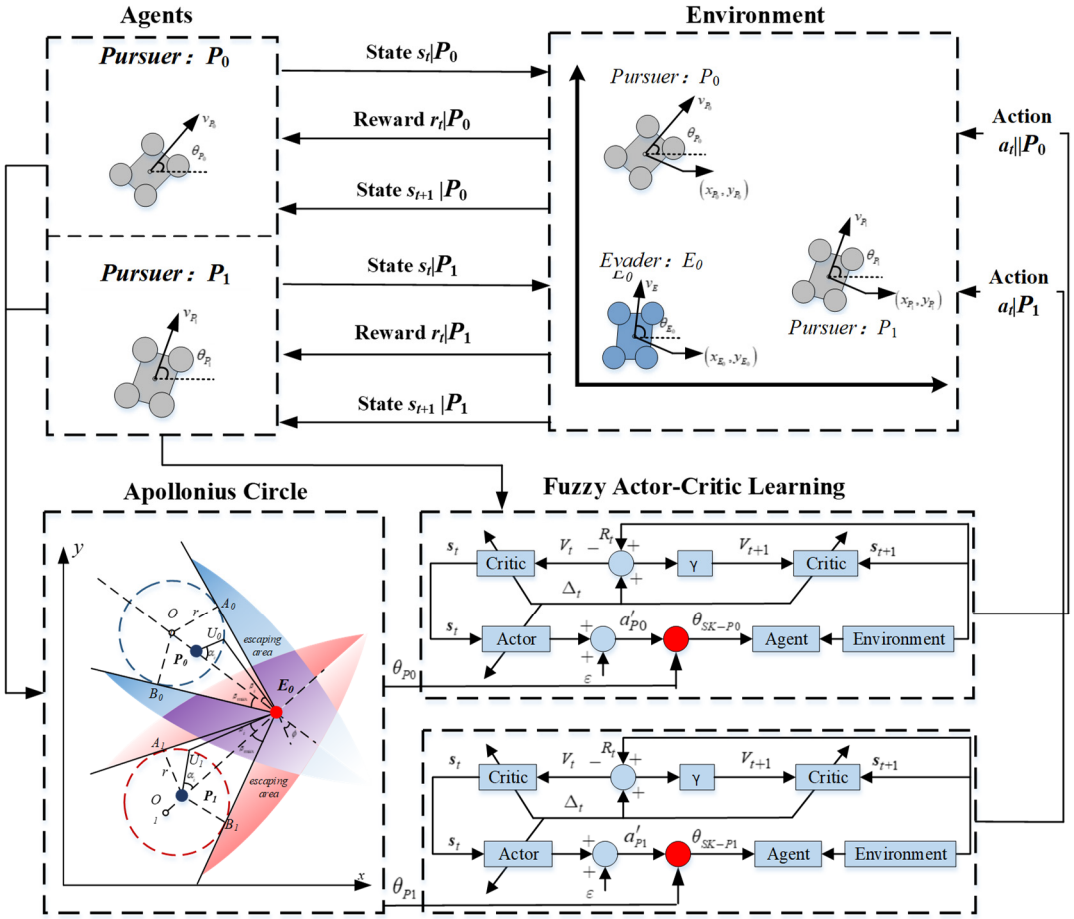



**Figure 4.** The overall design diagram of the SK-FACL.

It can be seen from the figure that there were two pursuers involved in this study, namely, $P_0$ and $P_1$, which could interact with the environment. By inputting state and action into the environment, the next state and reward can be obtained, which is a typical interactive process in RL. By extracting the available prior knowledge from the environment, the pursuer can plan a guided tracking policy for the target based on the Apollonius circle. The planned policies $\theta_{P0}$ and $\theta_{P1}$ are introduced into the FACL and integrated with the original reinforcement learning policies $a\prime_{P0}$ and $a'_{P1}$ to form $\theta_{SK-P0}$ and $\theta_{SK-P1}$, respectively, so that they continue to participate in the interaction process with the environment. Although the pursuers $P_0$ and $P_1$ are trained to track the target $E_0$ independently, they have a backup relationship with each other. When one of them successfully pursues, it can be regarded as a successful mission.

It can be seen that the SK-FACL designed in this study, in addition to inheriting the good interpretability brought about by fuzzy logic in the FACL, also has unique advantages in the following aspects: (1) it introduces the guided policy based on the Apollonius circle into the FACL, making the policy iteration start quickly and accelerating the learning process; (2) the policy based on the iteration of the FACL can correct the update mistakes caused by those guided policies with errors; and (3) the pursuers can quickly adapt the policy to the situation of the target turning suddenly because the guided policy generated by Apollonius circle is updated in real time.

## 4. Accelerating Fuzzy Actor–Critic Learning Algorithm via Suboptimal Knowledge

Based on the FACL introduced above, many decision problems can be handled. However, such a type of pure model-free learning method will make the learning efficiency low. Therefore, it is a reasonable idea that we combine the suboptimal policy obtained from the

knowledge control and the original FACL. In this way, the proposed accelerating fuzzy actor–critic learning algorithm via suboptimal knowledge (SK-FACL) helps the agent start quickly and speed up the learning process; furthermore, due to the advantages of fuzzy inference rules, the obtained policy will have its physical meaning.

### *4.1. Guided Policy Based on the Apollonius Circle*

For the involved scenario in Section 2.2, the basic idea to obtain the suboptimal knowledge is to use the characteristics of the Apollonius circle. This subsection presents a discussion of the necessary conditions that should be satisfied when the pursuers $P_0$ and $P_1$ have the ability to capture the high-speed evader $E_0$.

Figure 5 shows the initial positions of $P_i(i = 0, 1)$ and $E_0$, denoted as $(x_{Pi}, y_{Pi})$ and $(x_{E0}, y_{E0})$, respectively. Moreover, $O_i$ denotes the center of the Apollonius circle of the pursuer $P_i$, $r_i$ represents the radius of the Apollonius circle and $U$ represents the set of all points on the Apollonius circle that satisfies Equation (13):

$$\zeta = \frac{P_i U}{E_0 U} = \frac{V_p}{V_e} \tag{13}$$

where $\zeta \in (0, 1)$ is a constant scale factor, $V_p$ represents the constant speed of $P_i$ and $V_E$ represents the constant speed of $E_0$. The expressions of the center $O_i$ and the radius $r_i$ are given by

$$O_i = \left(\frac{x_{p_i} - \zeta^2 x_e}{1 - \zeta^2}, \frac{y_{p_i} - \zeta^2 y_e}{1 - \zeta^2}\right) \tag{14}$$

$$r_i = \frac{\zeta\sqrt{(x_{p_i} - x_e)^2 + (y_{p_i} - y_e)^2}}{1 - \zeta^2} \tag{15}$$

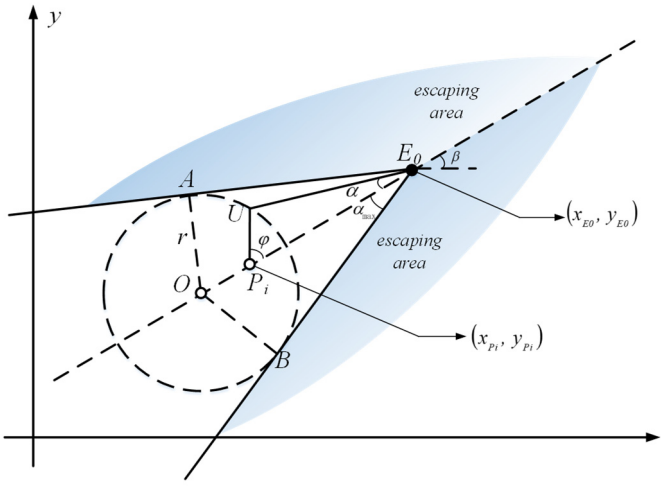



**Figure 5.** Geometric illustration for the capture condition based on the Apollonius circle.

From Equation (15), it is seen that the radius $r_i$ is monotonic with the scale factor $\zeta$, and when the value of $\zeta$ decreases, the value of $r_i$ goes down. Since the radius $r_i$ is used to specify the pursuer's region of responsibility, the smaller value of $r_i$, the smaller the region. Accordingly, the evader will have a wide path to escape from the pursuer.

