IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received April 27, 2020, accepted May 15, 2020, date of publication May 25, 2020, date of current version June 11, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2997304

IADRL: Imitation Augmented Deep Reinforcement Learning Enabled UGV-UAV Coalition for Tasking in Complex Environments

JIAN ZHANG[1], (Member, IEEE), ZHITAO YU[1,2], (Graduate Student Member, IEEE), SHIWEN MAO[2], (Fellow, IEEE), SENTHILKUMAR C. G. PERIASWAMY[1], JUSTIN PATTON[1], AND XUE XIA[2], (Graduate Student Member, IEEE)

[1]RFID Laboratory, Auburn University, Auburn, AL 36849, USA
[2]Department of Electrical and Computer Engineering, Auburn University, Auburn, AL 36849, USA

Corresponding author: Shiwen Mao (smao@ieee.org)

This work was supported in part by NSF under Grant ECCS-1923163, and in part by the RFID Laboratory and the Wireless Engineering Research and Education Center (WEREC), Auburn University, Auburn, AL, USA.

ABSTRACT Recent developments in Unmanned Aerial Vehicles (UAVs) and Unmanned Ground Vehicles (UGVs) have made them highly useful for various tasks. However, they both have their respective constraints that make them incapable of completing intricate tasks alone in many scenarios. For example, a UGV is unable to reach high places, while a UAV is limited by its power supply and payload capacity. In this paper, we propose an Imitation Augmented Deep Reinforcement Learning (IADRL) model that enables a UGV and UAV to form a coalition that is complementary and cooperative for completing tasks that they are incapable of achieving alone. IADRL learns the underlying complementary behaviors of UGVs and UAVs from a demonstration dataset that is collected from some simple scenarios with non-optimized strategies. Based on observations from the UGV and UAV, IADRL provides an optimized policy for the UGV-UAV coalition to work in an complementary way while minimizing the cost. We evaluate the IADRL approach in an visual game-based simulation platform, and conduct experiments that show how it effectively enables the coalition to cooperatively and cost-effectively accomplish tasks.

INDEX TERMS Unmanned aerial vehicle (UAV), unmanned ground vehicle (UGV), coalition, deep reinforcement learning (DRL), imitation learning.

I. INTRODUCTION

The last decade has witnessed significant developments in unmanned aerial vehicle (UAV) and unmanned ground vehicle (UGV) technologies, which have enabled their wide deployment for various applications, such as surveillance, search and rescue, inspection [1], inventory counting [2], [3], and more [4]–[7]. Recently, researchers have shown a growing interest to deploy them for more complex tasks that require multiple UAVs or UGVs to cooperatively work together to improve efficiency [8]. Most of the existing research focuses on the cooperation in a multi-agent (or multi-robot) system that consists of a group of UAVs or UGVs. For example, Koubâa *et al.* introduced COROS [9], a high-level conceptual architecture for multi-agent UGV/robotic systems that represents a generic architecture for cooperative multi-agent applications. A cooperative architecture for the navigation of a swarm of robots based on Dynamic Fuzzy Cognitive Maps was introduced in [10]–[12], which allows for the development of homogeneous autonomous robot navigation without a global controller. A multi-UAV system was introduced in [8] to optimize target assignment and path planning. In addition to these homogeneous systems, some works went further to create a system that consists of heterogeneous agents/robots with different capabilities. For example, Das *et al.* in [13] introduced a distributed algorithm for task allocation in a system of multiple heterogeneous, autonomous robots deployed in a healthcare facility.

There are some essential limitations for both UGVs and UAVs. For example, a UGV has limited vertical detective/access capability, and a UAV is restrained by inadequate

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

operation range and time due to its limited power supply capacity. These limitations impede them in many applications. For instance, a ground robot proposed in [2] failed to perform inventory counting of items stored on high racks. Recently, UAVs have been expected to be widely deployed for disaster relief (e.g., survey, search and rescue, and providing network access). However, the authors of [14] found that a UAV's limited fight time (usually 20-30 minutes) greatly reduces their operating range. Obviously, for the above scenarios, we cannot solve the problem by simply deploying a swarm of UGVs or a swarm of UAVs alone. Alternatively, to pair them as a complementary team would help to overcome these constraints for tasks that UGVs or UAVs would be incapable of completing alone. However, an effective and low-cost strategy for implementing such complementary UGV-UAV coalition is lacking.

