IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 4, 2019, accepted July 12, 2019, date of publication July 23, 2019, date of current version August 12, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2930640

Adaptive Regret Minimization for Learning Complex Team-Based Tactics

DUONG D. NGUYEN[1], ARVIND RAJAGOPALAN[2], JIJOONG KIM[2], AND CHENG-CHEW LIM[1], (Senior Member, IEEE)

[1]School of Electrical and Electronic Engineering, The University of Adelaide, Adelaide, SA 5005, Australia
[2]Weapons and Combat Systems Division, Defense Science and Technology Group, Edinburgh, SA 5111, Australia

Corresponding author: Cheng-Chew Lim (cheng.lim@adelaide.edu.au)

ABSTRACT This paper presents an approach and analysis for performing decentralized cooperative control of a team of decoys to achieve the Honeypot Ambush tactic. In this tactic, the threats are successfully lured into a designated region where they can be easily defeated. The decoys learn to cooperate by incorporating a game-theory-based online-learning method, known as regret minimization, to maximize the team's global reward. The decoy agents are assumed to have physical limitations and to be subject to certain stringent range constraints required for deceiving the networked threats. By employing an efficient coordination mechanism, the agents learn to be less greedy and allow weaker agents to catch up on their rewards to improve team performance. Such a coordination solution corresponds to achieving convergence to coarse correlated equilibrium. The numerical results verify the effectiveness of the proposed solution to achieve a global satisfaction outcome and to adapt to a wide spectrum of scenarios.

INDEX TERMS Multi-agent system, cooperative control, online learning, regret minimization.

I. INTRODUCTION

Multi-agent reinforcement learning must factor in the collective behavior of participating agents. When teaming conditions are to be satisfied, agent coordination is essential. Without this coordination mechanism, each agent often aims to maximize its individual reward without due concern for the obtainable rewards of the other agents and the joint team reward. Therefore, cooperation among all the agents is crucial in a multi-agent setting with teaming requirements.

Despite previous efforts to develop models and mechanisms to enhance cooperative behaviors in MAS [1], [2], many open questions about this research remain. In general, the convergence of reinforcement learning assumes a stable training environment where the outcome for an action is consistent. However, under a standard multi-agent reinforcement learning (RL) approach, an agent treats other agents as a part of the environment. Consequently, the environment under this treatment is non-stationary as the feedback an agent receives from this shared environment is more likely affected by the decisions of the other agents. Additionally, while an agent may decide its action based on some prior information and assumptions about the other agents, such information will most likely change over time as the others adapt their behaviors while learning. In turn, this will affect how each agent needs to respond through re-adapting. The speed of re-adapting appropriately is a critical consideration. Therefore, there is still an open research problem in terms of designing suitable learning goals, supporting scalability and accommodating the non-stationary conditions.

This study focuses on the design of an effective decentralized coordination mechanism that enables multiple agents to learn collaboratively and guarantees task completion. In particular, a decentralized multi-agent algorithm based on regret minimization [3] is proposed to learn a cooperative joint-policy. Unlike conventional RL algorithms (such as Q-learning) that must be trained offline, the regret minimization method is able to learn online. An online learning algorithm can continue to learn over time, and adapt rationally. This feature offers flexible to accommodate scenarios which may be missing in an offline training.

In addition, an enhancement for updating the regret computation is provided through a forgetting factor so that the learning agent is able to adapt quickly to its most recent observations when the shared environment changes rapidly. The proposed method preserves the convergence guarantee to an equilibrium solution for all the agents, despite modifying the standard regret minimization technique. Theoretical proof

The associate editor coordinating the review of this manuscript and approving it for publication was Yichuan Jiang.

for the convergence guarantee is provided. The regret minimization incorporated as part of the multi-agent RL solution to enable the online learners to make non-myopic optimal decisions. Such decision making avoids greedy decisions that may not be optimal when factoring in near future implications. Note that the result presented in this paper is applicable not only to multi-UAV systems, but also to MAS of different unmanned vehicles, such as mobile robots and autonomous underwater vehicles.

The contribution of this paper is summarized as follows:

1) In contrast to most of game-theory based approaches, this paper proposes a multi-agent learning strategy for ''cooperation'' to handle a type of team tasks in which the mission goal will be a failure if any of the agents do not succeed to perform their tasks. Focus is shifted to ensure that the rewards are evenly distributed to all the member agents (satisfaction-based rather than optimization-based).
2) This paper develops an easy-to-construct ''model free'' design that incorporates a simple joint reward function which exhibits robust performance, especially against highly dynamic changes in the learning environment. Using this model, a cooperative multi-agent algorithm using regret matching strategy is proposed to achieve both individual and collective goals. The paper demonstrates that the regret minimization based technique is applicable not only to non-cooperative game settings (which most existing regret-based algorithms focus on) but also to large-scale coordination games.
3) This paper proposes a novel way of adapting the regret matching update rule by using a recency bias and provides analytical proof for a guarantee of convergence and adaptation to a rapidly changing environment. This is a new contribution to the body of works since the majority of previous works on multi-agent learning are restricted to validate their solutions via simulation only.

II. RELATED WORK

This section reviews the major differences between the proposed approach and the relevant works.

A. MULTI-AGENT REINFORCEMENT LEARNING

Multi-agent reinforcement learning (MARL) is a relatively recent and expanding research endeavor. It encompasses the contributions made by the reinforcement learning and game theory communities. Single-agent RL has been solved using algorithms that display convergence and consistency properties. However, learning in a decentralized multi-agent setting is non-trivial since the agents update their states asynchronously and need to consider the global performance of all other agents when making an individual decision. The core issue for multi-agent learning with teaming requirements is how to learn an optimal cooperative policy by balancing the trade-off between individual and group rewards.

There are multiple decentralized coordination studies using Q-learning, policy gradients and deep learning to formulate a joint policy by considering the combined observations from the multiple agents [4]–[8]. The algorithms produced have been applied to devise cooperative or competitive or mixed behaviors. In [9], the authors investigate a deep learning approach, in which agents learn the policies by continuous communication among the parties involved to collaborate their behaviors in order to achieve a joint objective. Achieving this objective requires all agents to synchronize their observation histories with others. In [10], a coordination mechanism for solving single-agent tasks using multiple agents was proposed. The reward function of each agent is explicitly designed to take into account the actions of the other agents. This framework allows each agent to flexibly control its coordination action depending on the context. For example, an agent may act independently in situations with high environment reward while engaging in an appropriate relationship with others to satisfy reward expectations for situations of low environment reward.

One common feature of those earlier studies on decentralized coordination is that the agents balance their behaviors between the policy that is being learned and the communication with the other agents. It appears that these studies rely on simulations to evaluate the performance of their proposed solutions, while the associated theoretical analyses have not been fully addressed. In addition, the training is offline.

B. REGRET-BASED LEARNING FRAMEWORK FOR MAS

The design of RL-based algorithms using regret minimization technique [11]–[20] with global convergence guarantees has attracted significant attention in recent years. The idea behind regret-based algorithm is to adapt the decision-making policy according to changes in the environment. In this online-learning approach, an agent decides its action strategy (policy) probabilistically and observes the new regret to evaluate the policy. As time progresses, the agent enhances the repetition for making good decisions while attempting to minimize the number of times selecting unsuitable or wrong decisions which result in low instantaneous rewards and potentially lead to hazards in real-world situations. Through its online learning provisions, the agent can adapt its decisions as the environment changes.

Most of existing regret-based learning algorithms for MAS are based on non-cooperative approaches [16]–[20] and designed to reach a no-regret equilibrium point for all agents [12]–[15]. In such a situation, the average reward is no less than the corresponding amount that would have been achieved if other fixed strategies were chosen in all decisions. However, having convergence towards a no-regret solution is not necessarily a desirable outcome for a MAS in general. There is no guarantee that the MAS converges to a solution that meets the desired global objective.

The solution proposed in this paper follows the regret-based principles. Unlike most of the existing non-cooperative approaches, this solution focuses on the learning of a cooperative strategy by developing a simple joint reward model to improve coordination among autonomous

agents. It is shown that under certain assumptions about the learning environment, the agents' joint action converges to the set of no-regret solutions in a cooperative fashion. Both theoretical and experimental studies are provided to support the solution. Simulation results verify the effectiveness of the proposed learning algorithm in deriving complex and adaptive team-based tactics in a set of real world scenarios.