On the basis of the trigonometric function, we have

$$\frac{V_e}{\sin(\varphi)} = \frac{V_p}{\sin(\alpha)} \rightarrow \sin(\alpha) = \frac{V_p}{V_e}\sin(\varphi) = \zeta sin(\varphi) \tag{16}$$

since the evader is set to move faster than the pursuer, which means that $\zeta < 1$. Define the angle between the line-of-sight and the evader's heading direction as $\alpha$; the max value $\alpha_{max}$ can be calculated using

$$\alpha_{max} = \arcsin\left(\frac{V_p}{V_e}\right) = \arcsin(\zeta) \tag{17}$$

As shown in Figure 5, when the angle $\alpha$ is not greater than $\alpha_{max}$, it is obvious that the pursuer $P_i$ can always find an angle $\varphi$ that ensures the capture of the high-speed evader $E_0$. In other words, if the movement direction of $E_0$ is within the angle $\angle AEB$, the pursuer will capture the evader; otherwise, the evader can escape. The lines $EA$ and $EB$ in Figure 5 are tangent to the Apollonius circle, and the two points $A$ and $B$ are called the virtual targets. Therefore, the pursuer $P_i$ can adapt to the evader's movement within an angle of $\eta$, as shown below:

$$\eta = 2\alpha_{max} = 2\arcsin(\zeta) \tag{18}$$

The angle of $\overrightarrow{PE}$ is defined as $\beta$; thus, the evader $E_0$ cannot escape within the region

$$\mathcal{H} = (\beta + \alpha_{max} - \pi, \beta - \alpha_{max} + \pi) \tag{19}$$

where the region $\mathcal{H}$ should satisfy $-\pi < \mathcal{H} \leq \pi$.

Therefore, if the evader $E_0$ moves within the region $\mathcal{H}$, the pursuer $P_i$ will always find an optimal moving direction to capture it. It is seen from Figure 5 that if $E_0$ moves along $\overrightarrow{E_0U}$, where $\alpha < \alpha_{max}$ holds, $P_i$ will be able to move along $\overrightarrow{P_iU}$. In this condition, $E_0$ will be captured by $P_i$ at $U$.

Therefore, based on the Apollonius circle, it was concluded that if the moving angle of $E_0$ is $\beta - \alpha + \pi$ ($\alpha < \alpha_{max}$), the responded moving angle of $P_i$ will be $\beta + \varphi$ along with $\overrightarrow{P_iU}$. According to the law of sines, we have

$$\frac{\sin\varphi}{\sin\alpha} = \frac{E_0U}{P_iU} = \frac{1}{\zeta} \tag{20}$$

Then, the optimal moving policy for $P_i$ can be obtained as below:

$$\theta_{Pi} = \beta + \varphi = \sin^{-1}\left(\frac{\sin\alpha}{\zeta}\right) + \beta(\alpha < \alpha_{max}) \tag{21}$$

When the condition satisfies $\alpha_{max} < \alpha < \pi$, the pursuer $P_i$ cannot capture the evader $E_0$. However, if $P_i$ still moves with the directional angle $\beta + \varphi$, it will get as close as possible to $E_0$, which can be seen in Figure 6.

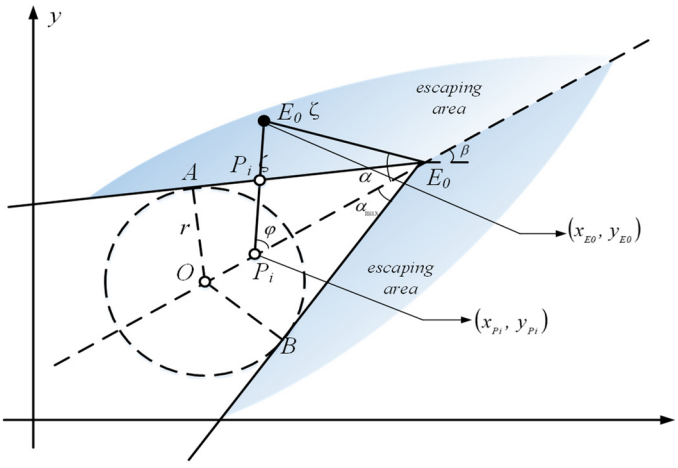



**Figure 6.** The optimal policy of the pursuer when the escaper escapes.

From Figure 6, it is seen that if $E_0$ moves to $E_0'$ in a time interval $\Delta t$, the optimal movement for $P_i$ will be heading to $E_0'$. Therefore, based on the geometric relation, it can be expressed that

$$\theta_{Pi} = \beta + \varphi = \sin^{-1}(\frac{\sin\alpha}{\zeta}) + \beta(\alpha < \alpha_{max}) \tag{22}$$

Therefore, if $\alpha_{max} < \alpha < \pi$, the optimal moving policy for $P_i$ can be obtained:

$$\theta_{Pi} = \beta + \varphi = \cot^{-1}\left[\frac{|E_0P_i|}{V_e\Delta t\sin\alpha} - \cot\alpha\right] + \beta \tag{23}$$

To sum up, in a ground tracking problem, the optimal moving policy for the pursuer $P_i$ should be as follows:

$$\theta_{Pi} = \beta + \varphi = \begin{cases} \beta + \sin^{-1}(\frac{\sin\alpha}{\zeta}) if \alpha \le \alpha_{max} \\ \beta + \cot^{-1}\left[\frac{|E_0P_i|}{V_e\Delta t\sin\alpha} - \cot\alpha\right] if \alpha_{max} < \alpha < \pi \end{cases} \tag{24}$$

This optimal moving policy $\theta_{Pi}$ was obtained from the Apollonius circle, which uses the concept of knowledge control. If the Apollonius circle is not accurate enough, the policy will have a decreased confidence level. However, it still has the potential to be employed in model-free reinforcement learning to accelerate the process.

### *4.2. Accelerating Fuzzy Actor–Critic Learning via Suboptimal Knowledge*

From Section 4.1, it is seen that if the velocities $V_p$ and $V_e$ and the positions of the pursuer and evader can be accurately known, the ideal optimal capturing policy can be obtained based on the Apollonius circle. However, in the real world, this kind of information cannot be measured accurately; therefore, we can only obtain the suboptimal policy based on rough data. Such a suboptimal policy can be seen as a type of knowledge control, helping the agent (especially the pursuing team) to quickly start the learning process and learn faster.

The differential game involved in this study was about two pursuers and one evader, which is illustrated in Section 2.1. Based on the Apollonius circles, the game can be shown in Figure 7. From the figure, it is seen that the pursuing team has $P_0$ and $P_1$, and the evader is denoted as $E_0$. If $P_0$ and $P_1$ can obtain the rough information $\hat{V}_p$ and $\hat{V}_e$, the suboptimal policy $\theta_{P1}$ and $\theta_{P0}$ will be obtained based on Equation (24).

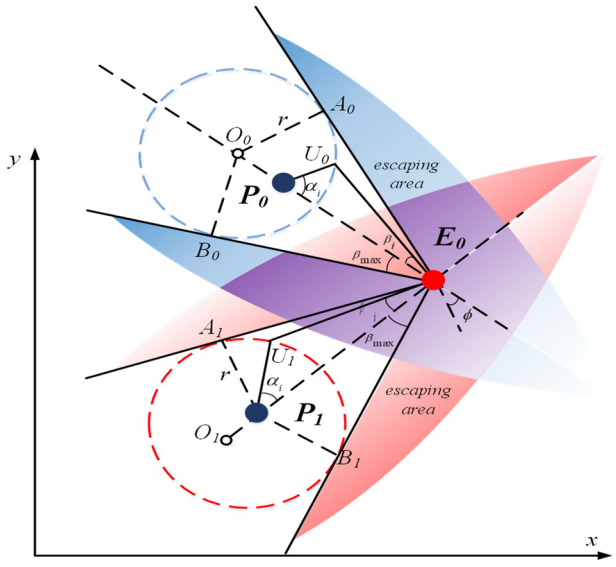



**Figure 7.** A game between two pursuers and one evader based on Apollonius circles.

Since the suboptimal policies $\theta_{P1}$ and $\theta_{P0}$ cannot guarantee the pursuers' successful capture of the evader, the FACL was introduced here to form a new learning algorithm called the accelerating fuzzy actor–critic learning algorithm via suboptimal knowledge (SK-FACL). The overall logic architecture of the proposed SK-FACL is shown in Figure 8.