To remedy these limitations, this paper presents an innovative method, named Imitation Augmented Deep Reinforcement Learning (IADRL), that enables a UGV and a UAV to form a coalition that can complement each other for complex tasks. The complementary UGV-UAV coalition can be deployed for applications that are usually incapable of being completed by a UGV or UAV alone. Using the disaster relief scenario as an example, an IADRL-enabled coalition can be deployed for autonomous search-and-rescue tasks. In the chaotic and hazardous environment following a disaster, a powerful UGV can autonomously carry a UAV to remote destinations usually out of the UAV's flight range. Additionally, the UGV provides communication and a power supply that greatly extends the operational range of a resource-constrained UAV, and the UAV helps the UGV with finding the best route and with navigating through complex terrains that are out of the UGV's navigational capability (e.g., vertically unreachable or invisible to the UGV). To ensure that the coalition can successfully and effectively accomplish tasks, the cooperation of its agents (i.e., the UGV and UAV) must follow an underlying and complex model that varies depending on the task or operating environment.

The proposed IADRL model can learn the complementary features of UGV-UAV from a demonstration dataset that is collected from a simple and imperfect scenario. The model also learns a policy that responds to the environment, such as collision avoidance when around obstacles and other agents. Based on observations of the UAV and UGV, the IADRL model provides a series of actions for the UGV and UAV that ensures an optimized and complimentary strategy for a given task. Additionally, we extend the IADRL to support multiple UGV-UAV coalitions working together within the same space. To the best of our knowledge, this is the first work to focus on creating such a coalition of robots with complementary capabilities for task completion, where a single agent in the team alone is incapable of completing. In a complex scenario, a task is executed by the first agent, and then another agent must continue the task based on the previous agent's success. Thus, the actions of all agents in the coalition are dependent upon each other, and agents must work as a complementary, cooperative team. The main contributions of this work are summarized as follows:

1) The proposed network enables a UGV and UAV to form a coalition to complement and enhance each other to accomplish complex tasks that either agent alone could not complete. It also optimizes the complementary coordination strategy among the agents to accomplish various tasks with the lowest cost (e.g., minimum power consumption, optimized navigational trajectory with the minimum number of steps, etc.).
2) We develop an imitation learning model to learn the intricate complementary features of UGVs and UAVs in the coalition using demonstration data that was collected from simple scenarios with non-optimized strategies. This will greatly reduce the effort of modeling the complementary behaviors of agents in the coalition.
3) We test IADRL in a visual game-based simulated environment, and show that the proposed IADRL approach exploits the complementary behaviors of UGVs and UAVs during search-related tasks and over-performs in several baseline schemes.

In the remainder of this paper, we discuss related work in Section II, introduce and analyze the proposed IADRL model in Section III, present our experimental study in Section IV, and conclude our work in Section V.

II. RELATED WORKS

A. IMITATION LEARNING

Imitation learning methods focus on the problem of learning and perform a task by learning from demonstration data. These methods can be roughly divided into three categories: Behavior Cloning (BC; or supervised learning) [15], [16], Inverse reinforcement learning (IRL) [17], and Generative Adversarial Network (GAN) imitation learning [18].

1) BEHAVIOR CLONING (BC)

This type of imitation learning was motivated by humans' tendency to learn skills by imitating the behaviors of others, and has been widely used in autonomous driving [19], [20], wireless communication [21], [22], and smart grids [23], [24]. In BC, agents receive instructions from a hand-crafted demonstrator (which serves as training data), and then replicate actions from the expert policy. BC is able to imitate the demonstrator immediately without any interaction with the environment. However, these agents cannot handle situations that are not included in the demonstrator. Furthermore, when the agents are limited in capacity, wrong or unnecessary behavior may be replicated. The method is simple, but is useful only with large amounts of high quality training data. Additionally, because agents merely learn single-step decisions, the compounding error accumulation caused by the covariate shift problem could lead to a large learning deviation.

2) INVERSE REINFORCEMENT LEARNING (IRL)

In a classic Reinforcement Learning (RL) setting, the ultimate goal is for an agent to learn a decision process to generate behaviors that could maximize accumulated rewards by some predefined reward functions. As demonstrated by Ng *et al.* in [25], IRL is given the observed agent's behaviors and observations of the environment to infer the optimal reward function. IRL generally has a reward function that is difficult to accurately quantify, and another system has to be able to complete the tasks well to offer instructions for the model. The difference between IRL and BC is that IRL generates a reward function to infer an optimal policy instead of using a fixed replication policy.