C. COORDINATED CONTROL OF MAS

Multiple spatially-distributed agents need coordination or cooperation when an individual agent has insufficient sensing, knowledge, or capability to execute a decision. Some application examples are in cooperative search and rescue, bushfire fighting, defense against salvo of threats, and traffic control. Recent developments in autonomous vehicles have seen the coordinated control of MAS being applied to unmanned aerial vehicles, robots and aircraft [21]. Current techniques for the coordinated control of MAS include dynamic programming [22], fuzzy-based model [23], model predictive control [24]–[26], machine learning [27]–[29].

This paper focuses on the application of coordinated control techniques for multi-UAV systems. Cooperative multi-UAV control problems are solved by centralized and decentralized approaches. The centralized approach (i.e., [30]) uses a controller to solve the global optimization problem. Here, each UAV uploads its state parameters and environmental information to the controller. The controller computes a solution in terms of the requirement for each UAV's flight path and speed. This information is then communicated to the respective UAVs and executed in order to achieve the collective goal. The advantage of this approach is that it can find a globally optimal solution for cooperative control problems. However, the solution can be communication bandwidth intensive. Furthermore, the bandwidth requirements can scale up rapidly with additional UAVs.

In decentralized approaches, such as [31], the large optimization problem is decomposed into multiple smaller problems which are easier to solve. Each UAV can obtain environmental information from its neighboring UAVs but calculates its own flight path independently. On the whole, optimal flight path of each UAV can ensure that the whole system will achieve the best result. The advantage of this approach is that it spreads out the computation which can reduce local computation per UAV as well as the communication bandwidth requirement. Furthermore, robustness for the entire UAV system against single point of failure is greatly increased. The disadvantage relates to achieving coordination among different UAVs which are no longer considered altogether during the solving of the path-planning.

In this work, a decentralized approach is used to address the problem of controlling multiple UAVs for a defense application, in which a high-value asset (HVA) is escorted by a group of UAV decoys as it travels to its destination in a hostile environment. The task of the UAV team is to protect the HVA from simultaneous attack by a group of high-speed networked aerial threats. For the rest of the paper, we refer to the HVA as target and the threats that attempt to hit the target as attackers. Several works address a similar problem [20], [32]. In these works, the multiple UAVs deployed countermeasures to deflect enemy attackers, but did not cooperate. The work in [33] accounted the possibility of coordination between UAVs in order to maximize their own safety and the safety of the other members of the group but did not address a joint task among the participant UAVs, which is a focus of this work. Also note that while this paper seeks to address a particular application of a multi-UAV system, the proposed algorithm is general and can be applied to a range of large-scale decentralized MASs.

Remark 1: The system considered here is a decentralized multi-agent control system. It does not rely on a master agent to impose individual decisions. The control actions for the agents are implemented in a distributed manner since each agent updates its own control input based on information from its neighboring agents in order to coordinate [1]. There is no requirement for tightly coupled actions between agents to fulfill their joint mission. Note that under game theory concept, the proposed game-based algorithm is classified as a partially distributed learning approach due to the fact that each player (agent) must use information about the other players (i.e., chosen actions) to update its decision-making strategy. It is considered partially distributed only because in a fully distributed learning approach, players are able to make decisions based solely on their local observations, without extra knowledge about the other players or the complete system information [34].

III. PROBLEM FORMULATION

A. SYSTEM DYNAMICS AND ASSUMPTIONS

For a concept demonstration, it is assumed that the target is moving due North at a constant speed v. Each UAV decoy (a RL agent) processes the attacker radar signal and radiates back to the attacker a target-like radio frequency signature. This will make the attacker see a false target (FT) placed some distance behind the decoy. Since the networked attackers are homing onto a single target, the decoys must coordinate their motion to present a single consistent FT to all of the attackers. To do so, the real target and the FT need to be in the same range gate (similar distances away from the attacker). This is important in terms of not triggering the threats' electronic countermeasure (ECM) logic. Our work focuses on proposing an effective decentralized cooperative solution for controlling the UAV decoys to deceive and steer all the networked threats into a honeypot trap location. Figure 1 provides a 2D graphical representation of the deception for two attackers using two UAV decoys.

Let $\mathbf{p}_t^i$ denote the location of the FT generated by the decoy agent i at time step t. Then, its dynamic model is given by

$$\mathbf{p}_{t+1}^i = \mathbf{p}_t^i + \Delta t \times \|\mathbf{v}_t^i(\theta)\| \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \end{bmatrix},$$

where Δt is the time interval between discrete position update, θ is the heading angle which is also used as the

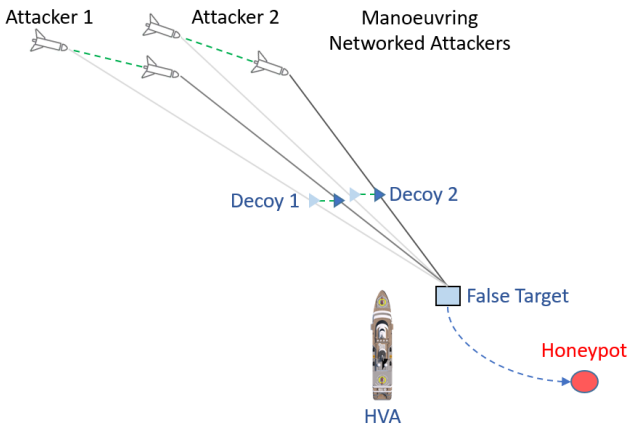


FIGURE 1. Deception for two attackers using two UAV decoys.

decision variable for the RL model, and $\mathbf{v}_t^i$ is the velocity vector which has a constant magnitude (i.e., $\|\mathbf{v}_t^i(\theta)\| = \nu$ for all $\theta \in [0, 2\pi]$) but a changing direction.[1] The heading angle θ is decided by the agent i with the aim to obtain the best next waypoint for its respective FT.

An attacker is assumed to have locked onto the FT which is initially co-located with the HVA. The attacker is also assumed to be fully observable by the associated RL agent. The dynamics of an attacker $m \in \mathcal{M}$ (the attacker set) is modeled by a constant-velocity model with time-varying heading angle

$$\mathbf{p}_{t+1}^m = \mathbf{p}_t^m + \Delta t \times \mathbf{v}_t^m(\mathbf{p}_{t+1}^i),$$

where the attacker's velocity vector $\mathbf{v}_t^m$ is defined using the proportional navigation guidance law [35].

To enable collaboration across the agents, it is assumed that agents share their own states and the states of their respective attackers in real-time. Thus, at a given time t, the following global state information is available to all of the RL agents:

$$\left[\mathbf{p}_t^s, \mathbf{p}_t^h, \mathbf{p}_t^i, \mathbf{p}_t^m, \mathbf{v}_t^m\right] \text{ for all } i \in \mathcal{N} \text{ and } m \in \mathcal{M},$$

where $\mathbf{p}_t^s$ and $\mathbf{p}_t^h$ are the positions of the target and the honeypot at time t, respectively. All measurements of the FT and the attacker dynamics are assumed to be noise-free.[2] Based on this available information, each agent makes its decision independently without depending on the decisions made by the others. An action decided by an agent is the heading angle θ of its generated FT. The set of available actions controlled by an agent varies over time depending on its previous action, due to the following two constraints:

(i) The maximum difference in turning angle must not exceed $\frac{\pi}{2}$ rad, which means the next action θ' given the current action θ must satisfy: $\theta - \frac{\pi}{2} \le \theta' \le \theta + \frac{\pi}{2}$.

(ii) Each agent must maintain its generated FT within a sufficient range when the target is contained inside the field-of-view (FOV)[3] range of its assigned attackers. A chosen action θ under this situation has to satisfy the constraint that: $D(\theta) = \left|\|\mathbf{p}_{t+1}^m - \mathbf{p}_{t+1}^i\| - \|\mathbf{p}_{t+1}^m - \mathbf{p}_{t+1}^s\|\right| \le \lambda$, where $\lambda > 0$ is a certain threshold specified by the UAV team.

