2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Collaboration Strategies for Two Heterogeneous Pursuers in A Pursuit-Evasion Game Using Deep Reinforcement Learning

Zhanping Zhong[1] , Zhuoning Dong[1], Xiaoming Duan[2] and Jianping He[2]

***Abstract*— We investigate a pursuit-evasion game taking place in an unbounded three-dimensional space, where a flexible pursuer with hybrid dynamics collaborates with a fast pursuer and aims to capture a flexible evader within a finite time. The key feature of this problem lies in the hybrid dynamics of the flexible pursuer, which can change its dynamics once during the game and switch to a fast pursuer with increased speed but lower maneuverability. To address this challenge, we devise a hybrid strategy based on the soft actor-critic framework, tailored specifically for the flexible pursuer, which encompasses both maneuvering and switch tactics. We introduce a switch factor to the input of the actor network and incorporate switch actions to further expand the action space. These additions enable the flexible pursuer to execute maneuvering actions and determine a moment to switch to a fast pursuer. The reward function is designed to account for related angle, altitude, speed, and sparse reward. Through extensive ablation experiments conducted in a simulated environment, we demonstrate the efficacy of our algorithm in facilitating the learning of hybrid strategies for the flexible pursuer, resulting in significantly improved capture rates compared to alternative methods.**

I. INTRODUCTION

In a pursuit-evasion game, typically involving two teams with one or more players each, the pursuers aim to capture the evaders as quickly as possible, while the evaders strive to evade capture. Developing efficient pursuit or escape strategies is a central focus of pursuit-evasion game research. It was initially conceptualized by R. Isaacs in his book, known as the *Homicidal Chauffeur* problem [1], and subsequently, researchers delved into this concept to determine the most effective strategies for the players in a game resembling hide and seek [2].

An extension of the *Homicidal Chauffeur* is the *Two Cars* differential game, in which two players, each controlling a vehicle with a restricted turning radius. Meier in [3] explored the *Two Cars* problem where both players' vehicles shared an identical minimum turning radius constraint, and the pursuer had a lower maximum speed compared to the evader. Subsequently, the pursuit-evasion game was introduced into three-dimensional (3D) space, including common aerial, surface, and underwater adversarial environments. In these environments, players also have constraints on the turning radius, and with the increase in dimensions, the state space, action space, and others become more complex. Shinar and Gutman derived an analytical closed-form solution for a 3D pursuit-evasion game between a missile and an aircraft [4]. Pachter and Milch formulated a two-player game incorporating the dynamics of ships [5]. In certain special games, players may switch their dynamics at certain moments. Shinar et al. investigated pursuit-evasion games involving players with hybrid dynamics and demonstrated the winning region can be expanded when dynamics are switched at the right moment [6]. Additionally, Yan et al. further explored more complex scenarios, and proposed strategies for a planar reach-avoid game where a Dubins car pursuer defends against an evader, offering conditions for guaranteed success and extending the approach to multiplayer scenarios in [7].

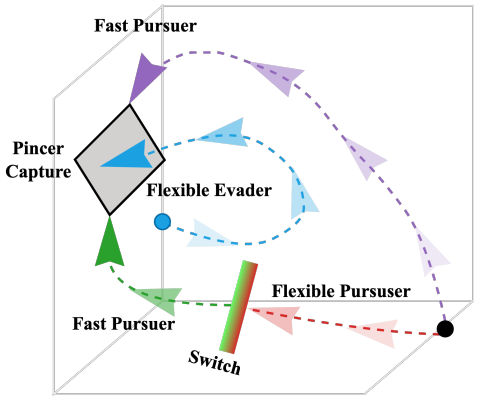


Fig. 1. Schematic diagram of the pursuit-evasion game. A flexible pursuer and a fast pursuer start from the same position to capture a flexible evader at a distance. The flexible pursuer has the opportunity to switch to a fast pursuer at some point in time, and the evader is finally captured by two fast pursuers performing a pincer movement.

In recent years, deep reinforcement learning (DRL) has been extensively applied to the decision-making process in pursuit-evasion games. DRL synergizes the powerful representational capabilities of deep learning with the excellent decision-making prowess of reinforcement learning [8]. It can tackle the challenges of situational awareness and action decision-making in games, thereby enabling agents to exhibit superior situational forecasting and action decision-making abilities. To study the impact of hybrid dynamic pursuers on the game and enhance the capture rate of the pursuer team, we propose a DRL algorithm in this paper to provide a hybrid strategy for a flexible pursuer with hybrid dynamics which can switch to a fast pursuer collaborating with another fast pursuer. While the fast pursuer utilizes its higher speed

This work was supported in part by Shanghai Pujiang Program under Grant 22PJ1404900, the Natural Science Foundation of Shanghai under Grant 23ZR1428900, and the National Natural Science Foundation of China under Grant 62373247 and 62303314.