3) GAN FOR IMITATION LEARNING

Ho and Ermon proposed Generative Adversarial Imitation Learning (GAIL) in 2016 [18]. They introduced the idea of a GAN combined with imitation learning. Unlike GAN, GAIL does not have an explicit Generator that acts as the policy of agents. Learning in GAIL is divided into two steps. First, to train the Discriminator adversarially with the data obtained from the current policy sampling and expert data. Second, the Discriminator serves as the replaced reward function to train the policy. GAIL is superior for large-planning and high-dimensional problems as compared to BC and IRL.

B. MULTI-AGENT SYSTEM PLANNING AND CONTROL

This is a hot topic that has attracted considerable research interest in recent years. The existing studies have mainly focused on operating multiple UGVs/robots and UAVs in the same environment. For example, Sariel-Talay *et al.* proposed a multi-robot cooperation framework to solve complex tasks in a cost-efficient manner [26]. Swarm intelligence is inspired by social animals and aims to form the behavior of many decentralized autonomous cooperative agents. For example, Wang *et al.* solved the multi-robot task allocation problem using an ant colony algorithm [27]. In recent years, RL has become extremely trendy in the field of multi-agent systems. In [8], the author presented an innovative artificial intelligence method combined with a well-known RL method, the Multi-Agent Deep Deterministic Policy Gradient Algorithm, to solve path planning and task allocation problems in dynamic environments. However, these existing methods have never been applied to a coalition of multiple UGVs/robots and UAVs before.

Few studies have considered the use of multiple UGVs and UAVs simultaneously to solve complex tasks in dynamic environments. For example, Ghamry *et al.* proposed an algorithm that controls UAV's autonomous take-off, tracking, and landing with a UGV [28]. They also presented an interesting study on forming a team of cooperating UAVs-UGVs for forest monitoring and fire detection [29]. Khaleghi *et al.* studied the team formation approach of multiple UGVs and UAVs [30]. The author in [31] introduced an auction-based approach for applying an estimated utility to task assignment for heterogeneous, multi-agent teams. But these studies only focus on one area (i.e., team formation or task allocation) because of the huge computational cost and the communication difficulties between agents. Meanwhile, some companies (e.g., Quanser Inc.) provide a variety of mobile robots and UAV swarm systems, but none of them focus on creating a UGV-UAV coalition for complex tasks. Unlike these existing methods, our proposed approach creates a coalition that enables a UGV and a UAV to complement each other during complex tasks that are incapable of being completed by a single UGV or UAV or by a swarm of UGVs or UAVs alone. This approach not only concerns the optimization of path planning, but also learns an underlying complementary model for the agents from a set of non-optimized demonstration data.

III. THE PROPOSED APPROACH

Our proposed IADRL approach enables a coalition consisting of a UGV and UAV to complement each other for complex tasks. Additionally, we extend IADRL to include a system of multiple UGV-UAV coalitions working together.

A. IADRL ENABLED UGV-UAV COALITION

1) PROBLEM DEFINITION AND CHALLENGES

There are several essential limitations of UGVs and UAVs that prevent them from being deployed for some tasks. Fig. 1 illustrates a motivating scenario where rescue teams must reach a high-altitude position. The UAV is capable of reaching that position; however, the destination is too far for it to fly from the starting point with its limited battery capacity. Alternatively, the UGV can move closer to the destination, but is incapable of climbing up the high altitude. An intuitive idea to reach the destination is to pair the UGV and UAV together as a coalition that complements each other: the UGV can carry the UAV closer to the destination, and then the UAV launches from the UGV and flies to the target.

Motivated by the Decentralized Partially Observable Markov Decision Processes (Dec-POMDPs) [32], this

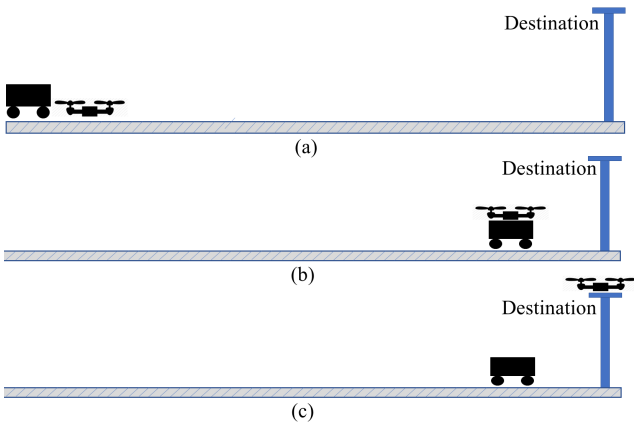

FIGURE 1. An example of a UGV-UAV complementary coalition for task completion: (a) the target destination is too far for the UAV to reach, while too high for the UGV alone, (b) the UGV carries the UAV closer to the destination, and, finally, (c) the UAV flies to the high-altitude destination.