B. MULTI-AGENT GAME-THEORETIC FRAMEWORK

Consider a group of $N \ge 2$ agents, denoted by the finite set $\mathcal{N} = \{1, \ldots, N\}$, that performs a team mission in a common environment. Each agent $i \in \mathcal{N}$ has a finite set of states $\mathcal{S}^i$, a finite action set $\mathcal{A}^i$, and a local reward function $U^i : \mathcal{A} \times \mathcal{S} \to \mathbb{R}$, where $\mathcal{S} = \Pi_{\ell \in \mathcal{N}} \mathcal{S}_\ell$ is the global state space shared by all the agents, and $\mathcal{A} = \Pi_{\ell \in \mathcal{N}} \mathcal{A}_\ell$ is their joint action set. More precisely, the reward function of an agent i can be expressed in the form $U^i(a_t^i, a_t^{-i}, s_t)$. Besides its action a_t^i, the reward of agent i depends on the global state of the world $s_t = (s_t^1, \ldots, s_t^N)$ and the action $a_t^{-i} = \{a_t^\ell\}_{\ell \in \mathcal{N} \setminus i}$ of all other agents in the team. The multi-agent game then can be denoted by $\mathcal{G} = (\mathcal{N}, (\mathcal{S}^i)_{i \in \mathcal{N}}, (\mathcal{A}^i)_{i \in \mathcal{N}}, (U^i)_{i \in \mathcal{N}})$.

Now, suppose the game $\mathcal{G}$ is being repeatedly played over time. At each time step t, all the agents observe the global state $s_t \in \mathcal{S}$ and perform a joint action $a_t = (a_t^i, a_t^{-i}) \in \mathcal{A}$, where $a_t^i \in \mathcal{A}^i$ denotes the pure action of an agent i and $a_t^{-i} \in \mathcal{A}^{-i} = \Pi_{\ell \ne i} \mathcal{A}_\ell$ denotes the action combination of the other agents. It is also assumed that agents exchange their current states and chosen actions globally but only observe their rewards locally. Also, the agents know their local reward functions. Therefore, knowing the action profile of other agents and the current global state, each agent is able to compute it expected reward. The reward observed by an agent i at time t can be denoted by $U_t^i(a_t^i, a_t^{-i}) = U^i(a_t^i, a_t^{-i}, s_t)$.

Denote by $\Delta(\mathcal{A})$ the set of all probability mass functions (pmf) on $\mathcal{A}$, and similarly, for each $i \in \mathcal{N}$, $\Delta(\mathcal{A}^i)$ as the set of pmf over $\mathcal{A}^i$, and $\Delta(\mathcal{A}^{-i})$ as the set of pmf over $\mathcal{A}^{-i}$. Let $\pi^i \in \Delta(\mathcal{A}^i)$ (i.e., a probability distribution over the set $\mathcal{A}^i$) denote a strategy for agent i with $\pi^i(a^i)$ equal to the probability that i chooses the pure action $a^i \in \mathcal{A}^i$. A strategy π^i is called a pure strategy if $\pi^i(a^i) = 1$ for some $a^i \in \mathcal{A}^i$. A strategy is called a mixed strategy if it is not a pure strategy. Let $\pi^{-i} \in \Delta(\mathcal{A}^{-i})$ denote the joint strategy of all the other agents (except i). Similarly, let $\pi \in \Delta(\mathcal{A})$ denote the joint mixed strategy of all agents, with $\pi(a)$ is a joint probability on the set $\mathcal{A}$, where $a = (a^i, a^{-i})$ is a joint action profile of all the agents. In this context, under randomized actions with overall probability $\pi = (\pi^i, \pi^{-i}) \in \Delta(\mathcal{A})$, the expected reward obtained by an agent i is defined by extending the domain of definition of U^i to $\Delta(\mathcal{A})$ according to[4]

$$U^i(\pi) = \sum_{a \in \mathcal{A}} \pi(a)\, U^i(a).$$

In this work, the focus is on investigating the use of a probabilistic game-theoretic concept, known as coarse correlated

[1] Note that the FT is required to move with the same speed as the target to make it indistinguishable from the target. Thus, while the speed of the FT stays constant, its direction changes when selecting a different action.

[2] Note that a probabilistic model which takes into account noisy observations is more realistic but is outside the scope of this work.

[3] FOV is the angular extent observed by the attacker, which allows it to find an object with increased location uncertainty.

[4] Notice that $U^i(\cdot)$ is a linear function.

equilibrium (CCE) [36]–[38]. A CCE point is a mixed joint action profile (probability distribution over the joint action), where no agents can obtain any expected gain by deviating from said strategy. Unlike Nash equilibrium (NE) which might not exist in certain games, CCE not only always exist but also are easy to compute in arbitrary finite games with two or more players [36]. The main advantage of reaching a CCE solution is that by enabling the players to coordinate their actions, CCE provide a balance between the non-cooperative solution (where all the players work independently but yield poor collective behavior) and the fully cooperative solution (which requires stringent coordination between players but can be highly efficient in team performance). Thus, CCE are expected to provide superior collective behavior for the MAS compared to a NE solution where only the non-cooperative aspect is considered [38].

Definition 1: A probability distribution $\pi^* \in \Delta(\mathcal{A})$ is said to be a CCE for the game $\mathcal{G}$ if for every agent $i \in \mathcal{N}$ and for every action $\theta \in \mathcal{A}^i$ of player i, it holds that [37]

$$\sum_{a \in \mathcal{A}} \pi^*(a) \left(U^i(\theta, a^{-i}) - U^i(a) \right) \leq 0. \tag{1}$$

The proposed algorithm for obtaining a CCE for the problem introduced in Section III-A is presented in Section IV.

C. DESIGN OF REWARD FUNCTION

The next step is to define a suitable reward function to obtain cooperative behaviors among the agents. In this problem, one objective for an agent is to keep the perpendicular distance from its assigned attacker to the true target (the miss-distance) as large as possible subject to its kinematic constraints. In addition, to behave as a unique single FT, it is also important to have all the FTs generated by the agents maintaining at the same locations over time and move to the honeypot simultaneously. To achieve those objectives together, the reward function at a time t is defined as follows:

$$U_t^i(a_t) = \min_{i \in \mathcal{N}} \left\{ d_t^i(a_t) \right\} - \omega_1 \times \max_{i,j \in \mathcal{N}} \left\{ r_t^{i,j}(a_t) \right\} - \omega_2 \times \ell_t^i(a_t). \tag{2}$$

In (2), $a_t = (a_t^i, a_t^{-i})$ is the joint action of all agents at time t. As a result of the joint action a_t, $d_t^i(a_t)$ is the miss-distance obtained by the agent i, $r_t^{i,j}(a_t)$ is the relative distance between the FTs generated by the agent i and an agent $j \in \mathcal{N}$, and $\ell_t^i(a_t)$ is the distance from the agent i's FT to the honeypot location. Here, $[\omega_1, \omega_2]$ are positive weight parameters. The geometry for determining d_t^i, $r_t^{i,j}$ and ℓ_t^i is illustrated graphically in Figure 2.

The reward function can be interpreted as an obtainable payoff for completing the joint task by each team player (agent) at the current point in time. This individual payoff measures the amount of penalty given as a weighted sum of: (1) minimum predicted miss distance assuming straight line trajectory of the threats, (2) measure of aggregation of the false targets, and (3) distance-to-go to the desired destination. The rationale here is that the relative distances between all FT pairs as well as the direct distance from the false target to the honeypot position must be included in the reward function to obtain the cooperative team behavior. Specifically, the major differences between this proposed cooperative approach and a non-cooperative approach using regret-based learning framework, i.e., [20], are as follows:

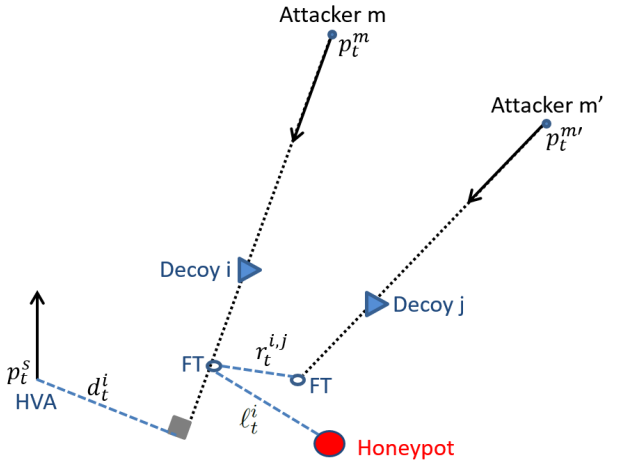


FIGURE 2. Geometry for determining the reward function.

(a) First, by using the minimum function of all the miss-distances instead of an individual measure of the miss-distance obtained by each agent, this reward provides a feedback on the effectiveness of a joint action taken by all agents rather than the selfish action of a single agent.