[1]The School of Automation Science and Electrical Engineering, Beihang University, Beijing, China. Email: {zzp, dongzhuoning}@buaa.edu.cn

[2]The Department of Automation, Shanghai Jiao Tong University, and Key Laboratory of System Control and Information Processing, Ministry of Education of China, Shanghai, China. Email: {xduan, jphe}@sjtu.edu.cn

to close in on the evader, the flexible pursuer leverages its agile maneuverability to seek more advantageous positions and later switches to a fast pursuer at a critical moment to execute a pincer capture on the evader. The general process of the pursuit-evasion game is illustrated in Fig. 1. The contributions of this paper are outlined as follows

- We devise a two-on-one pursuit-evasion game, which includes a flexible pursuer with the ability to switch to a fast pursuer at a specific moment, another fast pursuer, and a flexible evader.
- We design a hybrid strategy for the flexible pursuer based on the soft actor-critic (SAC) framework. A switch factor is introduced to represent the switch inclination of the flexible pursuer in the current state, which is concatenated with state vectors to serve as inputs to the actor network. Additionally, switch actions are introduced to expand the action space, which enables the flexible pursuer to not only select actions to seek advantageous positions but also autonomously switch to a fast pursuer, flanking the evader with another fast pursuer. Furthermore, we also design a reward function related to angle, altitude, and speed, along with sparse reward, to incentivize the flexible pursuer.
- We construct a simulation environment and agent models that closely align with physical reality. Through ablation experiments, we demonstrate the effectiveness of the maneuvering and switching strategies in the hybrid approach compared to traditional fixed strategies. Extensive simulations validate that our hybrid strategy can significantly improve the capture rate.

II. RELATED WORKS

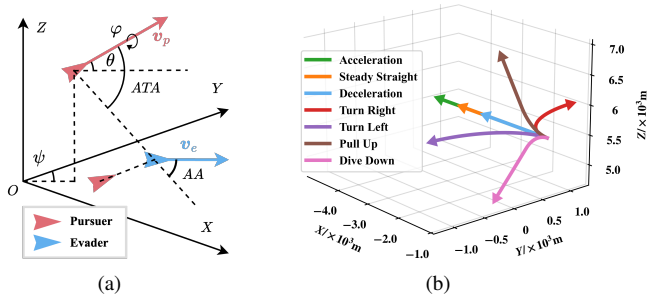
The main objective of this study is to provide a hybrid strategy for a flexible pursuer with hybrid dynamics in 3D space using DRL, so we survey related works on three specific topics, including pursuit-evasion games in 3D space, pursuit-evasion games with hybrid dynamics and 3D pursuit-evasion games in 3D space using DRL.

Pursuit-Evasion in 3D Space. In 3D environments, the motion space for both the pursuer and evader is more open, and the action strategies are more diverse and complex. Compared to two-dimensional planes, the third dimension of height adds an extra degree of freedom, making actions during the pursuit-evasion process more flexible and variable. Some typical 3D pursuit-evasion scenarios include aerial combat, reach-avoid games, and spacecraft tracking. Shinar explored a realistic engagement scenario involving a missile targeting an aircraft, as well as air-to-air scenarios, by employing variational methods [9]. Greenwood formulated a realistic 3D differential game by modeling fighter aircraft dynamics in [10], where the dynamics of two fighter aircraft in space are incorporated, and firing envelopes are factored in as part of the barrier analysis. In [11], a differential game was proposed that involved a pursuit-evasion engagement between a missile and an aircraft. The game formulation employed a nonlinear miss-distance as the payoff functional. Yan et al. presented a 3D reach-avoid game with three pursuers and one evader, offering a closed-form barrier to predict the game's outcome and an efficient algorithm for real-time updates with minimal computational demands in [12]. They also presented a strategy for a 3D multiplayer reach-avoid game, where pursuers with different speeds and capture radii defend a goal region by capturing evaders, using subgame decomposition, a polynomial-time approximation algorithm, and receding horizon optimization to maximize the number of captured evaders in [13]. Yan et al. proposed a constrained optimal lateral evasion guidance strategy for hypersonic vehicles against coordinated pursuers, ensuring overload limits while achieving effective evasion through specified miss distances and direction reversal maneuvers in [14].

Pursuit-Evasion with Hybrid Dynamics. In some cases, the dynamics of the pursuer during the pursuit process may switch at certain moments. J. Shinar extensively studied the properties of pursuers with hybrid dynamics in the pursuit-evasion process in two-dimensional space in [15], introducing impulse dynamics to rigorously demonstrate the existence of a saddle point solution in the differential game problem. This indicates that pursuers with hybrid dynamics should strive to transition to a mode with better dynamics and higher consumption during the pursuit. In [16], Glizer studied the cases where at least for one dynamic mode of the pursuer the capture zone is bounded, and conditions for increasing the pursuer's hybrid capturability are derived for these instances, with respective capture zones constructed.

Pursuit-Evasion Using DRL in 3D Space. The applications of DRL in 3D pursuit-evasion games are primarily concentrated in the field of aerial engagements. Cao et al. combined the double deep Q network (DDQN) and the minimax algorithm to provide optimal maneuver decisions for the pursuer in one-on-one air combat [17]. Li et al. improved the actor-critic framework by introducing gated recurrent units in the actor and utilizing an attention mechanism to design a centralized critic, providing cooperative strategies for multiple unmanned vehicles [18]. Liu et al. introduced the multi-agent proximal policy optimization (MAPPO) to provide new tactical strategies for the pursuers in a two-on-one pursuit-evasion [19]. Fan et al. proposed an asynchronous advantage actor-critic (A3C) method for autonomous unmanned aerial vehicle maneuver decisions in air combat scenarios, utilizing neural networks trained via multi-threaded environment interaction to obtain optimal strategies [20]. Zhang et al. investigated pursuit-evasion games in urban obstacle environments, employing the multi-agent deep deterministic policy gradient (MADDPG) algorithm and extensions to provide motion planning for teams of pursuers [21].