UGV-UAV complementary coalition for task completion with minimum cost can be described by the tuple $< \varepsilon, \mathbf{o}, \mathbf{a}, r, \gamma, \mathcal{M} >$, where ε denotes the environment the coalition will interact with; $\mathbf{o} = (o_1, o_2)$ is the joint observations of the coalition, and consists of the UGV's observation, o_1, and the UAV's observation, o_2; $\mathbf{a} = (a_1, a_2)$ denotes the joint actions of the UGV, a_1, and UAV, a_2, in the coalition; r is the reward function of the coalition while joint actions $\mathbf{a}$ impose ε with joint observations $\mathbf{o}$; $\gamma \in [0, 1)$ is a discount factor for future rewards; and $\mathcal{M}$ defines the complementary cooperation model of the UGV and UAV. To achieve successful task completion, the UGV and UAV must collaborate with and complement each other; thus, their joint actions satisfy $\mathbf{a} = (a_1, a_2) \sim \mathcal{M}$.

The goal of IADRL is to learn a joint value-action function $Q_c^{\pi}(\mathbf{o}, \mathbf{a}; \theta)$ that enables a complementary UGV and UAV coalition to achieve maximum overall rewards (or minimal overall costs) while accomplishing various tasks. The equation for this complementary coalition is formulated as (1):

$$\underset{\mathbf{a}\sim\pi}{\operatorname{argmax}} Q_c^{\pi}(\mathbf{o}, \mathbf{a}; \theta) \tag{1}$$

$$\text{s.t. } \mathbf{a} = (a_1, a_2) \sim \mathcal{M}, \tag{2}$$

where θ is the parameter of the value-action function Q_c^{π}. Note that $\mathbf{o}$ and $\mathbf{a}$ represent the joint observations and actions in the coalition, and the joint actions follow an underlying model, $\mathcal{M}$, that complements each action during tasks. To explicitly model the underlying complementary cooperation model, $\mathcal{M}$, of the UGV-UAV coalition during tasks is difficult and, at least, requires significant effort and expertise.

We faced several challenges when creating the IADRL model under these requirements. For our method to successfully complete generic and complex tasks, we have to develop a straightforward way to represent the coalition's complementary cooperation model. Equation (1) shows that the proposed network has to learn an optimized policy, π, for UGV-UAV joint actions. Reference [33] suggests that the joint-action space increases exponentially with the number of agents. Consequently, it is difficult for deep reinforcement learning (DRL) methods to reach the optimized policy, π, in such huge searching space. Furthermore, the trained policy, π, not only needs to provide optimized actions for task execution, but also needs to follow the underlying model $\mathcal{M}$ to enable the UGV-UAV coalition to successfully complete tasks. State-of-the-art methods such as Value-Decomposition Networks (VDN) [34] and QMIX [33] require that the actions of agents at the same time step are independent so they can be factorized. Obviously, this assumption does not hold true for the UGV-UAV coalition. Additionally, it is necessary to train the proposed model in a continuous-action space that empowers the UGV-UAV coalition's operation in complex environments. This further increases the size of the joint-action space and challenges the training of the IADRL model.

2) THE IADRL MODEL

To tackle the above challenges, first, instead of explicitly modeling the collaboration between the UGV and UAV, we captured their complementary cooperation using a set of demonstration data. The dataset was collected by manually controlling the UGV-UAV coalition to complete several simple tasks. The demonstration data do not need optimization, but only a set of the most basic and important rules of the collaborative and complementary actions. As such, our method needs to teach the coalition just as one would teach a new sports skill to a team of kids, by showing them how to play through imitation.

Therefore, we design IADRL by combining an imitation model with a DRL model. The architecture of IADRL is presented in Fig. 2. The imitation model and the DRL model are contained in a pink block and green block, respectively. The imitation model learns the cooperative features, $\mathcal{M}$, of complementary cooperation from the non-optimized demonstration dataset and augments the DRL model's training to develop an optimized strategy. As such, we learn the optimized policy, π, while following the complementary cooperation model. Meanwhile, the DRL model also learns a strategy to respond to dynamic environments, such as avoiding collisions with obstacles and other UGVs and UAVs.