(b) Second, as included in the second term of (2), the proposed solution tries to bring together the FTs whenever possible by adding in a new penalty term as a linear function of maximum inter-FT separation. The weight ω_1 is used to influence the inter-FT separation.

(c) Third, each agent not only tries to reach a large miss-distance but also aims to move towards the honeypot such that it reduces its direct distance to the honeypot. This is obtained by discouraging the action that drags the FTs away from the honeypot. The weight ω_2 is used to penalize the unsuitable behaviors.

In this study, an agent learns to minimize the loss between prediction and actual reward obtained at each time step using a regret minimization based strategy rather than a reward maximization mechanism. The outcome is that the expected average reward (or long-term payoff) for a learning agent is guaranteed to be no worse than choosing the best fixed policy at any time while it is learning. Note that the polarity of the reward function values is irrelevant for the algorithm analyzed in this paper. It is also worthwhile to mention that the specific setting of the reward function presented in (2) is only one among possible designs that can factor in the impact of other agents' actions to encourage cooperation and to improve overall team performance. The solution proposed in this work is general and can flexibly support a wide range of team-reward functions.

IV. PROPOSED ALGORITHM AND ANALYSIS

A. REGRET MINIMIZATION BASED LEARNING

The proposed algorithm is based on the regret matching procedure in [3], in which agents choose their actions based on their "regrets" for not having selected other available options. In particular, at each iteration $t \geq 1$, an agent i calculates the average regret for not having played a particular action $\theta \in \mathcal{A}^i$ in all the previous time steps using the following recursive formula:

$$\bar{R}_t^i(\theta) = \frac{1}{t}\sum_{\tau=1}^{t} R_\tau^i(\theta) = \left(1 - \frac{1}{t}\right)\bar{R}_{t-1}^i(\theta) + \frac{1}{t}R_t^i(\theta). \quad (3)$$

In (3), $R_t^i(\theta) = U_t^i(\theta, a_t^{-i}) - U_t^i(a_t)$ is the immediate regret, in which $U_t^i(a_t)$ is the actual received reward and $U_t^i(\theta, a_t^{-i})$ is the potential reward that agent i could have obtained at time t if choosing the other action $\theta \neq a_t^i$ instead of its chosen action a_t^i assuming that all the actions of the other agents are unchanged. $\bar{R}_{t-1}^i(\theta)$ corresponds to the cumulative average regret experienced up until time $(t-1)$. In order to compute this regret, each agent is assumed to have access to a common dataset that records the global state space for all agents and their chosen actions.

In a multi-agent environment, the reward obtained by an agent for choosing a particular action will vary if the other agents deviate from the assumed strategies. As a result, past regrets may become irrelevant or outdated when making decisions at the current state. Thus, to enhance the learning process for the dynamic situation, a discount factor is used in the regret updating formula as follows:

$$\bar{R}_t^i(\theta) = (1-\rho)\bar{R}_{t-1}^i(\theta) + \rho\, R_t^i(\theta), \quad (4)$$

where $(1-\rho) \in (0, 1)$ is a discounted weight used to regulate the influence of outdated values of past regrets with respect to its immediate value. It follows that a chosen value of ρ close to 1 will make the agent "myopic" by only considering instantaneous regret, while a ρ factor approaching 0 will make the agent factors in more strongly on the past observed regrets rather than just the instantaneous regret. Similar approaches can be found in [12], [18]. The proposed algorithm to compute the CCE solution is summarized in Algorithm 1. Note that we use the notation $|x|^+ = \max\{x, 0\}$ for any $x \in \mathbb{R}$ and $|\mathcal{A}^i|$ to denote the cardinality of the action set $\mathcal{A}^i$.

The regret-based decision-making mechanism presented in Algorithm 1 belongs to a well-known class of no-regret algorithms which guarantee that the payoff of a learning agent in the long run is close to the maximum it could expect to achieve by consistently deviating from the algorithm's suggested action. No-regret was chosen in order to ensure that all agents are able to choose sequential actions such that mission objectives could be met. Failure of any one agent in performing its role amounts to a team-failure. All agents were required to progressively make better decisions over time which translated to minimizing their respective regrets.

Algorithm 1 No-Regret Multi-Agent Learning Algorithm

1: *Initialization:* Generate random $\pi_1^i(\theta)$ for all $\theta \in \mathcal{A}^i$.
2: **for** $t = 1, 2, \ldots$ **do**
3: *Action Selection:* Select action a_t^i according to π_t^i.
4: *Reward Observation:* Obtain $U_t^i(a_t)$ and compute $U_t^i(\theta, a_t^{-i})$ for all θ using equation (2).
5: *Regret Update:* Compute the regret vector $\bar{R}_t^i$ using equation (4) for all $\theta \neq a_t^i$.
6: *Dynamic Constraint Checking:* Limit the available action set $\mathcal{A}^i$ at the next time step by
$\begin{cases} \theta - \frac{\pi}{2} \leq \theta' \leq \theta + \frac{\pi}{2} \\ D(\theta) = \left| \|\mathbf{p}_{t+1}^m - \mathbf{p}_{t+1}^i\| - \|\mathbf{p}_{t+1}^m - \mathbf{p}_{t+1}^s\| \right| \leq \lambda \end{cases}$
7: *Policy Learning:* Update the action selection strategy $\pi_{t+1}^i(\theta) = \mathbb{P}\left(a_{t+1}^i = \theta | s_t\right)$ according to
if $\exists\, R_t^i(\theta) > 0$ **then** $\pi_{t+1}^i(\theta) = \dfrac{|\bar{R}_t^i(\theta)|^+}{\sum_{\theta'=1}^{|\mathcal{A}^i|} |\bar{R}_t^i(\theta')|^+}$
else $\pi_{t+1}^i(\theta) = \begin{cases} 1 & \text{if } \theta = \underset{\theta' \in \mathcal{A}^i}{\operatorname{argmin}}\, D(\theta') \\ 0 & \text{otherwise} \end{cases}$
end if
8: **end for**

B. MAIN RESULT

Algorithm 1 has similar convergent results as Theorem B of [3]. However, due to the modification made in (4) to update $\bar{R}_t^i(k)$ compared to its original form, it becomes necessary to revisit the convergence proof for this algorithm. Differential inclusion framework introduced in [39], [40] is used for analyzing the convergence properties of the proposed algorithm. Note that the convergence of the proposed algorithm does not rely on an explicit form of the reward function being used. The following theorem is the main result of the paper.

Theorem 1: If all agents apply Algorithm 1, the empirical distributions of the joint actions converges to the set of coarse correlated equilibrium (CCE), in which the regrets of all the agents vanish simultaneously and thus no agent has an incentive to deviate from the CCE solution.

Remark 2: Within the scope of this paper, we only deal with homogeneous multi-agent systems, which assumes that all agents apply the same learning rule.

Proof: We view the game from the point of view of an arbitrary agent i. For simplicity of notation, we drop the subscript i on $\bar{R}^i$, R^i, and U^i; and thus write $\bar{R}$, R, and U in the remainder of the paper. Define the Lyapunov function:

$$P(\bar{R}_t) = \frac{1}{2}\left(\text{dist}[\bar{R}_t, \mathbb{R}^-]\right)^2 = \frac{1}{2}\sum_{\theta}\left(|\bar{R}_t(\theta)|^+\right)^2. \quad (5)$$

In (5), $\text{dist}[\bar{R}_t, \mathbb{R}^-]$ is the distance from $\bar{R}_t$ (the time average vector of R) to the negative orthant set $\mathbb{R}^-$.