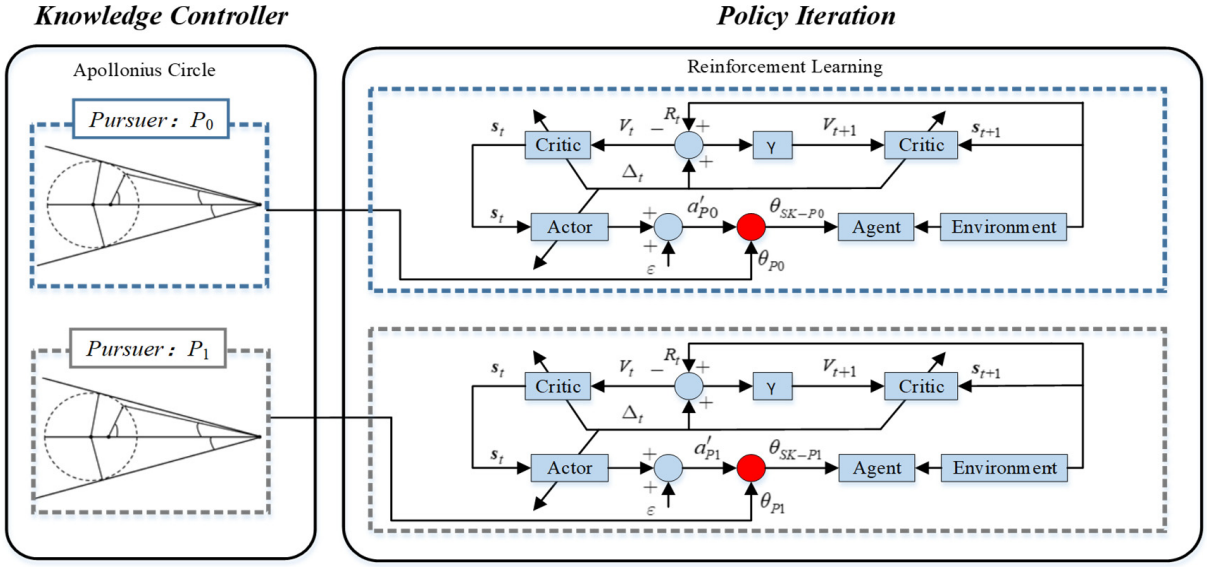



**Figure 8.** The learning logic of the proposed accelerating fuzzy actor–critic learning algorithm.

As can be seen from Figure 8, the logic of the proposed algorithm is mainly divided into two parts: the knowledge controller part and the policy iteration part. In the architecture, the role of knowledge controller is played by the Apollonius circle method. Through the rough information of the estimated speed $\hat{V}_p$ and $\hat{V}_e$, the suboptimal tracking angles $\theta_{P1}$ and $\theta_{P0}$ can be calculated as the initial policies for the pursuing team. The part of policy iteration is mainly built using the fuzzy actor–critic learning framework. Different from the FACL, the output of the actor $a'_{P_i}(i = 0, 1)$ is superimposed with the output of the knowledge control $\theta_{Pi}(i = 0, 1)$ to form $\theta_{SK-Pi}(i = 0, 1)$. In this way, the knowledge-driven part and the data-driven part are effectively combined. The knowledge controller is responsible for guiding the direction of initial policy generation and the policy iteration part can finely adjust the capturing policy based on the interaction with the environment.

In this game, the state $\boldsymbol{s}_t = [s_1, s_2, s_3]$ is designed as a vector, which contains three types of information. The symbol $s_1$ represents the relative distance between $P_i$ and $E_0$, $s_2$ represents the heading angle difference between $P_i$ and $E_0$, and $s_3$ represents the derivative of the angle difference. The design of the reward $r_t$ is expressed as follows:

$$r_t = v_1((d_{k-1}(\boldsymbol{s}_{t-1}) - d_{k-1}(\boldsymbol{s}_t)) - (d_k(\boldsymbol{s}_{t-1}) - d_k(\boldsymbol{s}_t))) + v_2(d_{k-1}(m) - d_k(m)) + v_3{}^{(v_4 - d_k(m))} \tag{25}$$

where $v_1, v_2, v_3$ and $v_4$ are coefficients. Meanwhile, $d_{k-1}(\boldsymbol{s}_{t-1})$ represents the relative distance at $\boldsymbol{s}_{t-1}$ in the $(k-1)$th training instance and $d_{k-1}(\boldsymbol{s}_t)$ represents that at $\boldsymbol{s}_t$ in the $(k-1)$th training instance. The symbol $d_k(m)$ is defined as the minimum distance between $P_i$ and $E_0$ in the $k$th training instance and $d_{k-1}(m)$ is defined as the minimum distance between $P_i$ and $E_0$ in the $(k-1)$th training instance. It is seen that the reward $r_t$ can be divided into three parts. The first part gives the agent $P_i$ a local bonus to help $P_i$ to get closer to the evader as quickly as possible. The second part gives the agent $P_i$ a global bonus so that $P_i$ can get as close to the evader as possible at a certain time during each training session. Furthermore, in the final part, if $d_k(m) < v_4$, $P_i$ will obtain an exponentially increasing global bonus, which can help $P_i$ to obtain a global minimum distance that is as small as possible during each training session.

Set the inputs $s_1, s_2$ and $s_3$ to have three triangular membership functions with uniform distributions in $s_1 \in [-10, 10]$, $s_2 \in [-1, 1]$ and $s_3 \in [-1, 1]$ (for an out-of-range state, the membership function value nearest to that state can be set to one). The flow of the proposed accelerating fuzzy actor–critic learning algorithm via suboptimal knowledge is given in Algorithm 1.

**Algorithm 1** Accelerating fuzzy actor–critic learning algorithm via suboptimal knowledge

1: Initialize the state $\boldsymbol{s}_0$ and discount factor $\gamma$ for each pursuer and the evader
2: Initialize the membership functions for $s_1, s_2$ and $s_3$
3: Initialize the consequent set of the actor and critic $\{\omega^1, \omega^2, \ldots, \omega^{12}\}$ and $\{\vartheta^1, \vartheta^2, \ldots, \vartheta^{12}\}$
4: Initialize the learning rates $\alpha_L$ and $\beta_L$
5: Estimate the velocities $\hat{V}_p$ and $\hat{V}_e$
6: **For each iteration** $i$ **do**
7: Calculate the proper $\theta_{P1}$ and $\theta_{P0}$ for $P_0$ and $P_1$ according to the Apollonius circle
8: **For each step do**
9: Calculate the output of the critic $V_t$ in the current state $\boldsymbol{s}_t$
10: Obtain the actions $a'_{P0}$ and $a'_{P1}$ generated from the actor
11: Combine the $\theta_{Pi}$ and $a'_{Pi}$ to form $\theta_{SK-Pi}$ for each pursuer
12: Perform the action $\theta_{SK-Pi}$ for each pursuer, and get the next state $s_{t+1}$, and the reward $r_t$
13: Calculate the time difference $\Delta_t$
14: Update the consequent parameters $\vartheta^l$ for the critic
15: Update the consequent parameters $\omega^l$ for the actor
16: **End for**
17: **End for**

## 5. Numerical Results

To verify the effectiveness and show the superiority of the proposed SK-FACL, three kinds of cases were simulated here. The simulated results indicated that pure knowledge control and the pure FACL have shortcomings when the agents face an incompletely known environment. However, if the two methods are combined to form the SK-FACL, the tracking performance of pursuers can be improved greatly. Since the game is an abstraction of the real pursuit–evasion problem, the involved position and velocity information is dimensionless.

### *5.1. Case 1: Pure Knowledge Control*

In order to verify that the Apollonius circle can guide the initial policy in the game, this case drove the pursuers under pure knowledge control. The positions of the two pursuers were $(x_{P0}, y_{P0}) = [2500, -1500]$ and $(x_{P1}, y_{P1}) = [3200, 2500]$, and that of the evader was $(x_{E0}, y_{e0}) = [-500, -500]$. Moreover, the initial heading angles of $P_0$, $P_1$ and $E_0$ were $\theta_{P0} = 0rad$, $\theta_{P1} = 3rad$ and $\theta_{E0} = 0.25rad$, respectively. The pursuers and the evader held constant velocities, where $v_{P0} = v_{P1} = 45/s$ and $v_{E0} = 50/s$, and it was supposed there existed no velocity estimation error. The simulated time was set to 200 s, and the results with different time steps are shown in Figure 9.