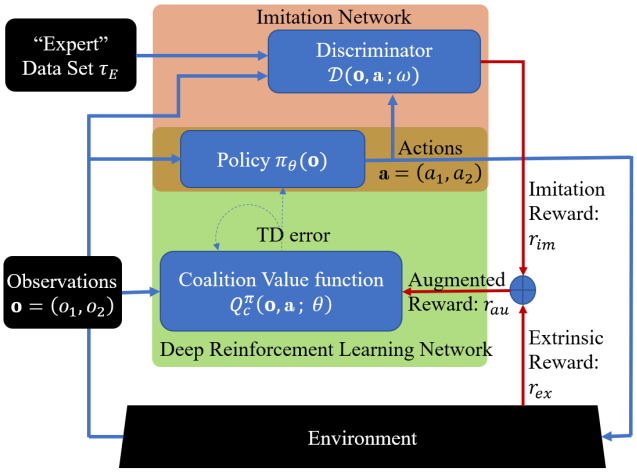

FIGURE 2. The architecture of the IADRL model.

a: THE IMITATION MODEL

The imitation model is inspired by the study of GAIL [18], and it is based on a GAN [35] architecture that comprises two basic entities: a discriminator, $\mathcal{D}$, and a generator, $\mathcal{G}$. Discriminator $\mathcal{D}$ is created to distinguish between the "expert" data and the data produced by generator $\mathcal{G}$. Additionally, $\mathcal{D}$ and $\mathcal{G}$ are simultaneously trained in an adversarial way: $\mathcal{G}$ is updated to produce "counterfeited" data that could pass the detection of $\mathcal{D}$, while $\mathcal{D}$ is improved to distinguish the "counterfeited" data from the true "expert" data. The resulting competition drives both entities to improve their capabilities. Thus, a well-trained imitation model not only generates data with almost

the same distribution of the "expert" data, but also precisely measures the similarity of any given data with the "expert" data.

Different from the original GAIL model, we replaced the Trust Region Policy Optimization-based [36] generator with the latest Proximal Policy Optimization (PPO)-based [37] generator, which also serves as the policy, π, of the DRL model. Thereby, the term generator $\mathcal{G}$ and policy π will be used interchangeably in the rest of this paper. Policy π has two roles in our IADRL, as it not only generates actions following the distribution of the "expert" data, but also reacts to the environment with an optimized strategy. The details of policy π will be introduced when we discuss the DRL model. Here, we focus on the discriminator, $\mathcal{D}(\mathbf{o}, \mathbf{a}; \omega)$, of the imitation model.

In our imitation model, $\mathcal{D} : \mathbf{O} \times \mathbf{A} \mapsto (0, 1)$ is a discriminator function with weight ω, and $\mathbf{O}$ and $\mathbf{A}$ are the observation and action space, respectively, of the UGV-UAV coalition. We implement the discriminator $\mathcal{D}$ with a deep neural network, which is a fully connected neural network with M_D hidden layers. Each hidden layer has the same number of N_D units. The size of the input layer is determined by the size of the concatenated input $(\mathbf{o}, \mathbf{a})$. The size of $\mathcal{D}$ can be configured using N_D and M_D. Usually, a larger-sized network is required if the UGV-UAV coalition is deployed for more complex environments and tasks.

During the training process, we can improve the discriminator $\mathcal{D}$ by maximizing the following value function:

$$\mathcal{V}(\omega) = \mathbb{E}_{\pi}[\log(\mathcal{D}(\mathbf{o}, \mathbf{a}; \omega))] + \mathbb{E}_{\tau_E}[\log(1 - \mathcal{D}(\mathbf{o}, \mathbf{a}; \omega))] - \lambda H(\pi), \quad (3)$$

where $H(\pi)$ represents the causal entropy [38] of π defined as $H(\pi) \equiv \mathbb{E}_{\pi}[-\log \pi(\mathbf{a}|\mathbf{o})]$, and it severs as a policy regulator to make the distribution of policy as evenly as possible; $\lambda \geqslant 0$ is the discount factor of H; and τ_E refers to the "expert" policy provided by a demonstrated dataset with length N, i.e., $\tau_E = [\eta_1, \eta_2, \ldots \eta_N]$. Here $\eta_n = [(\mathbf{o}^0, \mathbf{a}^0), (\mathbf{o}^1, \mathbf{a}^1), \ldots, (\mathbf{o}^T, \mathbf{a}^T)]$ is the record of an episode with T steps. It represents the model of the complementary cooperation between the UGV and UAV; thus, $\tau_E \sim \mathcal{M}$. Again, τ_E is not a perfect policy, but is collected from a few sample scenarios in controlled settings navigated by manual control and is, therefore, considered to be the "expert."