Taking the time-derivative of (5) yields

$$\frac{d}{dt}P(\bar{R}) = \sum_{\theta}|\bar{R}(\theta)|^+ \times \frac{d}{dt}\bar{R}(\theta). \quad (6)$$

First, we find $d\bar{R}(\theta)/dt$ by rewriting $\bar{R}(\theta)$ from (4) in the following form

$$\bar{R}_t(\theta) = \bar{R}_{t-1}(\theta) + \rho\big[R_t(\theta) - \bar{R}_{t-1}(\theta)\big]. \tag{7}$$

It can be seen that (7) has the form of a stochastic approximation algorithm, where $\rho > 0$ serves as a constant step size, and satisfies Theorem 17.1.1 of [41]. Therefore, its dynamics can be characterized by an ordinary differential equation and the system can be approximated by replacing $R_t(\theta)$ with its expected value. Thus, $\bar{R}_t(\theta)$ converges in distribution to the averaged system corresponding to (7)

$$\begin{aligned}\frac{d}{dt}\bar{R}(\theta) &= \mathbf{E}_\pi\left\{R(\theta) - \bar{R}(\theta)\right\}\\ &= \mathbf{E}_\pi\left\{\Big[U(\theta, a^{-i}) - U(a)\Big] - \bar{R}(\theta)\right\}\\ &= \sum_a \pi(a)\Big[U(\theta, a^{-i}) - U(a)\Big] - \bar{R}(\theta)\\ &= \sum_{a^{-i}} \pi^{-i}(a^{-i})U(\theta, a^{-i}) - \sum_a \pi(a)U(a) - \bar{R}(\theta)\\ &= \Big[U(\theta, \pi^{-i}) - U(\pi)\Big] - \bar{R}(\theta).\end{aligned}$$

Next, substituting $d\bar{R}(\theta)/dt$ into (6), we obtain

$$\frac{d}{dt}P(\bar{R}) = \sum_\theta |\bar{R}(\theta)|^+ \times \Big[U_i(\theta, \pi^{-i}) - U_i(\pi)\Big] - \sum_\theta |\bar{R}(\theta)|^+ \times \bar{R}(\theta). \tag{8}$$

Also, recall that the action selection strategy of agent i is defined based on its average non-negative regret function

$$\pi^i(\theta) = \frac{|\bar{R}(\theta)|^+}{\sum_{\theta'=1}^{|\mathcal{A}^i|} |\bar{R}(\theta')|^+}.$$

Thus, substitute $|\bar{R}(\theta)|^+ = \pi^i(\theta)\ \sum_{\theta'=1}^{|\mathcal{A}^i|} |\bar{R}(\theta')|^+$ into the first term on the right hand side of (8), we obtain

$$\begin{aligned}&\sum_\theta |\bar{R}(\theta)|^+ \times \Big[U(\theta, \pi^{-i}) - U(\pi)\Big]\\ &\quad= \sum_{\theta'=1}^{|\mathcal{A}^i|} |\bar{R}(\theta')|^+ \ \sum_\theta \pi^i(\theta) \times \Big[U(\theta, \pi^{-i}) - U(\pi)\Big].\end{aligned}$$

Since $\sum_\theta \pi^i(\theta) \times \big[U(\theta, \pi^{-i}) - U(\pi)\big] = 0$ by the linearity of U (as a function of probability distributions), then the first term in (8) is equal to zero.

Now consider the last term on the right hand side of (8)

$$\sum_\theta |\bar{R}(\theta)|^+ \times \bar{R}(\theta) = \sum_\theta \left(|\bar{R}(\theta)|^+\right)^2 = 2P(\bar{R}) \tag{9}$$

by substituting $\sum_\theta \left(|\bar{R}(\theta)|^+\right)^2 = 2P(\bar{R})$ from (5).

Therefore, combining (8) and (9), we obtain

$$\frac{d}{dt}P(\bar{R}) = -2P(\bar{R}).$$

Consequently,

$$P\big(\bar{R}(t)\big) = P(\bar{R}(0))\ \exp(-2t).$$

This implies that $P\big(\bar{R}(t)\big)$ approaches zero at an exponential rate. Thus,

$$\lim_{t\to\infty} \operatorname{dist}\big[\bar{R}(t), \mathbb{R}^-\big] = 0.$$

This proves that the approachability of the regrets of the agent i to the negative orthant (i.e., all the regrets approach zero).

We now prove that the empirical distribution of the joint actions converges to the CCE set if all agents use the same policy. Let $\bar{\pi}_t$ denote the empirical frequency of joint action by all agents, which can be defined using the stochastic approximation recursion as follows

$$\begin{aligned}&\bar{\pi}_t\big(a_t = a\big)\\ &\quad= \bar{\pi}_{t-1}\big(a_{t-1} = a\big) + \rho\big[\mathbb{1}\{a_t = a\} - \bar{\pi}_{t-1}(a_{t-1} = a)\big]\\ &\quad= \rho\ \sum_{\tau\le t}(1-\rho)^{t-\tau}\ \mathbb{1}\{a_\tau = a\},\end{aligned} \tag{10}$$

where $\mathbb{1}\{\cdot\}$ denotes the indicator function. The main result is immediate from the definition of "regret" in (4) as follows

$$\begin{aligned}\bar{R}_t(\theta) &= \bar{R}_{t-1}(\theta) + \rho\left(R_t(\theta) - \bar{R}_{t-1}(\theta)\right)\\ &= \rho\ \sum_{\tau\le t}(1-\rho)^{t-\tau} R_\tau(\theta)\\ &= \rho\ \sum_{\tau\le t}(1-\rho)^{t-\tau}\left(U(\theta, a_\tau^{-i}) - U(a_\tau)\right)\\ &= \sum_a \rho\ \sum_{\tau\le t}(1-\rho)^{t-\tau}\,\mathbb{1}\{a_\tau = a\}\left(U(\theta, a^{-i}) - U(a)\right)\\ &= \sum_a \bar{\pi}_t\big(a_t = a\big)\left(U(\theta, a^{-i}) - U(a)\right).\end{aligned}$$

In the last line, we substituted $\bar{\pi}_t(a_t = a)$ from (10). Note that $\bar{\pi}_t$ denotes the time average empirical distribution of the joint action of all agents and $\bar{\pi}_t(a_t = a)$ is the pmf defined on $\Delta(\mathcal{A})$ with all mass at a specific joint action $a \in \mathcal{A}$. On any convergent subsequence $\lim_{t\to\infty} \bar{\pi}_t \to \pi^*$, then

$$\lim_{t\to\infty} \bar{R}_t(\theta) = \sum_a \pi^*(a)\left(U(\theta, a^{-i}) - U(a)\right) \le 0.$$

Finally, comparing with the definition of the CCE as defined in (1), the desired result follows. This completes the proof. ■

V. SIMULATION RESULTS

In the simulation setup, the HVA traveling due North is protected by a group of four UAV decoys. These decoys are used to seduce the range-gate networked attackers into a honeypot zone. The honeypot is predefined as an area of a circle with a diameter of 50 m, located 300 m diagonally behind the moving HVA and moving at half the speed of the HVA. The initial distance between the attackers and the HVA is set at 18 km. The attackers, HVA and FT are traveling at 306.27 m/s, 10.28 m/s, and 10.28 m/s, respectively. The range threshold $D(\theta)$ to satisfy the range gate constraint is chosen at $\lambda = 50$ m. The weight coefficients in the reward function are set at $w_1 = w_2 = 20$. The results of the simulation are presented in 2D environment.

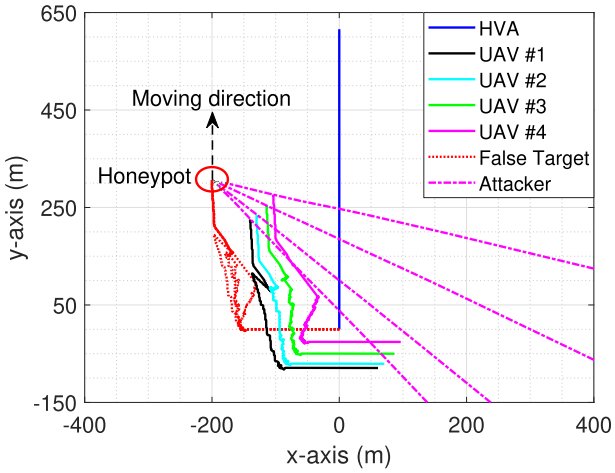


FIGURE 3. UAV team defends four attackers diagonally from behind.

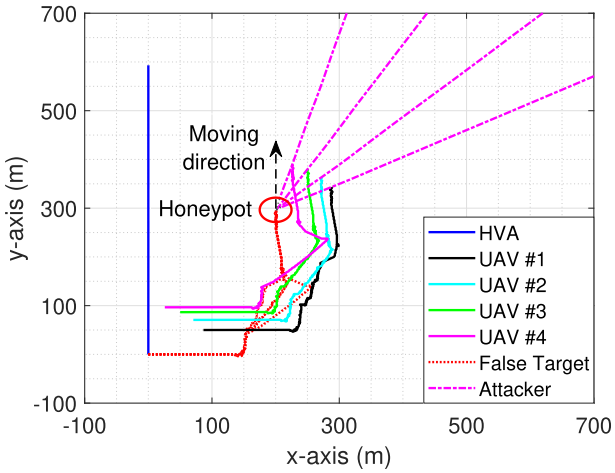


FIGURE 4. UAV team defends four attackers diagonally from front.