Distinct from existing works, we introduce DRL for a flexible pursuer with hybrid dynamics in 3D pursuit-evasion games. Guided by the SAC framework, we offer hybrid strategies for the flexible pursuer to achieve coordinated pursuit and dynamic switching simultaneously.



(a) (b)

Fig. 2. (a): Schematic diagram of two-on-one pursuit-evasion confrontation. (b): Schematic diagram of the seven basic maneuvers.

III. METHODOLOGY

In the subsequent subsections, we will first outline the problem formulation and then introduce the reward function and the DRL algorithm devised to address the problem.

A. Problem Formulation

We consider a pursuit-evasion game in 3D space, featuring a flexible pursuer with hybrid dynamics and a fast pursuer with fixed dynamics. Both pursuers commence from the same location, working together to capture a distant evader with fixed dynamics. More specifically,

- **The flexible pursuer** possesses superior maneuverability, with a smaller turning radius under identical speed conditions.
- **The fast pursuer** has a higher speed but lacks agility, exhibiting significantly weaker turning capabilities compared to the flexible pursuer.
- **The flexible evader** shares similar capabilities with the flexible pursuer: although its speed is noticeably slower than that of the fast pursuer, it can evade pursuit by employing sharp turns.

We adopt the point mass model for our agents in 3D space. Firstly, we establish a ground inertial coordinate system in 3D space, as illustrated in Fig. 2(a). In this figure, the X-axis points eastward, the Y-axis points northward, and the Z-axis is vertical upwards. The coordinate (x, y, z) denotes the position of the agent in the 3D coordinate system. Additionally, $\boldsymbol{v}$ represents the velocity vector of the agent, where $v = |\boldsymbol{v}|$, while ψ, θ, and φ denote the yaw, pitch, and roll angles of the agent, respectively. The state s can thus be represented by these 7 parameters

$$s = \left[x, y, z, \theta, \psi, \varphi, \boldsymbol{v}\right]. \tag{1}$$

We use AA to represent the aspect angle and ATA to represent the antenna train angle, which can be calculated as follows

$$\begin{cases} AA = \arccos \frac{\boldsymbol{v}_e \cdot \boldsymbol{d}}{|\boldsymbol{v}_e||\boldsymbol{d}|}, \\ ATA = \arccos \frac{\boldsymbol{v}_p \cdot \boldsymbol{d}}{|\boldsymbol{v}_p||\boldsymbol{d}|}, \end{cases} \tag{2}$$

where $\boldsymbol{v}_p, \boldsymbol{v}_e$ represent the velocities of the pursuer and the evader respectively, and the distance vector from the pursuer to the evader is denoted by $\boldsymbol{d} = [x_p - x_e, y_p - y_e, z_p - z_e]$.

To investigate maneuvering strategies, we introduce the following dynamic models for the agents

$$\begin{cases} \dot{\theta} = \frac{g}{v}\left(N_z \cos\varphi - \cos\theta\right), \\ \dot{v} = g\left(N_x - \sin\theta\right), \\ \dot{\psi} = \frac{gN_z \sin\varphi}{v\cos\theta}, \end{cases} \tag{3}$$

where g represents gravitational acceleration, N_x denotes the tangential overload of the agent, and N_z represents the normal overload of the agent. We represent the action taken by an agent during the game with

$$a = [N_x, N_z, \varphi]. \tag{4}$$

The action space encompasses seven fundamental maneuvers advocated by the National Aeronautics and Space Administration (NASA), which are widely utilized in UCAV operations [22]. These maneuvers include steady straight flight, acceleration, deceleration, left and right turns, as well as pull up and pitch down, illustrated in Fig. 2(b). We represent these seven maneuver actions by discretely encoding the three elements within each action. Essentially, the triplet $[N_x, N_z, \varphi]$ serves as the control variable for the seven discrete actions in the simulation experiments. It is assumed that the tangential overload N_x, normal overload N_z, and roll angle φ remain constant during maneuver selection. The discrete encoding of the maneuver actions for both the flexible pursuer and the fast pursuer is shown in Table I. It is worth noting that the flexible evader shares the same action encoding as the flexible pursuer, which is not detailed in the table.

TABLE I

THE DISCRETE ENCODING OF MANEUVER ACTIONS FOR THE FLEXIBLE PURSUER AND THE FAST PURSUER

Actions	Flexible Pursuer			Fast Pursuer		
	N_{x_1}	N_{z_1}	φ_1	N_{x_2}	N_{z_2}	φ_2
Steady Straight	0	+1	0	0	+1	0
Acceleration	+2	+1	0	+5	+1	0
Deceleration	−2	+1	0	−5	+1	0
Turn Right	0	+7	$-\pi/3$	0	+4	$-\pi/3$
Turn Left	0	−7	$+\pi/3$	0	−4	$+\pi/3$
Pull Up	0	+7	0	0	+4	0
Dive Down	0	−7	0	0	−4	0

Our work differs from typical 3D pursuit-evasion games in that the pursuer team includes a flexible pursuer with hybrid dynamics, who has one opportunity to switch to a fast pursuer. Similar to [15], the flexible pursuer chooses to switch to a fast pursuer during the game. Given the initial distance D_{init} and the finite time constraint $T_{\max}$, the flexible pursuer is unable to approach the evader without switching dynamics within the limited time. The game would degenerate into a one-on-one pursuit between the fast pursuer and the flexible evader with an extremely low capture rate, as we will demonstrate in subsequent experiments. In the game, if the flexible pursuer makes a switch, then we denote this time point as T_{s}, which is called the switch point.