Equation (3) is derived from the objective function of GAIL [18]. It shows that during the training process, as discriminator $\mathcal{D}$ is updated to increase $\mathcal{V}(\omega)$, its ability to detect the similarity of a policy and the "expert" data is improved. When it produces a lower value for a given action, $\mathbf{a}$, it indicates that the chance of action $\mathbf{a}$ is higher from the "expert" data, and thus, shows with higher confidence that it is following the underlying complementary model, $\mathbf{a} \sim \mathcal{M}$.

b: THE DRL MODEL

The proposed IADRL model must not only learn the complementary cooperation model, but must also react to the dynamics of an environment and provide an optimized navigation strategy for the UGV-UAV coalition. To this end, we created the DRL model based on a PPO network [37] with an actor-critic architecture, which enables the model to produce continuous actions for the UGV-UAV coalition during task completion in complex environments. The proposed DRL model consists of two separate components: an actor (i.e., policy π) and a critic (i.e., value function Q_c^{π}). Policy π is responsible for generating action $\mathbf{a}$ based on the given observation $\mathbf{o}$. Additionally, policy π is learnt by a neural network from the training and history data. The value function, Q_c^{π}, processes the received rewards and evaluates the current action prescribed by policy π.

We implement Q_c^{π} and π using two deep neural networks that are both fully connected networks with M_{π} hidden layers for π and M_Q hidden layers for Q_c^{π}. Each hidden layer has the same number of N_{π} and N_Q units for the π and Q_c^{π} networks, respectively. The size of the input layers is determined by the size of the input vectors. The size of the output layer of π is determined by the size of the joint action, $\mathbf{a}$, of the coalition. As in the case of discriminator network $\mathcal{D}$, usually larger-sized networks are required for π and Q_c^{π} if the UGV-UAV coalition is deployed for more complex environments and tasks.

Ultimately, the goal of training the DRL model is to maximize the UGV-UAV coalition's state-value function Q_c^{π} for a given policy π, given by

$$Q_c^{\pi}(\mathbf{o}, \mathbf{a}; \theta) = \mathbb{E}[r_{au}(\mathbf{o}, \mathbf{a}) + \gamma \mathbb{E}_{\mathbf{a}' \sim \pi}[Q_c^{\pi}(\mathbf{o}', \mathbf{a}')]], \quad (4)$$

where θ is the parameter of function Q_c^{π}; $\gamma \in (0, 1]$ is the discount factor for future rewards; r_{au} is the augmented reward function, given by

$$r_{au}(\mathbf{o}, \mathbf{a}) = \beta \cdot r_{im}(\mathbf{o}, \mathbf{a}) + (1 - \beta) \cdot r_{ex}(\mathbf{o}, \mathbf{a}), \quad (5)$$

where $\beta \in (0, 1)$ represents the confidence weight of the "expert" demonstration data, and a larger β can be deployed if τ_E is closer to the optimized policy; r_{ex} is the reward function that comes from the environment, which is the same as a traditional Markov Decision Processes (MDPs) environment; additionally, r_{im} is the reward function of the imitation model and measures how similar the coalition's joint actions $\mathbf{a}$ are with the "expert policy," as

$$r_{im}(\mathbf{o}, \mathbf{a}) = \log(1 - \mathcal{D}(\mathbf{o}, \mathbf{a}; \omega)). \quad (6)$$