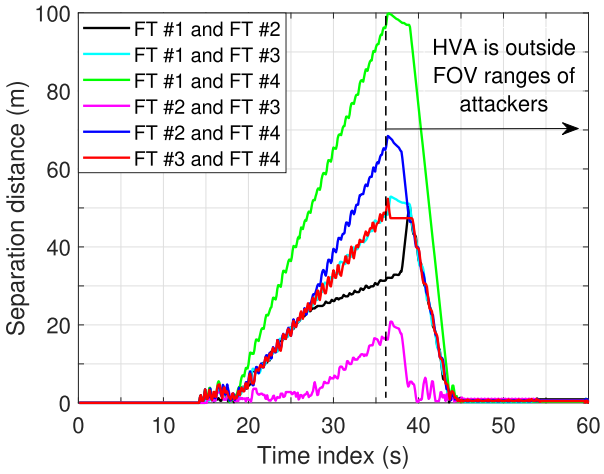


FIGURE 5. Separation distance between each FT pair versus time.

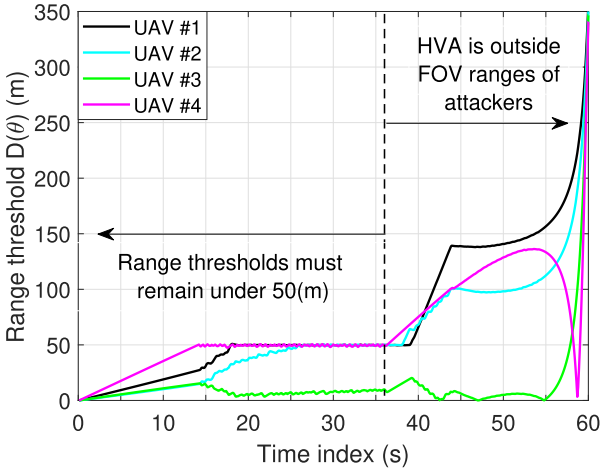


FIGURE 6. Range threshold constraint by each UAV decoy versus time.

Remark 3: The performance of the proposed algorithm is evaluated under the assumption that the communication topology between the agents is fully-connected (complete) and undirected, and all communications can be performed concurrently without delays at each time-step. Other communication topologies could be considered but are beyond the scope of this study and is left as part of future research.

Figures 3 and 4 respectively show the trajectories of the four UAVs defending against simultaneous attacks by four attackers diagonally from behind and front. As shown, to act cooperatively, the UAVs start moving from their initial positions that generate all the FTs at the same location of the HVA. Each UAV decoy then moves in a way to always maintain a position on a straight line connecting its generated FT and its associated attackers, with a fixed distance of 100 m from the FT. It can be seen that all the UAV decoys successfully seduce the four networked attackers to a same location within the honeypot region. Similar performance was observed for various approach angles of the attackers.

Figures 5 and 6 show the simulation results in term of the separation distance between the FTs and the maintained range thresholds by all the UAV decoys, respectively. The results show that in the time frame from 0 s to 15 s, all the UAVs are able to maintain their corresponding FTs moving together in the direction that increases the obtained miss-distances for all team members. However, when moving together violates the range constraint that each UAV needs to comply with, the UAV agents have no other options than to separate their FTs. Sometime later, from the 36th second onwards in the simulation as illustrated in Figures 5, the UAV team is able to regroup all the FTs at the same location again and continue to move this combined "single FT" toward the desired honeypot. This is achievable from the 36th second point as their protected target is already located outside the FOV ranges of all the enemy attackers, and thus the FTs no longer need to maintain the range constraints under 50 m as required previously as shown in Figure 6.

Figures 7 and 8 show the simulation results in term of the obtained miss-distance and the individual reward achieved by each agent versus time, respectively. The results illustrated in Figure 7 demonstrate that while completing the joint task

TABLE 1. Performance comparison of the proposed algorithm with a non-cooperative scheme under different metrics.

	TIME TO REACH HONEYPOT	WORST-CASE MISS-DISTANCE	MAXIMUM SEPARATION
NON-COOPERATIVE SCHEME	NOT APPLICABLE	843.5 ± 2.1 (m)	481.6 ± 1.7 (m)
PROPOSED ALGORITHM ($\rho = 1/T$)	44.5 ± 0.8 (s)	461.9 ± 6.5 (m)	81.3 ± 2.8 (m)
PROPOSED ALGORITHM ($\rho = 0.95$)	42.7 ± 0.7 (s)	470.6 ± 8.3 (m)	98.3 ± 2.1 (m)

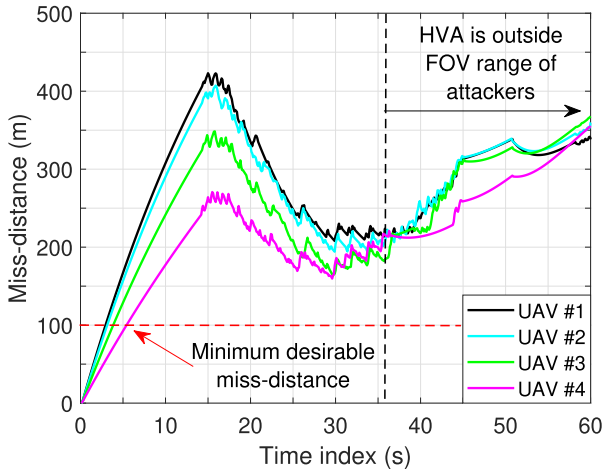


FIGURE 7. Miss-distance obtained by each UAV decoy versus time.

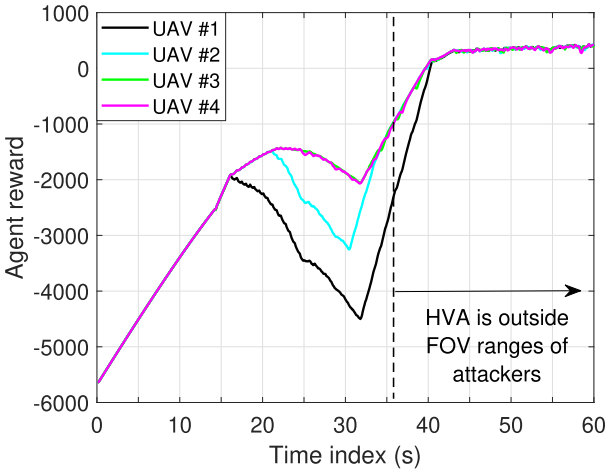


FIGURE 8. Individual reward obtained by each UAV decoy versus time.

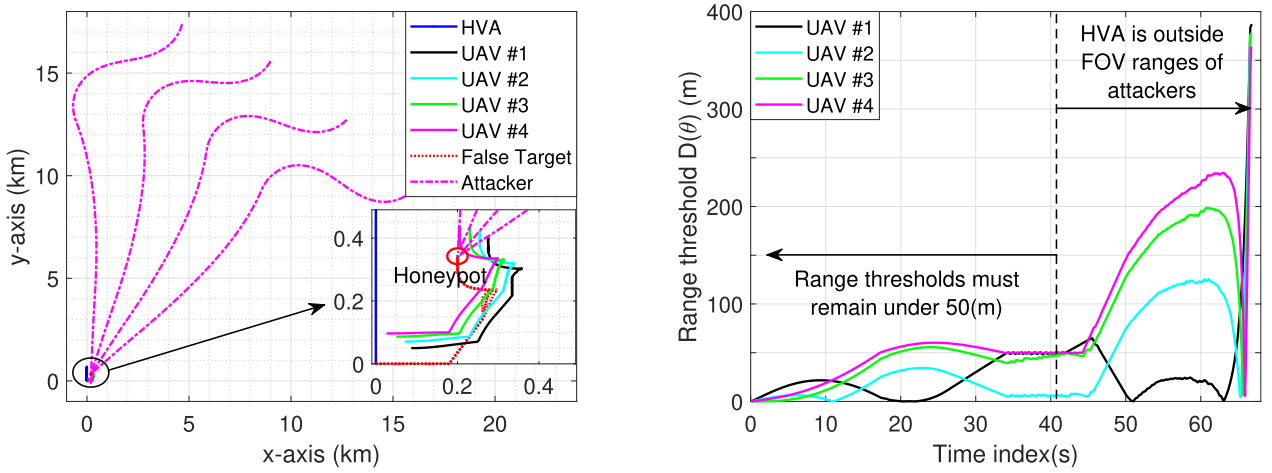
the UAV team also achieves the worst-case miss-distance above a minimum desirable value of 100 m. This is known as the satisfaction threshold an agent should obtain to keep its protected target at a safe distance from an attacker. Also, as can be seen through the redistribution of the rewards shown in Figure 8, the weaker agent (UAV #1) is able to catch up on its reward through coordination with the other stronger agents, which would otherwise act selfishly if not cooperating. As a result of cooperation, the whole UAV team obtains a globally desired outcome acceptable to all the decoy agents.