Next, we formulate this pursuit-evasion game as a Markov Decision Process (MDP). An MDP M is defined by a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$ where

- The state space is represented as $\mathcal{S} = \mathcal{S}_1 \times \mathcal{S}_2 \times \mathcal{S}_3$, where $\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3$ denote the state space of the flexible pursuer, the fast pursuer and the flexible evader, and s_1^t, s_2^t, s_3^t denote their states at time t, respectively. In addition, we concatenate the three states together as $s^t = [s_1^t, s_2^t, s_3^t]$, representing the global state of the whole system at time t.
- The joint action space is represented as $\mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_2 \times \mathcal{A}_3$, where $\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3$ denote the action space of the flexible pursuer, the fast pursuer and the flexible evader, and a_1^t, a_2^t, a_3^t denote their actions at time t, respectively.
- The transition kernel is represented as $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \mapsto [0, 1]$, which determines the probability of the next state s_{t+1} given the current state s_t and all agents' actions.
- The reward function is represented as $\mathcal{R} : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$. In this paper, we only consider the global reward of the pursuer team.
- The discount factor is represented as $\gamma \in (0, 1)$.

An episode starts at time $t = 0, t \in \mathbb{N}$, and it ends when the finite time $T_{\max}$ is reached. For clarity and conciseness, the flexible pursuer both before and after the dynamic switching has a subscript 1. The winning condition of the pursuer team is satisfied if any one of the pursuers successfully captures the evader, and the capture condition for one pursuer is as follows:

$$C := (AA \leq q_{\max}) \wedge (ATA \leq p_{\max}) \wedge (|\boldsymbol{d}| \leq D_{\mathrm{c}}), \quad (5)$$

where AA and ATA are defined in (2), $q_{\max}$ is the maximum locking aspect angle, $p_{\max}$ is the maximum locking antenna train angle, and D_{c} is referred to as the capture distance. Then the overall winning condition can be represented as:

$$W := C_1 \vee C_2, \quad (6)$$

where C_1 and C_2 represent the capture conditions for the two pursuers, respectively.

Our objective is to utilize DRL algorithms to devise a hybrid strategy for the flexible pursuer. This strategy should equip the pursuer with maneuvering decisions and empower it to autonomously select a suitable switch point T_{s}, with the goal of maximizing the capture rate of the pursuer team.

B. Reward Function

The reward function designed for the flexible pursuer integrates both the sparse reward mechanism and the intrinsic reward mechanism. To incentivize the flexible pursuer towards achieving advantageous positions, the intrinsic reward system comprises angle reward r_a, altitude reward r_h, and speed reward r_v. The angle reward of a single pursuer is defined as

$$r_a = 1 - \frac{AA_1 + ATA_1}{\pi}, \quad (7)$$

where AA_1 and ATA_1 respectively represent the aspect angle and antenna train angle of the flexible pursuer with respect to the evader.

Since we aim to encourage the pursuers to achieve a pincer movement on the evader through the design of the reward function, we provide the altitude reward as follows

$$r_h = \begin{cases} 2, & 2D_{\mathrm{c}} \leq z_1 + z_2 - 2z_3 \leq 6D_{\mathrm{c}}, \\ -1, & \text{others}, \end{cases} \quad (8)$$

where z_1, z_2, z_3 represent the altitude of the flexible pursuer, the fast pursuer, and the evader, respectively, and D_{c} is the capture distance as in (5).

The speed reward aims to leverage the advantage of different speeds of the flexible pursuer before and after the switching. We want the flexible pursuer to seek a more advantageous position before switching and then leverage its explosive speed advantage to chase the evader after T_{s}. The speed reward of the flexible pursuer is defined as follows

$$r_v = \frac{(V_{\min} + V_{\max}) - 2(v_1 - v_3)}{V_{\max} - V_{\min}} - 1, \quad (9)$$

where $[V_{\min}, V_{\max}]$ is the speed range of the flexible pursuer and the evader, and v_1, v_3 denotes the velocity of the flexible pursuer and the evader, respectively.

Finally, we introduce a sparse reward r_e at time t as

$$r_e = \begin{cases} 0, & t < T_{\mathrm{e}}, \\ 10, & t = T_{\mathrm{e}} \text{ and } W, \\ -10, & t = T_{\mathrm{e}} \text{ and } \neg W, \end{cases} \quad (10)$$

where T_{e} is the end time of the game, and C_1 and C_2 represent the capture conditions for the flexible pursuer and the fast pursuer, respectively.