Figure 9 shows the traces of the three agents without the velocity estimation error at different time stages. Comparing Figure 9a,b, it is seen that $P_0$ and $P_1$ started to track $E_0$ from different locations according to the policies calculated using Apollonius circles. Moreover, Figure 9c shows that $P_0$ continues to track $E_0$ along a straight line after moving in the same direction as $E_0$. With the increase in time, Figure 9d shows that $P_0$ and $P_1$ met and kept chasing $E_0$ together, which is an ideal tracking policy. In Figure 9d, the minimum distances between $P_0$ and $E_0$ and between $P_1$ and $E_0$ were 12.4 and 23.0, respectively. It is seen that if there was no estimated error of velocity, the pursuing team could track the evader perfectly based on Apollonius circles, which meant that the knowledge control worked well in this ideal environment.

However, if the environment is not ideal, the Apollonius circles will not be accurately obtained. When the estimated velocity holds an error of 30% compared with the real velocity, the tracking results are drawn in Figure 10.

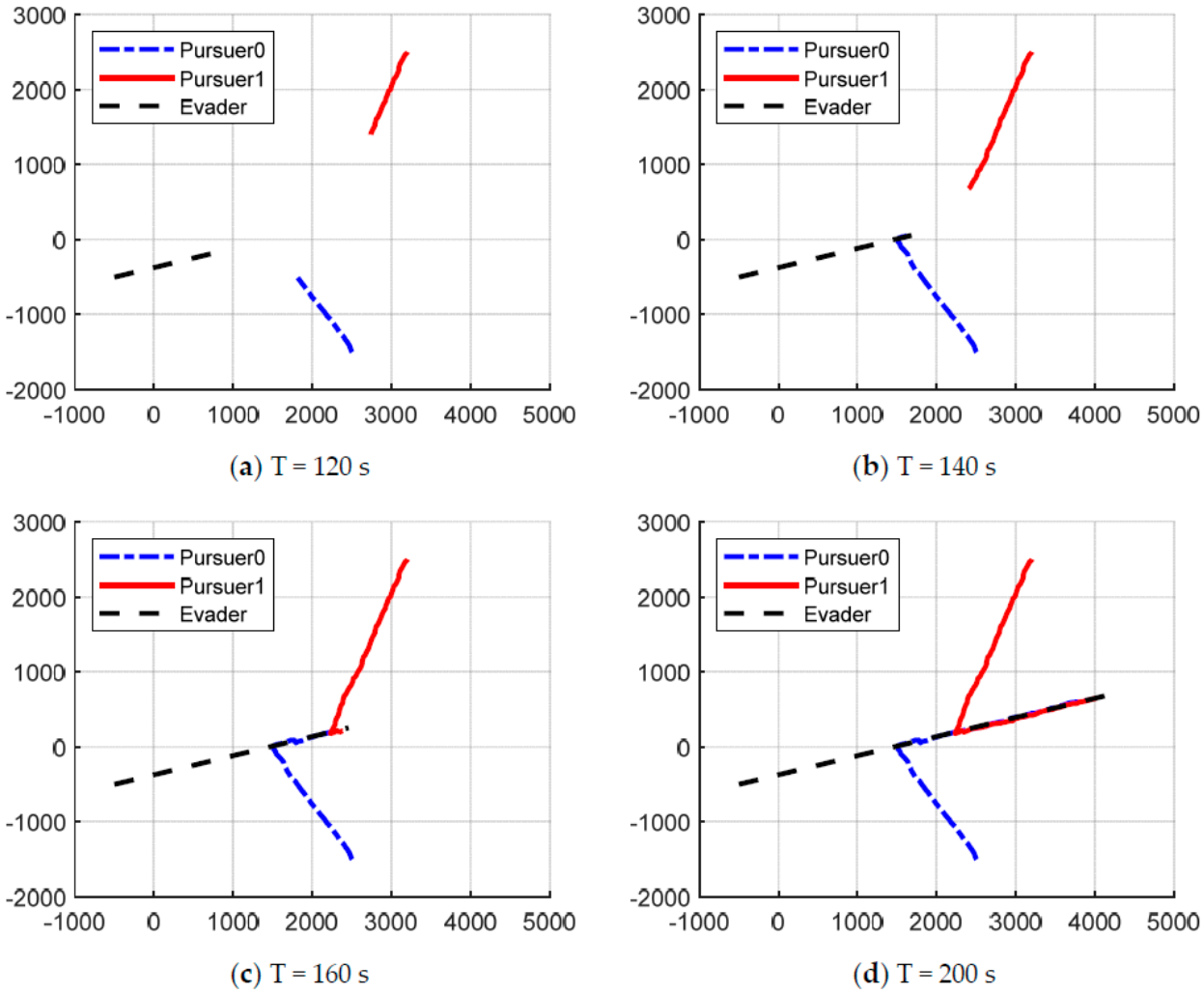



**Figure 9.** The traces of $P_0$, $P_1$ and $E_0$ based on the Apollonius circle without velocity error.

Figure 10 shows the traces in the presence of the velocity estimation error $\Delta v = 30\% v_P$. By comparing Figure 10a with Figure 10d, it can be seen that the policy calculated according to the Apollonius circles still retained a strong tracking ability in the presence of the estimated error. This indicates that the Apollonius circle method has the potential to be employed as the knowledge control part and plays a good guiding role for the policy in the incompletely known environment. However, different from the results in Figure 9, the traces of $P_0$ and $P_1$ appeared to have arcs and jitters in Figure 10 due to the velocity estimation error. The error caused inaccuracy in the Apollonius circles. In Figure 10d, the minimum distances between the $P_0$ and $E_0$ and between $P_1$ and $E_0$ were 155.6 and 112.7, respectively. Thus, the tracking traces were not ideal and not as good as the results shown in Figure 9d. Based on the above analysis, it is seen that if we combined the policy iteration method (which mainly refers to reinforcement learning), the tracking performance could be modified on the premise of retaining the guiding role of knowledge control.

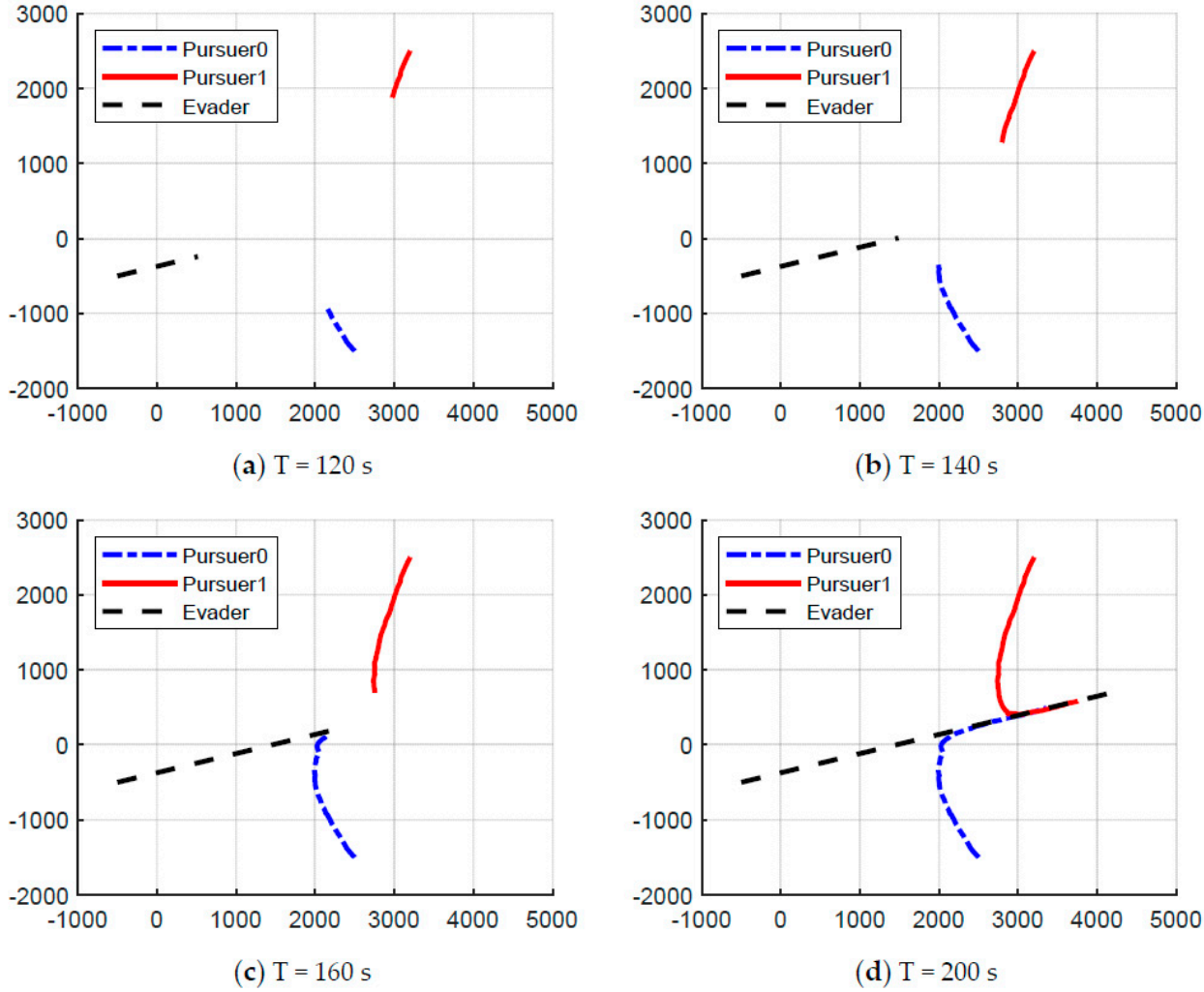



**Figure 10.** The traces of $P_0$, $P_1$ and $E_0$ based on Apollonius circles with an estimated speed error $\Delta v = 30\% v_P$.