During the training, we aim to increase Q_c^{π}. Equations (3), (4), and (6) show that as we increase Q_c^{π}, we decrease the value of $V(\omega)$. Thus, from the results of [18], we increase the similarity of the policy, π, and the "expert" dataset, τ_E, as we increase Q_c^{π}. Note that our goal is not to train the policy, π, to copy τ_E, but to learn the complementary cooperative model that underlays τ_E while maximizing the extrinsic reward. Therefore, we introduced the confidence parameter β to augment the learning process by trading-off between learning from expert data and the environment. Alternatively, r_{ex} guides the IADRL to learn a strategy that

reacts to the environment. Its configuration is straightforward and lists several rules for the coalition when interacting with the extrinsic environment. Usually, we can assign a penalty for the coalition if any agent collides with either an obstacle or other agent. This way, a trained Q_c^π enables the UGV and UAV to choose the action that does not cause a collision. We can also assign a small penalty for every step taken by each agent, and this enables Q_c^π to provide the coalition's best navigational route for reaching a target. Here, the best route is the one with the lowest sum of navigational costs of the UGV and UAV. Note that the cost of operating a UAV is usually higher than that of the UGV. Additionally, an example of r_{ex} will be provided in the later experimental section. The value function Q_c^π of the proposed DRL network can be trained end-to-end by minimizing the following loss function:

$$\mathcal{L}(\theta) = \mathbb{E}_{\mathbf{a}\sim\pi}[y - Q_c^\pi(\mathbf{o}, \mathbf{a}; \theta)]^2, \tag{7}$$

where $y = r_{au} + \gamma \cdot \max_{\mathbf{a}'}[Q_c^\pi(\mathbf{o}', \mathbf{a}'; \theta^-)]$ and θ^- are parameters trained by the previous iteration. During the training, we try to decrease the stochastic gradient of (7) with respect to θ. Then, a trained state-value function Q_c^π precisely evaluates action $\mathbf{a}$ of the UGV-UAV coalition.

From (4) and (5), we know that as long as a policy, π, is found that guides the UGV-UAV coalition to achieve a higher cumulative Q value, the proposed IDARL network will enable the complementary cooperation between agents and find the best strategy to accomplish a given task. To better explain the process of updating the policy in our PPO-based DRL model, we introduce an additional objective function with respect to the φ weighted policy, π_φ, as:

$$J(\varphi) = \mathbb{E}_t\left[\min\left(\frac{\pi_\varphi(\mathbf{o})}{\pi_{\varphi_{old}}(\mathbf{o})}Q_c^{\pi_{\varphi_{old}}},\ f(\epsilon, Q_c^{\pi_{\varphi_{old}}})\right)\right], \tag{8}$$

where ϵ is a hyper-parameter set to 0.1 or 0.2; $\pi_{\varphi_{old}}$ and π_φ denote the policy before and after the training update, respectively; and $f(\cdot)$ is a clip function defined as:

$$\begin{cases} f(\epsilon, Q) = (1+\epsilon)Q, & \text{if } Q > 0 \\ f(\epsilon, Q) = (1-\epsilon)Q, & \text{if } Q < 0. \end{cases} \tag{9}$$

The training process aims to maximize $J(\varphi)$ by ascending the stochastic gradient of (8) with respect to φ. Thus, policy π tends to provide actions that can impose higher Q values. During the training, (9) limits the updated range of π_φ so that it remains close to the last policy, $\pi_{\varphi_{old}}$. This greatly improves training stability by avoiding too much of a policy update in one step.

We summarize the training process of IADRL in Algorithm 1. During the training process, we recursively update discriminator $\mathcal{D}$ of the imitation model to provide a more accurate evaluation of how good the complementary cooperation is between the UGV and UAV. Then, the value function, Q_c^π, is updated to enable the model to precisely assess the joint-action, $\mathbf{o}$, of the coalition as compared to the extrinsic environment and the intrinsic complementary cooperation model. Last, IADRL updates policy π that provides a series of actions to accomplish given tasks and to receive higher cumulative Q values. Thus, a well-trained IADRL model enables the UGV-UAV coalition to follow the complementary model, $\mathcal{M}$, and provides an optimized strategy when the coalition is deployed for various tasks.

Algorithm 1: The Training Procedure of IADRL

```
Input: "Expert" dataset τ_E, and initial parameters ω_0 and θ_0;
for episode i = 1 to M do
    Sample training dataset π_i;
    Update discriminator D by ascending the stochastic gradient of (3) with respect to ω;
    Update value function Q_c^π of the DRL by decreasing the stochastic gradient of (7) with respect to θ;
    Update policy π_φ of the DRL by ascending the stochastic gradient of (8) with respect to φ;
end
```