In summary, the reward obtained by pursuers at time t can be expressed by the following formula

$$r_t = w_a r_a^t + w_h r_h^t + w_v r_v^t + r_e^t, \quad (11)$$

where $w_a \geq 0$, $w_h \geq 0$, and $w_v \geq 0$ with $w_a + w_h + w_v = 1$ serve as weights that can be used to adjust the importance of different rewards.

C. Algorithm

We adopt the SAC framework [23] in this work, and by modifying the input and output of the actor network, we obtain a hybrid strategy that includes both maneuvering and switching strategies. In previous works, DRL has been employed to tackle pursuit-evasion problems, where deep neural networks typically take as input the state or observation vectors of agents, including physical information such as position and velocity. In our study, to enable the flexible pursuer to autonomously switch dynamics, we introduce decision points into the input of deep neural networks, referred to as "switching factors". We draw inspiration from similar techniques in the NLP field, where embedding enhancement methods add dimensions to text embeddings to represent categories such as sentiment (e.g., positive, negative, neutral) in sentiment analysis tasks or entity types (e.g., person names, locations, organizations) in named entity recognition tasks to aid the model in entity recognition and classification. Specifically, we augment the state vector s with an additional

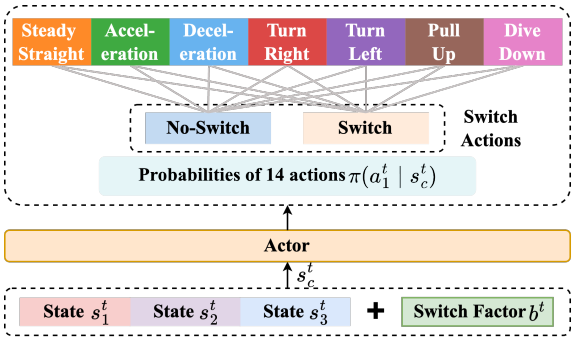


Fig. 3. The state vectors s_1 for the flexible pursuer, s_2 for the fast pursuer, and s_3 for the evader are concatenated with the switch factor b to form the input to the actor network. By combining 7 maneuver actions with two switch actions, the action space is expanded to 14 actions. The actor network outputs the probabilities of these 14 actions

parameter $b = [b_1, b_2]$, where $b_1, b_2 \in [0, 1]$ are called the switch factors, to represent the current state transition tendency. Through the concatenation operation, the state vector input to the actor network can be represented as

$$s_c^t = [s^t, b]. \tag{12}$$

Simultaneously, the action space of the flexible pursuer is expanded from the original 7 actions to incorporate two additional switch actions, namely "No-Switch" and "Switch", as illustrated in Fig. 3. Specifically, each action now incorporates an additional parameter indicating whether to switch or not, resulting in an expanded set of 14 actions. After the action selection in each step, the switch factor b is updated according to the switch actions as follows

$$\begin{cases} b_1 \leftarrow \frac{b_1+1}{b_1+b_2+1}, b_2 \leftarrow \frac{b_2}{b_1+b_2+1} & \text{No-Switch,} \\ b_1 \leftarrow \frac{b_1}{b_1+b_2+1}, b_2 \leftarrow \frac{b_2+1}{b_1+b_2+1} & \text{Switch.} \end{cases} \tag{13}$$

We present the overall framework in Fig. 4. We use the SAC framework as the decision model for the flexible pursuer. This framework comprises 1 actor $\pi_\phi(s_c^t, a_1^t)$, 2 V critics $V_\alpha(s_c^t), V_{\bar{\alpha}}(s_c^t)$, and 2 Q critics $Q_{\beta_1}(s_c^t, a_1^t), Q_{\beta_2}(s_c^t, a_1^t)$, making a total of 5 networks, where $\phi, \alpha, \bar{\alpha}, \beta_1, \beta_2$ represent the parameters of each network, respectively.

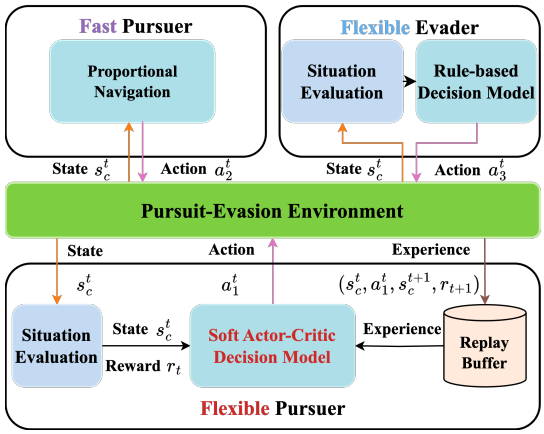


Fig. 4. The overall algorithm framework of the pursuit-evasion problem

The SAC algorithm introduces the concept of maximum entropy in addition to maximizing future accumulated rewards. The purpose of incorporating entropy is to enhance the robustness and the exploration capability of the agent. The objective function of the SAC algorithm is

$$J(\pi) = \sum_{t=0}^{T_e} E_{\left(s_c^t, a_1^t\right) \sim \rho^\pi} \left[r\left(s_c^t, a_1^t\right) + \eta H\left(\pi\left(\cdot \mid s_c^t\right)\right) \right], \tag{14}$$

where ρ^π represents the distribution of (s_t, a_t) under the policy π, $H(\cdot)$ denotes the entropy, and η is the entropy regularization coefficient used to control the importance of entropy. The entire training process is summarized in Algorithm 1.