### *5.2. Case 2: Accelerating Fuzzy Actor–Critic Learning via Suboptimal Knowledge*

The basic idea of the proposed SK-FACL is to employ the method of Apollonius circles as the knowledge control part and to combine this with the policy iteration method of the FACL to give full play to the guiding role of knowledge control and the correction role of data-driven method at the same time. In this way, the tracking abilities of the pursuers can be improved in the involved differential game. In order to highlight the advantages of the SK-FACL, this subsection compares the tracking performances under pure knowledge control and the SK-FACL. The settings of the position, speed and angle of $P_0$, $P_1$ and $E_0$ were the same as in Section 5.1. The hyperparameters were set to $\alpha_L = 0.001$, $\beta_L = 0.001, \gamma = 0.93, \sigma = 0.02, v_1 = 10, v_2 = 1, v_3 = 1.05$ and $v_4 = 80$.

From Figure 9, we know that the ideal tracking traces of $P_0$ and $P_1$ will be more like straight lines to chase $E_0$ more effectively. Therefore, although Figure 11 shows that $E_0$ could also be tracked by $P_0$ and $P_1$ based on inaccurate Apollonius circles, the traces were more like arcs, which are not ideal. After being combined with the FACL, Figure 12 shows that the traces were modified to be more like ideal ones based on the SK-FACL. It is seen that the agents $P_0$ and $P_1$ were using knowledge control as the guided policies and then correcting them based on reinforcement learning.

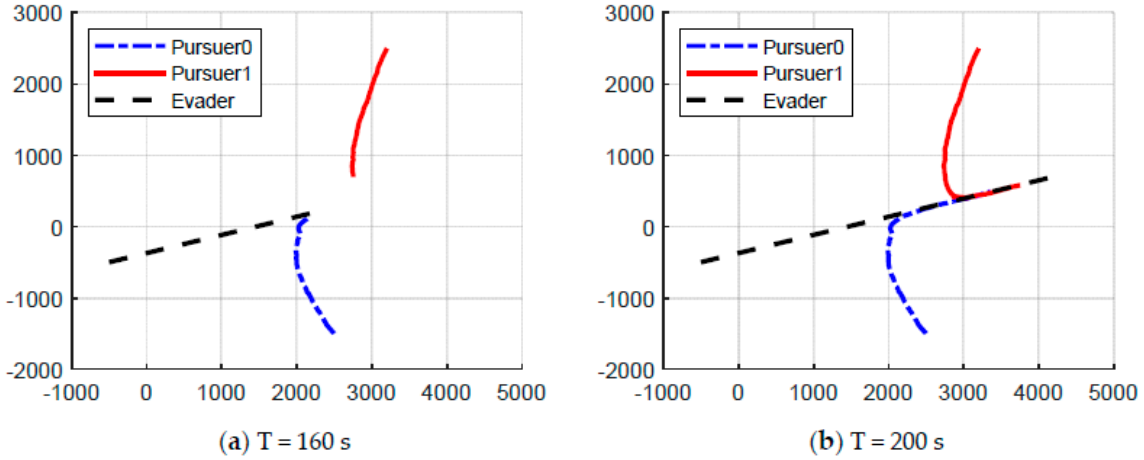



**Figure 11.** The traces of $P_0$, $P_1$ and $E_0$ under knowledge control with $\Delta v = 30\% v_P$.

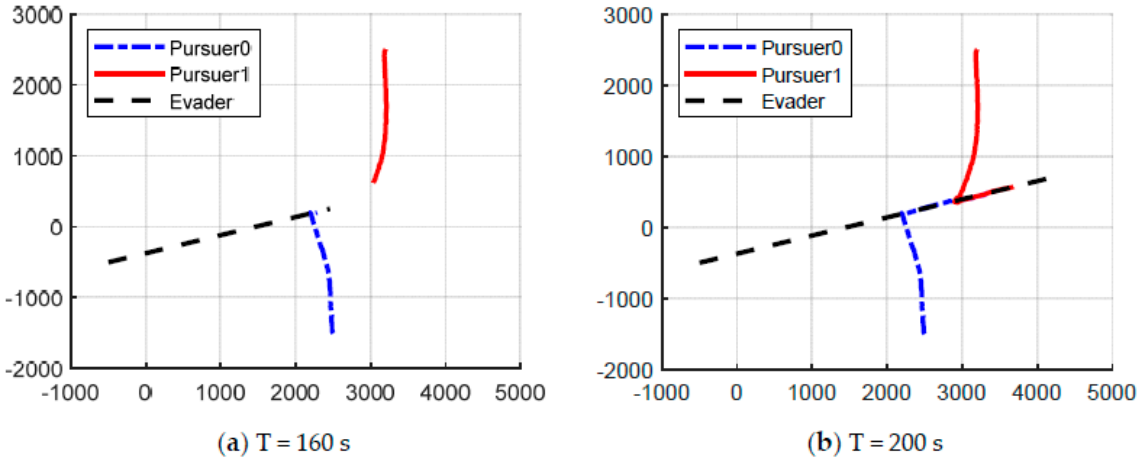
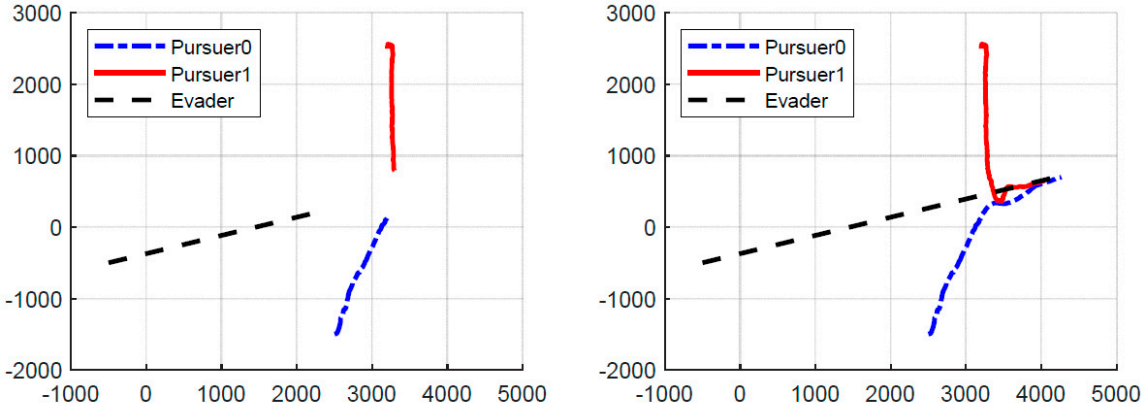



**Figure 12.** The traces of $P_0$, $P_1$ and $E_0$ under the proposed SK-FACL with $\Delta v = 30\% v_P$.

To further indicate the difference between the proposed SK-FACL and fuzzy actor–critic learning algorithm (FACL), another experiment was conducted to show the tracking traces under the SK-FACL and FACL.