Table 1 compares the performance of the proposed algorithm with the non-cooperative scheme proposed in [20] in terms of the time required to complete the joint task (time to reach honeypot), the smallest obtained miss-distance among all the agents (worst-case miss-distance) and the largest separation between all pair of FTs (maximum separation). As can be seen from the reported results, apart from the capability of successfully seducing all the networked attackers to the desired honeypot, the proposed solution also outperforms the non-cooperative scheme in maintaining a small separation between all the FTs, which is also another important task in term of decoy maneuvers as explained in Section III-A.

In our simulation, when the threats maneuver, the agents must adapt their actions rapidly in order to maximize their payoffs. Thus, a value for the discount factor closer to 1 is chosen to prioritize the recent observed rewards in response to rapid changes in threat behaviors. There is further scope to auto-tune the ρ-values which is a topic for future investigation. As a result, by using a fixed discounted weight $\rho = 0.95$, the proposed algorithm obtains a better performance in terms of the time it takes to complete the joint task and also slightly improves the worst-case miss-distance obtained by all the agents in comparison with the traditional regret matching proposed in [3]. This achieved improvement is because the influence of observations that each agent has learned in the past is reduced. Thus, each agent is able to adapt quickly to the most recent changes in the learning environment due to the autonomous behaviors of the other agents. In return, the cost for this improvement is a slightly larger separation of the FTs. This cost is acceptable since all the FTs finally regroup and move together when the target is no longer in the FOV ranges of the attackers as illustrated in Figures 5.

For further evaluation of the computation cost of the proposed algorithm in comparison with the non-cooperative scheme in [20], computational complexities of the two algorithms in term of the size of the agent team are summarized as follows. Let T be the number of time steps (between discrete position updates) until a certain simulation time is reached. Assume that η bits are used to represent the data of position or velocity vector of any moving object.

- **Non-cooperative scheme [20]:** To compute its expected rewards at every time step, each agent needs two pieces of data (position and velocity) from three objects: the ship, its associated threat, and itself – (6η bits). Thus the computation cost per agent is independent of the size of



(a) Curved attack engagement scenario

(b) Range threshold versus time

FIGURE 9. UAV team defends against the networked threats that perform curved attack trajectories.

the team. The total computation cost of the whole team with N agents is of order $6\eta NT$ (bits) $\sim O(TN)$.

- **Proposed algorithm:** To compute its expected rewards at every time step, each agent also needs two pieces of data (position and velocity) from $(2N + 2)$ objects, including the ship, the honeypot, the N agents, and their corresponding N threats – $(2\eta(2N + 2)$ bits). Thus, the computation cost per agent depends on the number of team members. The total computation cost of the whole team is of order $4\eta(N + 1)NT$ (bits) $\sim O(TN^2)$.

The complexity analysis reveals that the proposed algorithm requires an order of magnitude more information exchange to implement because its complexity is quadratic whereas the complexity of the algorithm in [20] is linear. This cost is caused by the computation of the joint team reward to enable the cooperative behavior between the learning agents. The algorithm in [20] does not support coordination among agents and hence requires less computational cost.

The performance of the proposed algorithm is further evaluated in a more challenging scenario in which the threats perform curved attack trajectories instead of heading directly toward their target. Under this circumstance, all the threats follow curved trajectories while their radars are still tracking their (same) target. This is an attempt to distinguish whether they are seeing a real target or actually chasing an unreal target. Therefore, it is important that all the FTs generated by the decoy agents must maintain their corresponding range thresholds within 50 (m) to make the networked attackers into believing they have locked on to a unique and consistent target. Simulation results illustrated in Figures 9(a) and 9(b) confirm that the UAV team implementing the proposed solution can adapt successfully to this situation and accomplishes the team task in term of attracting all the enemy threats into the honeypot trap concurrently. This is achieved without violating the range constraint when the real target (HVA) is still within the FOV of its attackers.

VI. CONCLUSION

This paper presents a decentralized threat deflection solution for a team of autonomous decoy agents. A multi-agent cooperative solution was proposed using a regret minimization based learning framework. Each decoy agent learns to cooperatively adapt its behavior by considering joint actions and rewards, and it was theoretically proven that the convergence is guaranteed and the achieved equilibrium equates to the optimum response of every decoy in terms of maximizing the team reward. The proposed reinforcement learning model was tested in a variety of defense scenarios, including honeypot ambush tactics against saturation attacks, and it demonstrated success even in the challenging cases in which the threats followed curved trajectories. Given its online learning capability, this solution can adapt better than the offline learning techniques and also scales well with increasing number of entities. The next planned activity is to address the uncertainties and increased number of defended assets, extending to the area defense rather than point defense, which will add another dimension to the solution space.

REFERENCES

[1] L. Busoniu, R. Babuska, and B. De Schutter, "A comprehensive survey of multiagent reinforcement learning," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 38, no. 2, pp. 156–172, Mar. 2008.

[2] J. R. Marden and J. S. Shamma, "Game theory and control," *Annu. Rev. Control, Robot., Auton. Syst.*, vol. 1, no. 1, pp. 105–134, 2018.

[3] S. Hart and A. Mas-Colell, "A simple adaptive procedure leading to correlated equilibrium," *Econometrica*, vol. 68, no. 5, pp. 1127–1150, Sep. 2000.

[4] M. Jiang, T. Hai, Z. Pan, H. Wang, Y. Jia, and C. Deng, "Multi-agent deep reinforcement learning for multi-object tracker," *IEEE Access*, vol. 7, pp. 32400–32407, 2019.

[5] R. Lowe, Y. Wu, A. Tamar, J. Harb, O. P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6379–6390.

[6] C. Amato, "Decision-making under uncertainty in multi-agent and multi-robot systems: Planning and learning," in *Proc. 27th Int. Joint Conf. Artif. Intell. (IJCAI)*, Jul. 2018, pp. 5662–5666.

[7] B. Yang and M. Liu, "Keeping in touch with collaborative UAVs: A deep reinforcement learning approach," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 562–568.

[8] K. Zhang, Z. Yang, H. Liu, T. Zhang, and T. Basar, "Fully decentralized multi-agent reinforcement learning with networked agents," in *Proc. 35th Int. Conf. Mach. Learn.*, Jul. 2018, pp. 5867–5876.

[9] S. Sukhbaatar, A. Szlam, and R. Fergus, "Learning multiagent communication with backpropagation," in *Advances in Neural Information Processing Systems*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds. Red Hook, NY, USA: Curran Associates, 2016, pp. 2244–2252.

[10] H. van Seijen, M. Fatemi, J. Romoff, and R. Laroche, "Separation of concerns in reinforcement learning," Dec. 2016, *arXiv:1612.05159*. [Online]. Available: https://arxiv.org/abs/1612.05159

[11] Q. Liu, J. Ma, and W. Xie, "Multiagent reinforcement learning with regret matching for robot soccer," *Math. Problems Eng.*, vol. 2013, Jul. 2013, Art. no. 926267.

[12] V. Syrgkanis, A. Agarwal, H. Luo, and R. E. Schapire, "Fast convergence of regularized learning in games," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 2989–2997.

[13] J. R. Marden, "Selecting efficient correlated equilibria through distributed learning," *Games Econ. Behav.*, vol. 106, pp. 114–133, Nov. 2017.

[14] N. Brown and T. Sandholm, "Regret-based pruning in extensive-form games," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 1972–1980.

[15] J. Hartline, V. Syrgkanis, and E. Tardos, "No-regret learning in Bayesian games," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 3061–3069.