Algorithm 1: Training process of flexible pursuer

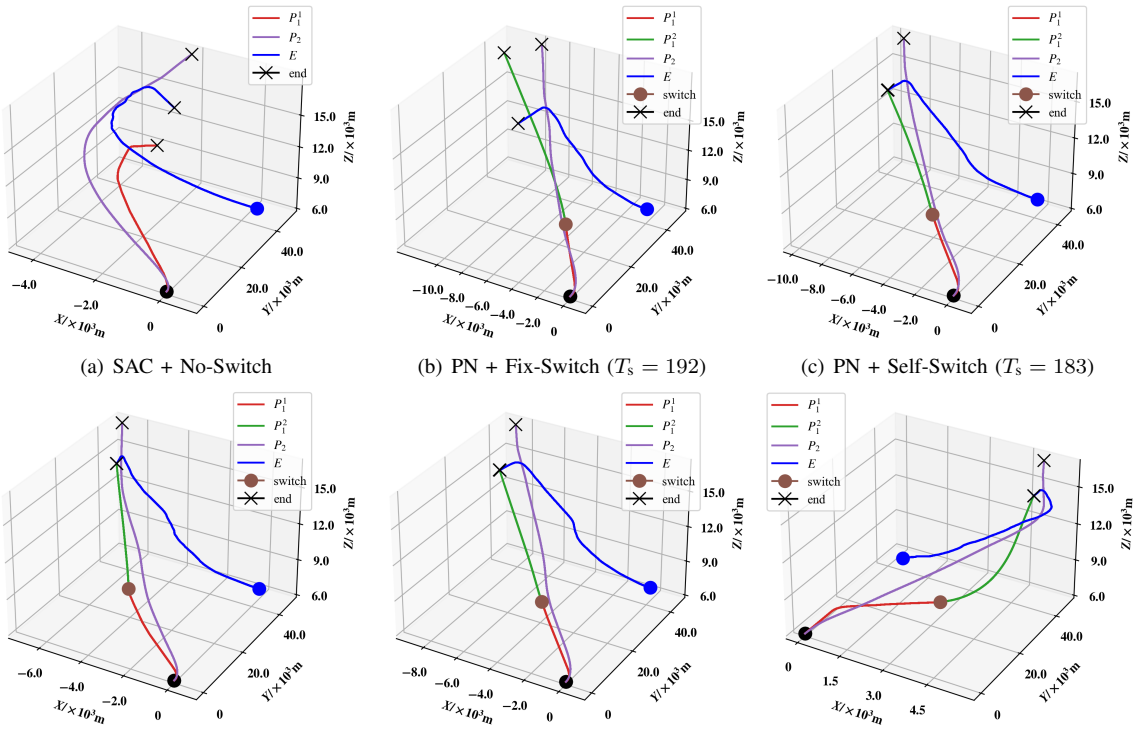
```
Initialize environment and replay buffer D;
Initialize parameter φ, α, ᾱ, β1, β2;
for episode = 1, M do
    Reset environment
    for t = 1, T_max do
        a_1^t ~ π_φ(a_1^t | s_c^t)
        if a_1^t contains action "Switch" then
            Switch to fast pursuer
            jump to subsequent simulation
        end
        s_c^{t+1} ~ p(s_c^{t+1} | s_c^t, a_1^t)
        D ← D ∪ (s_c^t, a_1^t, r(s_c^t, a_1^t), s_c^{t+1})
    end
    α ← α − λ_V ∇̂_α J_V(α)
    β_i ← β_i − λ_Q ∇̂_{β_i} J_Q(β_i)  for i ∈ {1, 2}
    φ ← φ − λ_π ∇̂_φ J_π(φ)
    ᾱ ← γα + (1 − γ)ᾱ
end
```

IV. EXPERIMENTS

A. Experiment Setup

The simulation experiments were run on an Apple M1 Pro CPU, 32 GB RAM, with a 10-core CPU and a 16-core GPU. The software environment used Python language with Torch software package.

In the training process, at the beginning of each episode, the pursuers are initialized to the same fixed position, while the evader appears at a random position within a certain range D_{init} outside the maximum capture distance D_c of the pursuers. At every interval Δt seconds, agents simultaneously execute a decision according to the current situation, which is called an execution step. Within the step, if any pursuer captures the evader, then pursuers are successful in this episode. Otherwise, if the evader is not captured within the given number of steps, then the pursuers have failed. After the end of one episode, the system starts the iteration of the next episode. The common parameter settings of the simulation are shown in Table II.



(a) SAC + No-Switch (b) PN + Fix-Switch ($T_s = 192$) (c) PN + Self-Switch ($T_s = 183$)

(d) SAC + Fix-Switch ($T_s = 188$) (e) SAC + Self-Switch (case 1) ($T_s = 171$) (f) SAC + Self-Switch (case 2) ($T_s = 166$)

Fig. 5. Comparison of simulation results for 5 different strategies. Where P_1^1 represents the trajectory of the flexible pursuer before the switch, and P_1^2 represents the trajectory of the flexible pursuer after switching to a fast pursuer.