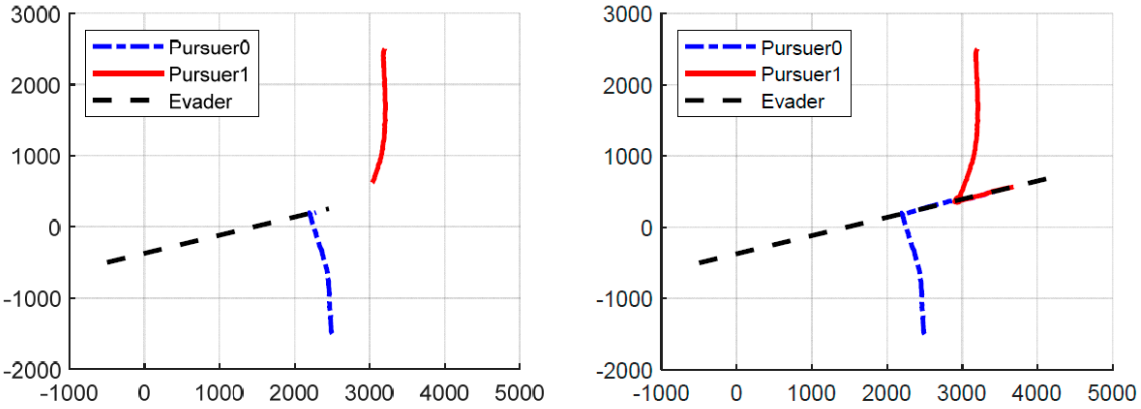
It is certainly theoretically feasible to eliminate the role of knowledge control and only optimize the policy using the purely data-driven FACL, and such a result is shown in Figure 13. Corresponding to the tracking performances in different time steps shown in Figure 13, Figure 14 shows the performances in the same steps under the SK-FACL. Since Figure 9 shows the ideal traces, the closer the tracking effect was to that shown in Figure 9, the better the tracking effect was. Figure 13 indicates that although $P_0$ and $P_1$ had the ability to track under FACL, the learned policy without the guidance of knowledge control was much weaker. Furthermore, after adding this guidance, Figure 14 shows the tracking performances under the SK-FACL were very close to the ideal ones shown in Figure 9, which indicates that the proposed SK-FACL could effectively correct the tracking traces and approach a nearly ideal capturing condition, even if facing the velocity estimation error $\Delta v$. Furthermore, it was noticed that although the trajectories shown in Figure 13 were similar to an interception, they were not optimal. This was because, for the 200 s tracking problem, not only the final tracking effect was considered but also the process tracking effect was considered. The situation where the pursuer was able to be near the evader at all time steps was seen as optimal because it maximized the pursuer's operational space. Therefore, although the trajectories of pursuers shown in Figures 9 and 14 were approximately perpendicular to that of the evader, they performed better because the pursuers were also as close to the evader as possible during the tracking process. Figure 15

shows the minimum distances between $P_0$, $P_1$ and $E_0$, along with the training times under the SK-FACL and FACL. The red lines and blue lines represent the minimum distances between $P_0$ and $E_0$ and between $P_1$ and $E_0$, respectively. It is seen that the SK-FACL could achieve the ideal performances after 50,000 training times, and the minimum average distance for $P_0$ was approximately 23.0, and for $P_1$, it was approximately 34.5. However, for the same number of training times, the FACL achieved the minimum average distance for $P_0$ was about 285.1, and for $P_1$, it was about 123.3. The results indicate that the performance under the SK-FACL was much better than under the FACL.




(**a**) T = 160 s (**b**) T = 200 s

**Figure 13.** The traces of $P_0$, $P_1$ and $E_0$ under the FACL with $\Delta v = 30\% v_P$.




(**a**) T = 160 s (**b**) T = 200 s

**Figure 14.** The traces of $P_0$, $P_1$ and $E_0$ under the proposed SK-FACL with $\Delta v = 30\% v_P$.

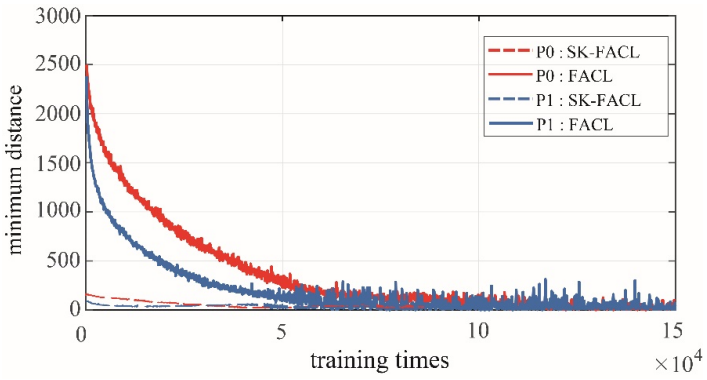
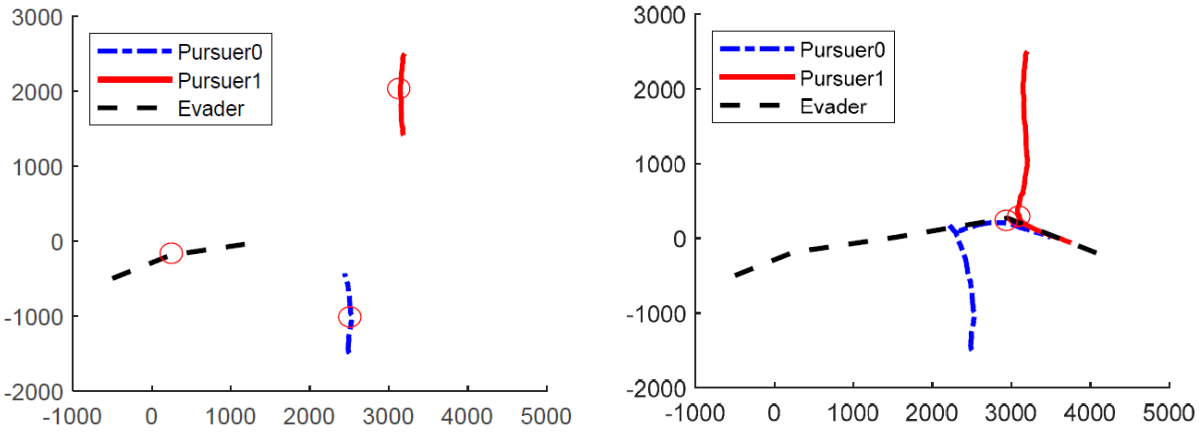



**Figure 15.** The minimum distances between P0, P1 and E0 along with the training times under the SK-FACL and FACL.

### *5.3. Case 3: Tracking a Smart Evader*

The results in case 2 show the superiority of the SK-FACL when the evader $E_0$ moves along a straight line. However, in reality, the evaders often have the ability to run when they see the pursuers getting close. Therefore, if the evader can suddenly turn in the game, it may affect the tracking results. In order to verify the adaptability of the proposed SK-FACL in this condition, $E_0$ can change its heading direction twice in this subsection, and we call this a "smart" evader. With the other settings the same as in Section 5.2, the simulated results were as follows.

Figure 16a shows the traces after the first run of $E_0$. It can be seen that when $E_0$ changed its heading angle, $P_0$ responded immediately and adjusted its trace in time according to the new trace of $E_0$. At the same time, $P_1$ also adjusted the trace due to this reason. However, $P_1$ was relatively far from $E_0$ when $E_0$ changed, and thus, the change in $P_1$ was not obvious enough. Moreover, Figure 16b shows the traces after the second run of $E_0$. Similar to the results in Figure 16a, $P_0$ and $P_1$ quickly respond to adapt to the new trajectory of $E_0$ and track it. From Figure 16, it is seen that the proposed SK-FACL had good generalization ability since it helped the agents adapt to the changes of evader, which was mainly due to the combination of the guidance of knowledge control and the policy iteration of the FACL.




(**a**) T = 160 s (**b**) T = 200 s

**Figure 16.** The traces of $P_0$, $P_1$ and $E_0$ under the SK-FACL when $E_0$ was smart.

In addition, in order to show the tracking accuracy of the proposed method, we compared the minimum tracking distances between the pursuer ($P_1$) and the evader $E_0$ under different $\Delta v$ values, as shown in Figure 17. The blue line shows the minimum distance based on pure knowledge control driven by Apollonius circles. In particular, $\Delta v = 0\%$ shows the theoretical possibility of the pursuer $P_1$ to capture the evader $E_0$ with pure knowledge. Furthermore, the red point shows the mean minimum distance under the SK-FACL when $\Delta v = 30\%$.