[16] D. D. Nguyen, L. B. White, and H. X. Nguyen, "Adaptive multiagent reinforcement learning with non-positive regret," in *Advances in Artificial Intelligence*, B. H. Kang and Q. Bai, Eds. Cham, Switzerland: Springer, 2016, pp. 29–41.

[17] D. D. Nguyen, H. X. Nguyen, and L. B. White, "Reinforcement learning with network-assisted feedback for heterogeneous RAT selection," *IEEE Trans. Wireless Commun.*, vol. 16, no. 9, pp. 6062–6076, Sep. 2017.

[18] N. Brown and T. Sandholm, "Solving imperfect-information games via discounted regret minimization," 2018, *arXiv:1809.04040*. [Online]. Available: https://arxiv.org/abs/1809.04040

[19] J. Cohen, A. Héliou, and P. Mertikopoulos, "Hedging under uncertainty: Regret minimization meets exponentially fast convergence," in *Algorithmic Game Theory*, V. Bilò and M. Flammini, Eds. Cham, Switzerland: Springer, 2017, pp. 252–263.

[20] D. D. Nguyen, A. Rajagopalan, and C.-C. Lim, "Online versus offline reinforcement learning for false target control against known threat," in *Intelligent Robotics and Applications*. Cham, Switzerland: Springer, 2018, pp. 400–412.

[21] A. Dorri, S. S. Kanhere, and R. Jurdak, "Multi-agent systems: A survey," *IEEE Access*, vol. 6, pp. 28573–28593, 2018.

[22] Y. Qu, A. Wang, and J. Liu, "Model-free cooperative control for multi-agent systems using the approximate dynamic programming approach," *IEEE Access*, vol. 6, pp. 37195–37203, 2018.

[23] Z. Zhang, Y. Shi, Z. Zhang, and W. Yan, "New results on sliding-mode control for Takagi–Sugeno fuzzy multiagent systems," *IEEE Trans. Cybern.*, vol. 49, no. 5, pp. 1592–1604, May 2019.

[24] B. Zhu, K. Guo, and L. Xie, "A new distributed model predictive control for unconstrained double-integrator multiagent systems," *IEEE Trans. Autom. Control*, vol. 63, no. 12, pp. 4367–4374, Dec. 2018.

[25] H. Li, W. Yan, and Y. Shi, "Triggering and control codesign in self-triggered model predictive control of constrained systems: With guaranteed performance," *IEEE Trans. Autom. Control*, vol. 63, no. 11, pp. 4008–4015, Nov. 2018.

[26] P. Liu, A. Kurt, and U. Ozguner, "Distributed model predictive control for cooperative and flexible vehicle platooning," *IEEE Trans. Control Syst. Technol.*, vol. 27, no. 3, pp. 1115–1128, May 2019.

[27] R. A. C. Bianchi, M. F. Martins, C. H. C. Ribeiro, and A. H. R. Costa, "Heuristically-accelerated multiagent reinforcement learning," *IEEE Trans. Cybern.*, vol. 44, no. 2, pp. 252–265, Feb. 2014.

[28] E. A. O. Diallo, A. Sugiyama, and T. Sugawara, "Coordinated behavior of cooperative agents using deep reinforcement learning," *Neurocomputing*, 2019. [Online]. Available: http://www.sciencedirect.com/science/ticle/pii/S0925231219304424. doi: 10.1016/j.neucom.2018.08.094.

[29] H. Ge, Y. Song, C. Wu, J. Ren, and G. Tan, "Cooperative deep Q-learning with Q-value transfer for multi-intersection signal control," *IEEE Access*, vol. 7, pp. 40797–40809, 2019.

[30] T. Shima and S. Rasmussen, *UAV Cooperative Decision and Control: Challenges and Practical Approaches*. Philadelphia, PA, USA: SIAM, 2009.

[31] R. W. Beard, T. W. McLain, D. B. Nelson, D. Kingston, and D. Johanson, "Decentralized cooperative aerial surveillance using fixed-wing miniature UAVs," *Proc. IEEE*, vol. 94, no. 7, pp. 1306–1324, Jul. 2006.

[32] M. Maskery, V. Krishnamurthy, and C. O'Regan, "Decentralized algorithms for netcentric force protection against antiship missiles," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 43, no. 4, pp. 1351–1372, Oct. 2007.

[33] M. Maskery and V. Krishnamurthy, "Network-enabled missile deflection: Games and correlation equilibrium," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 43, no. 3, pp. 843–863, Oct. 2007.

[34] D. D. Nguyen, H. X. Nguyen, and L. B. White, "Evaluating performance of RAT selection algorithms for 5G Hetnets," *IEEE Access*, vol. 6, pp. 61212–61222, Oct. 2018.

[35] R. Yanushevsky, *Modern Missile Guidance*. Boca Raton, FL, USA: CRC Press, 2007.

[36] S. Hart and D. Schmeidler, "Existence of correlated equilibria," *Math. Oper. Res.*, vol. 14, no. 1, pp. 18–25, Feb. 1989.

[37] H. P. Young, *Strategic Learning and Its Limits*. New York, NY, USA: Oxford Univ. Press, 2004.

[38] H. P. Borowski, J. R. Marden, and J. S. Shamma, "Learning efficient correlated equilibria," in *Proc. 53rd IEEE Conf. Decis. Control*, Dec. 2014, pp. 6836–6841.

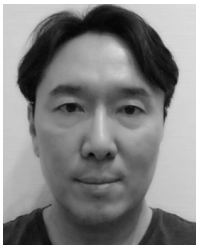
[39] M. Benaïm, J. Hofbauer, and S. Sorin, "Stochastic approximations and differential inclusions," *SIAM J. Control Optim.*, vol. 44, no. 1, pp. 328–348, 2005.

[40] M. Benaïm, J. Hofbauer, and S. Sorin, "Stochastic approximations and differential inclusions, part II: Applications," *Math. Oper. Res.*, vol. 31, no. 4, pp. 673–695, Nov. 2006.

[41] V. Krishnamurthy, *Partially Observed Markov Decision Processes: From Filtering to Controlled Sensing*. Cambridge, U.K.: Cambridge Univ. Press, 2016.

DUONG D. NGUYEN received the B.Sc. degree (Hons.) in electronic communication systems from the University of Plymouth, U.K., in 2008, the M.Sc. degree in mobile and personal communications from King's College London, U.K., in 2009, and the Ph.D. degree in engineering from The University of Adelaide, Australia, in 2018. He is currently a Postdoctoral Research Fellow with the Multi-agent Control Laboratory, School of Electrical and Electronic Engineering, The University of Adelaide. His research interests include game theory, deep reinforcement learning, and optimal control for autonomous systems and wireless networks.

ARVIND RAJAGOPALAN received the B.Eng. dual degrees (Hons.) in engineering with specialization in electrical and electronic engineering and computer science from the University of Adelaide, in 2005, and the Ph.D. degree from the University of South Australia, in 2017. He currently involves with the Algorithms Development Team, Advanced Guidance and Control Discipline, Weapons and Combat Systems Division, Defense Science and Technology Group, Australia. His research interests include autonomous systems, multi-agent systems, and guidance and control for unmanned systems.



JIJOONG KIM received the B.E. degree (Hons.) in electrical and electronic engineering (EEE) and M.Eng.Sc. degree from the University of Adelaide, Australia, in 1993 and 1995, respectively, and the Ph.D. degree in imaging processing and computer vision from the University of Wollongong, Australia, in 2006. He was a Research Assistant with the EEE Department, The University of Adelaide, for six months and then joined the Defense Science and Technology (DST) Group, Australia, in 1995. He has more than 20 years of experience in weapons guidance, navigation, and control. He is currently leading the AI and advance control research activities under collaborative and cognitive weapons program with DST. His research interests include missile guidance, optimal control, localization, filtering, machine learning, game theory, and fuzzy system.

CHENG-CHEW LIM (SM'02) received the B.Sc. (Hons.) and Ph.D. degrees in electronic and electrical engineering from Loughborough University, Leicestershire, U.K. He is currently a Professor with The University of Adelaide, Adelaide, SA, Australia. His research interests include control and systems theory, autonomous systems, machine learning, and optimization techniques and applications. Dr. Lim serves as an Editorial Board Member for the *Journal of Industrial and Management Optimization* and as an Associate Editor for the IEEE TRANSACTIONS OF SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS. He has served as a Guest Editor for a number of journals including *Discrete and Continuous Dynamical System-Series B*.