TABLE II

PARAMETER SETTINGS

Parameter	Value
Capture Distance D_c (10^3 m)	30
Initial Distance D_{init} (10^3 m)	40
Initial position of pursuers (10^3 m)	$(0, 0, 6000)$
Initial relative azimuth of evader (rad)	$(-\pi, \pi)$
Initial relative elevation angle of evader (rad)	$(\pi/3, \pi/2)$
Initial angles of pursuers (ψ, θ, φ) (rad)	$(\pi/2, 0, 0)$
Initial angles of evader (ψ, θ, φ) (rad)	$([-\pi, \pi), 0, 0)$
Initial Speed of flexible agent (10^3m/s)	0.3
Initial Speed of fast agent (10^3m/s)	1.2
Speed range of flexible agent (10^3m/s)	$[0.15, 0.35]$
Maximum locking departure angle q_{max} (rad)	$\pi/3$
Maximum locking deviation angle p_{max} (rad)	$\pi/3$
Maximum pursuit time T_{max} (s)	250

B. Experiment Results

We design ablation experiments, combining different maneuvering and switch strategies, categorize them into five distinct strategy combinations, and conduct extensive simulation and validation experiments. There are two types of maneuvering strategies

- SAC, which is an intelligent maneuvering strategy.
- Proportional navigation (PN), which is a fixed strategy widely used in the aerial combat domain.

There are also two types of switch strategies

- Self-Switch, obtained by the structure in Section III-C.
- Fix-Switch, where the flexible pursuer immediately switches its dynamics when the distance between the fast pursuer and the evader decreases to D_f.

Additionally, we include an experimental group that uses the SAC framework to provide the maneuvering strategy that does not switch, to demonstrate the importance of the switch point in this game. We employ a strategy provided by an expert system for the flexible evader, which selects actions based on the positions, speeds, and altitudes of the other two pursuers, and guides the evader towards the most favorable position. Finally, we employ the PN strategy for all fast pursuers. We set the initial states of the 5 experimental groups to be the same, and the specific pursuit-evasion trajectories are shown in Fig. 5.

1) Drawback of No-Switch Strategy: Firstly, Fig. 5(a) mainly shows the No-Switch strategy. Although the flexible pursuer can track the fast pursuer under the guidance of the SAC framework, it fails to reach the evader within the finite time T_{max}. In this case, regardless of the maneuvering strategy adopted by the flexible pursuer, it cannot provide any assistance to the pursuer team. Consequently, the original two-on-one pursuit-evasion scenario reduces to a one-on-one engagement between the fast pursuer and the flexible evader. Hence, it becomes crucial and imperative for the flexible pursuer to switch dynamics during the game.

2) Comparison of Switch Strategies: We further delve into the advantages of Self-Switch over Fix-Switch. Comparing Fig. 5(d) and Fig. 5(e), we observe the trajectory P_1^1 and switch time T_s of the flexible pursuer before the switch point. With the Fix-Switch strategy, T_s appears excessively long, necessitating pull-up maneuvers post-switch to pursue the evader. By contrast, by employing Self-Switch, the flexible pursuer executes the switch at the opportune moment, result-

ing in trajectories P_1^1 and P_1^2 with minimal curvature. This indicates that the flexible pursuer avoids unnecessary detours. Similarly, Fig. 5(b) and Fig. 5(c) illustrate the enhanced capture rate attributable to the Self-Switch strategy.

3) Comparison of Maneuvering Strategies: The SAC algorithm significantly aids in the maneuvering decisions of the flexible pursuer. Comparing the trajectories in Fig. 5(b) to Fig. 5(e), with the PN method, the trajectory of the flexible pursuer P_1^1 overlaps with that of the fast pursuer P_2 before the switch point. By contrast, with the SAC algorithm, the flexible pursuer distances itself from the fast pursuer before switching, assuming different positions and adjusting its angle. This enables better coordination with the fast pursuer to enclose the evader. Additionally, we present simulation results for the *SAC + Self-Switch* algorithm under different initial conditions to demonstrate the algorithm's robustness, as depicted in Fig. 5(f).

4) Comparison of Capture Rate: Through extensive simulation experiments, we have statistically analyzed the capture rates of various algorithm combinations, as documented in Table III. In this comparison, *SAC + Self-Switch* exhibits remarkable performance in most cases. On the other hand, our algorithm encounters challenges under specific initial conditions, such as when there are significant height differences between pursuers and the evader or when the initial velocity directions are tail-chasing. These factors make it challenging for pursuers to achieve capture, influenced by objective conditions such as D_{init} and T_{max}.

TABLE III

CAPTURE RATE OF DIFFERENT COMBINATIONS OF STRATEGIES

Strategies	Capture Rate (10^5 Times Simulation)
SAC + No-Switch	20.76%
PN + Fix-Switch	56.89%
PN + Self-Switch	63.74%
SAC + Fix-Switch	81.02%
SAC + Self-Switch	**88.18%**

V. CONCLUSIONS

In this paper, we adopt the SAC framework from DRL to provide a hybrid strategy including maneuvering and switch strategies for a flexible pursuer with hybrid dynamics. By concatenating the switch factor with the state vector and further extending the original action space with switch actions, our algorithm enables the flexible pursuer to effectively seek advantageous positions in the early stages and switch its dynamics at an opportune moment, thereby realizing a pincer maneuver on the evader in collaboration with a fast pursuer. Through extensive simulations and comparisons in the constructed 3D space, our algorithm can significantly improve the capture rate compared to traditional methods. Future research may include introducing more agents, discussing the optimality of the algorithm and its proofs, and validating the effectiveness of the algorithm in physical experiments.