As can be seen from Figure 17, with the increase in $\Delta v$, the minimum tracking distance of pure knowledge control increased, which meant that the tracking accuracy decreased rapidly. The reason for this was mainly because of the inaccurate Apollonius circles. From the figure, it is seen that if $\Delta v = 0\%$, the minimum distance under knowledge control averaged at 34.0 and floated within [34.0~74.4]. When $\Delta v = 30\%$, the minimum distance under knowledge control was averaged at 220.6 and floated within [220.6~256.0], and under the SK-FACL, it was averaged at 31.8 and floated within [1.1~63.7], which is shown by the red point in the figure. Therefore, it is seen that the performance with $\Delta v = 30\%$ under the SK-FACL was nearly equivalent to that with $\Delta v = 0\%$ under knowledge control. This says that the SK-FACL could effectively improve the tracking performance on the basis of pure knowledge control and finally caused the pursuers to approach using an ideal tracking condition.

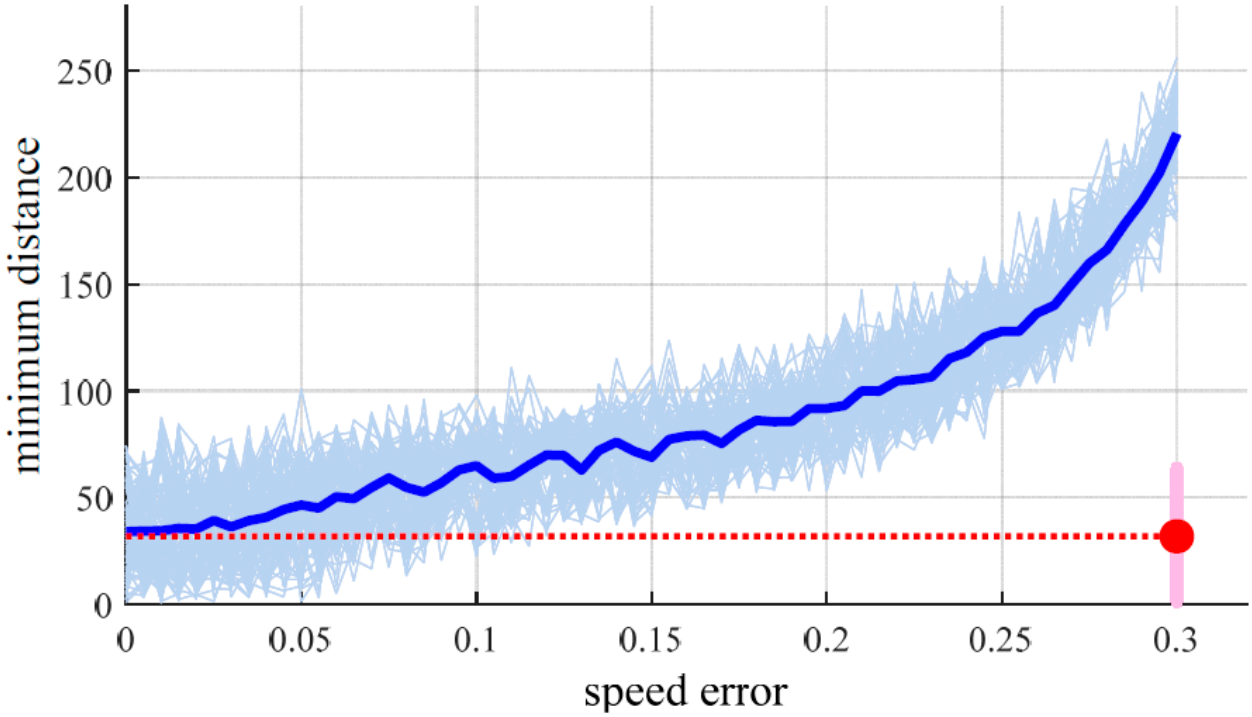



**Figure 17.** The minimum distance between the pursuer $P_1$ and evader with different $\Delta v$ values.

## 6. Conclusions

In this study, we proposed a novel reinforcement algorithm called the accelerating fuzzy actor–critic learning algorithm based on suboptimal knowledge (SK-FACL) to deal with the ground tracking problem. The proposed algorithm is mainly composed of two parts. The first part is a knowledge controller, which is obtained based on the Apollonius circle method. The second part is a policy iteration process driven by fuzzy actor–critic learning. In this novel SK-FACL, the knowledge controller helps the pursuers to start the learning process quickly and provides guided policies that enable the agents to learn faster. Moreover, the policy iteration part can modify the policy obtained from the inaccurate Apollonius circle, improving the tracking performance of the pursuers. The proposed algorithm provides a way to combine real model knowledge and data knowledge. Therefore, the proposed algorithm enables the agent to make full use of the model knowledge known by humans and improve the control strategy through data information. Hence, the intelligent method can be applied to a situation where the environmental model is particularly complex and difficult to model and the agent can only try a few attempts to achieve the optimal control strategy. The numerically simulated results show the advantages of the SK-FACL compared with pure knowledge control and pure policy iteration when there was velocity estimated error. Furthermore, SK-FACL also has the ability to deal with the condition where the evader suddenly turns during the tracking.

In the future, our team will continue to work on policy decisions in differential games through reinforcement learning, aiming at making the decision process more robust and reliable.

**Author Contributions:** Conceptualization, X.W., Z.M. and J.L.; Methodology, X.W., Z.M., L.M., K.S., X.H., C.F. and J.L.; Software, J.L.; Validation, X.W.; Writing—original draft, J.L.; Writing—review & editing, X.W. and J.L. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

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

## Nomenclature

| | |
|---|---|
| $P_i$ | Pursuers |
| $E_0$ | Evader |
| $\{x_{P_i}, y_{P_i}\}$ | Initial positions of the pursuers |
| $\{x_{E0}, y_{E0}\}$ | Initial position of the evader |
| $d_c$ | Capture distance |
| $v_{Pi}$ | Speed of pursuers |
| $v_E$ | Speed of evader |
| $\theta_{E0}$ | Heading angle of the evader |
| $t$ | Current time step |
| $\gamma$ | Discount factor |
| $r_t$ | Immediate reward |
| $\delta_t$ | TD error |
| $\omega^l(t)$ | Consequent parameter in the FLC |
| $\mu^{F_i^l}$ | Membership degree of input $x_i$ in the fuzzy rule $F_i^l$ |
| $\varepsilon$ | White noise |
| $\vartheta^l(t)$ | Consequent parameter of the critic |
| $\alpha_L$ | Learning rate for the critic |
| $\beta_L$ | Learning rate for the FLC |
| $\theta_{P0}$ | Guided policy for $P_0$ |
| $\theta_{P1}$ | Guided policy for $P_1$ |
| $a\prime_{P0}$ | Reinforcement learning policy for $P_0$ |
| $a'_{P1}$ | Reinforcement learning policy for $P_1$ |
| $\zeta$ | Constant scale factor |
| $r_i$ | Radius of the Apollonius circle |
| $O_i$ | Center of the Apollonius circle |
| $\alpha$ | Angle between the line of sight and the evader's heading direction |
| $\varphi$ | Angle that ensures the capture of the high-speed evader |
| $\hat{V}_p$ | Estimated speed for pursuer |
| $\hat{V}_e$ | Estimated speed for evader |
| $\boldsymbol{s}_t$ | State vector |
| $s_1$ | Relative distance between $P_i$ and $E_0$ |
| $s_2$ | Heading angle difference between $P_i$ and $E_0$ |
| $s_3$ | Derivative of the angle difference |
| $d_{k-1}(s_{t-1})$ | Relative distance at $s_{t-1}$ in the $(k-1)$th training |
| $d_{k-1}(s_t)$ | Relative distance at $s_t$ in the $(k-1)$th training |
| $d_k(m)$ | Minimum distance between $P_i$ and $E_0$ in the $k$th training |
| $d_{k-1}(m)$ | Minimum distance between $P_i$ and $E_0$ in the $(k-1)$th training |
| $r_t$ | Immediate reward |
| $\{v_1, v_2, v_3, v_4\}$ | Coefficients in reward design |