REFERENCES

[1] R. Isaacs, "Differential games: A mathematical theory with applications to warfare and pursuit, control and optimization," *Wiley*, 1965.

[2] M. M. Flood, "The hide and seek game of Von Neumann," *Management Science*, vol. 18, no. 5, pp. 107–109, 1972.

[3] L. Meier, "A new technique for solving pursuit-evasion differential games," *IEEE Transactions on Automatic Control*, vol. 14, no. 4, pp. 352–359, 1969.

[4] J. Shinar and S. Gutman, "Recent advances in optimal pursuit and evasion," in *IEEE Conference on Decision and Control*, 1979, pp. 960–965.

[5] M. Pachter and T. Milch, "The 'Homicidal Chauffeur' model in naval pursuit-evasion," in *Guidance, Navigation and Control Conference*, 1987, pp. 347–353.

[6] J. Shinar, V. Y. Glizer, and V. Turetsky, "Pursuit-evasion game of kind between hybrid players," *Advances in Dynamic and Evolutionary Games: Theory, Applications, and Numerical Methods*, pp. 187–208, 2016.

[7] R. Yan, R. Deng, H. Lai, W. Zhang, Z. Shi, and Y. Zhong, "Homicidal Chauffeur reach-avoid games via guaranteed winning strategies," *IEEE Transactions on Automatic Control*, vol. 69, no. 4, pp. 2367–2382, 2024.

[8] S. Yang, Z. Shan, Y. Ding, and G. Li, "Survey of research on deep reinforcement learning," *Computer Engineering*, vol. 47, no. 12, pp. 19–29, 2021.

[9] J. Shinar, "Solution techniques for realistic pursuit-evasion games," *Advances in control and dynamic systems*, vol. 17, pp. 63–124, 1981.

[10] N. Greenwood, "A differential game in three dimensions: The aerial dogfight scenario," *Dynamics and Control*, vol. 2, no. 2, pp. 161–200, 1992.

[11] F. Imado and T. Kuroda, "A method to solve missile-aircraft pursuit-evasion differential games," in *IFAC World Congress*, 2005, pp. 176–181.

[12] R. Yan, Z. Shi, and Y. Zhong, "Construction of the barrier for reach-avoid differential games in three-dimensional space with four equal-speed players," in *IEEE Conference on Decision and Control*, 2019, pp. 4067–4072.

[13] R. Yan, X. Duan, Z. Shi, Y. Zhong, and F. Bullo, "Matching-based capture strategies for 3D heterogeneous multiplayer reach-avoid differential games," *Automatica*, vol. 140, p. 110207, 2022.

[14] T. Yan, Y. Cai, and B. Xu, "Evasion guidance algorithms for air-breathing hypersonic vehicles in three-player pursuit-evasion games," *Chinese Journal of Aeronautics*, vol. 33, no. 12, pp. 3423–3436, 2020.

[15] J. Shinar, V. Y. Glizer, and V. Turetsky, "A pursuit-evasion game with hybrid pursuer dynamics," *European Journal of Control*, vol. 15, no. 6, pp. 665–684, 2009.

[16] V. Y. Glizer and V. Turetsky, "Increasing pursuer capturability by using hybrid dynamics," *International Journal of Applied Mathematics and Computer Science*, vol. 25, no. 1, pp. 77–92, 2015.

[17] Y. Cao, Y. X. Kou, Z. W. Li, A. Xu, *et al.*, "Autonomous maneuver decision of UCAV air combat based on double deep Q network algorithm and stochastic game theory," *International Journal of Aerospace Engineering*, vol. 2023, pp. 1–20, 2023.

[18] S. Li, Y. Jia, F. Yang, Q. Qin, H. Gao, and Y. Zhou, "Collaborative decision-making method for multi-UAV based on multiagent reinforcement learning," *IEEE Access*, vol. 10, pp. 91 385–91 396, 2022.

[19] X. Liu, Y. Yin, Y. Su, and R. Ming, "A multi-ucav cooperative decision-making method based on an mappo algorithm for beyond-visual-range air combat," *Aerospace*, vol. 9, no. 10, pp. 563–581, 2022.

[20] Z. Fan, Y. Xu, Y. Kang, and D. Luo, "Air combat maneuver decision method based on A3C deep reinforcement learning," *Machines*, vol. 10, no. 11, pp. 1033–1051, 2022.

[21] R. Zhang, Q. Zong, X. Zhang, L. Dou, and B. Tian, "Game of drones: Multi-UAV pursuit-evasion game with online motion planning by deep reinforcement learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 10, pp. 7900–7909, 2022.

[22] F. Austin, G. Carbone, M. Falco, H. Hinz, and M. Lewis, "Automated maneuvering decisions for air-to-air combat," in *Guidance, navigation and control conference*, 1987, pp. 659–669.

[23] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International conference on machine learning*, 2018, pp. 1861–1870.