## References


1. Chen, J.; Zha, W.; Peng, Z.; Gu, D. Multi-player pursuit–evasion games with one superior evader. *Automatica* **2016**, *71*, 24–32. [CrossRef]
2. Fang, B.F.; Pan, Q.S.; Hong, B.R.; Lei, D.; Zhong, Q.B.; Zhang, Z. Research on High Speed Evader vs. Multi Lower Speed Pursuers in Multi Pursuit-evasion Games. *Inf. Technol. J.* **2012**, *11*, 989–997.
3. Feng, J. Development tendency and key technology of system intellectualization. *Mod. Def. Technol.* **2020**, *48*, 1–6.
4. Ma, X.; Dai, K.; Li, M.; Yu, H.; Shang, W.; Ding, L.; Zhang, H.; Wang, X. Optimal-Damage-Effectiveness Cooperative-Control Strategy for the Pursuit–Evasion Problem with Multiple Guided Missiles. *Sensors* **2022**, *22*, 9342. [CrossRef]
5. Rilwan, J.; Ferrara, M.; Ja'afaru, A.; Pansera, B. On pursuit and evasion game problems with Grönwall-type constraints. *Qual. Quant.* **2023**. [CrossRef]
6. Liu, H.; Wu, K.; Huang, K.; Cheng, G.; Wang, R.; Liu, G. Optimization of large-scale UAV cluster confrontation game based on integrated evolution strategy. *Clust. Comput.* **2023**, 1–15. [CrossRef]
7. Souli, N.; Kolios, P.; Ellinas, G. Multi-Agent System for Rogue Drone Interception. *IEEE Robot. Autom. Lett.* **2023**, *8*, 2221–2228. [CrossRef]
8. Forestiero, A. Bio-inspired algorithm for outliers detection. *Multimed. Tools Appl.* **2017**, *76*, 25659–25677. [CrossRef]

9. Forestiero, A. Heuristic recommendation technique in Internet of Things featuring swarm intelligence approach. *Expert Syst. Appl.* **2021**, *187*, 115904. [CrossRef]
10. Dimeas, A.; Hatziargyriou, N. Operation of a Multiagent System for Microgrid Control. *IEEE Trans. Power Syst.* **2005**, *20*, 1447–1455. [CrossRef]
11. Burgos, E.; Ceva, H.; Perazzo, R.P.J. Dynamical quenching and annealing in self-organization multiagent models. *Phys. Rev. E* **2001**, *64*, 016130. [CrossRef] [PubMed]
12. Lin, Z.; Wang, L.; Han, Z.; Fu, M. Distributed Formation Control of Multi-Agent Systems Using Complex Laplacian. *IEEE Trans. Autom. Control* **2014**, *59*, 1765–1777. [CrossRef]
13. Flores-Resendiz, J.F.; Avilés, D.; Aranda-Bricaire, E. Formation Control for Second-Order Multi-Agent Systems with Collision Avoidance. *Machines* **2023**, *11*, 208. [CrossRef]
14. Do, H.; Nguyen, H.; Nguyen, V.; Nguyen, M.; Nguyen, M.T. Formation control of multiple unmanned vehicles based on graph theory: A Comprehensive Review. *ICST Trans. Mob. Commun. Appl.* **2022**, *7*, e3. [CrossRef]
15. Zhang, X.; Sun, J. Almost equitable partitions and controllability of leader–follower multi-agent systems. *Automatica* **2021**, *131*, 109740. [CrossRef]
16. Zhang, X.; Xie, S.; Tao, Y.; Li, G. A robust control method for close formation of aerial-refueling UAVs. *Acta Aeronaut. Astronaut. Sin.* **2023**. [CrossRef]
17. Sun, F.; Wang, F.; Liu, P.; Kurths, J. Robust fixed-time connectivity preserving consensus of nonlinear multi-agent systems with disturbance. *Int. J. Robust Nonlinear Control* **2021**, *32*, 1469–1486. [CrossRef]
18. Sutton, R.; Barto, A. *Reinforcement Learning: An Introduction*; MIT Press: Cambridge, MA, USA, 1998.
19. Doroodgar, B.; Nejat, G. A hierarchical reinforcement learning based control architecture for semi-autonomous rescue robots in cluttered environments. In Proceedings of the 2010 IEEE International Conference on Automation Science and Engineering, Toronto, ON, Canada, 21–24 August 2010.
20. Barros, P.; Yalçın, N.; Tanevska, A.; Sciutti, A. Incorporating rivalry in reinforcement learning for a competitive game. *Neural Comput. Appl.* **2022**, 1–14. [CrossRef]
21. Sniehotta, F.F. Towards a theory of intentional behaviour change: Plans, planning, and self-regulation. *Br. J. Health Psychol.* **2009**, *14*, 261–273. [CrossRef]
22. Sewak, M. *Temporal Difference Learning, SARSA, and Q-Learning: Some Popular Value Approximation Based Reinforcement Learning Approaches*; Springer: Singapore, 2019.
23. Woeginger, G.J. Exact Algorithms for NP-Hard Problems: A Survey. In Proceedings of the Combinatorial Optimization-Eureka, You Shrink!, Papers Dedicated to Jack Edmonds, International Workshop, Aussois, France, 5–9 March 2001.
24. Cui, Y.; Zhu, L.; Fujisaki, M.; Kanokogi, H.; Matsubara, T. Factorial Kernel Dynamic Policy Programming for Vinyl Acetate Monomer Plant Model Control. In Proceedings of the 14th IEEE International Conference on Automation Science and Engineering, Munich, Germany, 20–24 August 2018.
25. Wang, X.; Shi, P.; Zhao, Y.; Sun, Y. A Pre-Trained Fuzzy Reinforcement Learning Method for the Pursuing Satellite in a One-to-One Game in Space. *Sensors* **2020**, *20*, 2253. [CrossRef] [PubMed]
26. Neu, G.; Szepesvari, C. Apprenticeship Learning using Inverse Reinforcement Learning and Gradient Methods. *arXiv* **2012**, arXiv:1206.5264.
27. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. [CrossRef]
28. Liu, M.; Zhu, Y.; Zhao, D. An Improved Minimax-Q Algorithm Based on Generalized Policy Iteration to Solve a Chaser-Invader Game. In Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN), Glasgow, UK, 19–24 July 2020.
29. Vamvoudakis, K.G. Non-zero sum Nash Q-learning for unknown deterministic continuous-time linear systems. *Automatica* **2015**, *61*, 274–281. [CrossRef]
30. Lin, K.; Zhao, R.; Xu, Z.; Zhou, J. Efficient Large-Scale Fleet Management via Multi-Agent Deep Reinforcement Learning. *ACM* **2018**, 1774–1783. [CrossRef]
31. Chi, C.; Ji, K.; Song, P.; Marahatta, A.; Zhang, S.; Zhang, F.; Qiu, D.; Liu, Z. Cooperatively Improving Data Center Energy Efficiency Based on Multi-Agent Deep Reinforcement Learning. *Energies* **2021**, *14*, 2071. [CrossRef]
32. Li, B.; Yue, K.Q.; Gan, Z.G.; Gao, P.X. Multi-UAV Cooperative Autonomous Navigation Based on Multi-agent Deep Deterministic Policy Gradient. *Yuhang Xuebao J. Astronaut.* **2021**, *42*, 757–765.
33. Wang, X.; Shi, P.; Wen, C.; Zhao, Y. Design of Parameter-Self-Tuning Controller Based on Reinforcement Learning for Tracking Noncooperative Targets in Space. *IEEE Trans. Aerosp. Electron. Syst.* **2020**, *56*, 4192–4208. [CrossRef]
34. Wang, X.; Shi, P.; Schwartz, H.; Zhao, Y. An algorithm of pretrained fuzzy actor–critic learning applying in fixed-time space differential game. *Proc. Inst. Mech. Eng. Part G J. Aerosp. Eng.* **2021**, *235*, 2095–2112. [CrossRef]
35. Wang, K.; Xing, R.; Feng, W.; Huang, B. A Method of UAV Formation Transformation Based on Reinforcement Learning Multi-agent. In Proceeding of the 2021 International Conference on Wireless Communications, Networking and Applications, Hangzhou, China, 13–15 August 2021. [CrossRef]
36. Xu, D.; Chen, G. Autonomous and cooperative control of UAV cluster with multi-agent reinforcement learning. *Aeronaut. J.* **2022**, *126*, 932–951. [CrossRef]

37. Cardarilli, G.C.; Di Nunzio, L.; Fazzolari, R.; Giardino, D.; Re, M.; Ricci, A.; Spanò, S. An FPGA-based multi-agent Reinforcement Learning timing synchronizer. *Comput. Electr. Eng.* **2022**, *99*, 107749. [CrossRef]
38. Wang, X.; Shi, P.; Wen, C.; Zhao, Y. An Algorithm of Reinforcement Learning for Maneuvering Parameter Self-Tuning Applying in Satellite Cluster. *Math. Probl. Eng.* **2020**, *2020*, 1–17. [CrossRef]
39. Dorothy, M.; Maity, D.; Shishika, D.; Von Moll, A. One Apollonius Circle is Enough for Many Pursuit-Evasion Games. *arXiv* **2021**, arXiv:2111.09205.