B. MULTI-COALITION SYSTEMS

Our IADRL model can be easily extended to support a system with multiple UGV-UAV coalitions. This system follows the traditional Dec-POMDPs, and the coordination among the coalitions is loose and satisfy the model of VDN [34]. Therefore, the global joint-action value function, denoted by Q_g, of a system with N coalitions can be represented as:

$$Q_g(\mathbf{s}, \mathbf{u}) = \sum_{i=1}^{N} Q_{c\ i}^\pi(\mathbf{o}_i, \mathbf{a}_i; \theta_i), \tag{10}$$

where $\mathbf{o}_i = (o_{1i}, o_{2i})$ and $\mathbf{a}_i = (a_{1i}, a_{2i})$ denote the joint observations and actions, respectively, of the UGV and UAV in coalition i. Additionally, $\mathbf{s} = (\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_N)$ and $\mathbf{u} = (\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_N)$ refer to the joint observations and actions, respectively, for all N coalitions in the system. A joint observation, $\mathbf{s}$, is created by concatenating all observations, $\mathbf{o}_i$, from all the coalitions. Equation (10) indicates that based on the current joint observation, $\mathbf{s}$, we find a best joint-action for the system, $\mathbf{u}$, by decomposing the problem and finding all of the best joint-action, $\mathbf{a}_i$, for each coalition, which is determined by the trained IADRL model based on its observation $\mathbf{o}_i$.

The UGV-UAV coalition requires wireless communications to function well. From (1), the optimized policy, π, of the coalition requires joint observation and joint action data, which are created by the observations and actions from both the UGV and the UAV. Thus, wireless communications within the coalition is essential for sharing this information. On the other hand, communications among UGV-UAV coalitions is not mandatory. In (10), it is shown that the global joint-action value function, Q_g, is the sum of individual coalition value functions, Q_c^π, which is conditional on the coalition's observations and actions. Therefore, a decentralized optimized policy for a system with multiple coalitions can be achieved when each coalition selects its own optimized

policy, π, from a trained IADRL model without sharing information among coalitions.

IV. EXPERIMENTAL STUDY AND DISCUSSIONS

A. EXPERIMENT CONFIGURATION

1) SIMULATION PLATFORM

We designed a simulation training and evaluation platform for the IADRL system based on the Unity3D ML-Agents platform [39]. The platform is illustrated in Fig. 3. It is designed to simulate the scenario of deploying UGV-UAV coalitions in a giant, high-bay warehouse crowded with high racks and shelves. The coalitions are tasked with reaching given targets to mimic item scanning applications (i.e., RFID or barcode) in indoor spaces. The platform's dimension is $50 \times 50 \times 7$ m^3, and is divided into 4 sub-zones by cross shaped obstacles. As Fig. 3 depicts, orange agents represent UGVs, blue agents represent UAVs, and the green spheres suspended in air (they are actually on different levels of racks in this space) represent given targets.

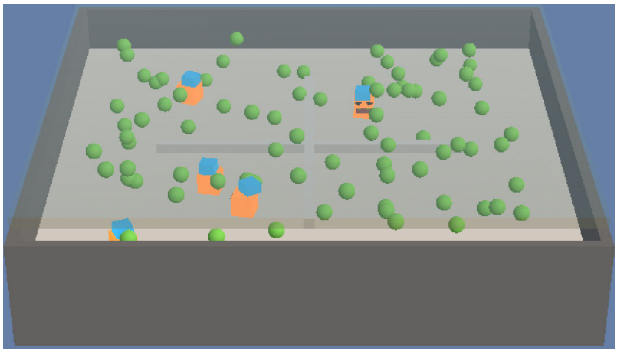

FIGURE 3. Basic simulation experimental setup for five UGV-UAV coalitions performing tasks cooperatively using the IADRL system. The UGV-UAV coalitions are marked as orange (UGV) and blue (UAV) block pairs; the tasks are marked as green balls.

We implemented our IADRL model using Tensorflow on a computer with an Intel 9900K CPU and two Nvidia 2080 GPUs. We conducted each experiment with the same IADRL configuration: the discriminator, $\mathcal{D}$, has $M_D = 2$ hidden layers and $N_D = 128$ units per layer; the coalition value function, Q_c^π, has $M_Q = 3$ hidden layers and $N_Q = 512$ units per layer; the policy, π, has $M_\pi = 3$ hidden layers and $N_\pi = 512$ units per layer. In the following experiments, we deployed 5 UGV-UAV coalitions. Their initial positions and the positions of all targets were randomly generated.

The observations (or states of the environment) are collected by each agent's Ray-cast sensor, which is provided by Unity3D. Similar to a Lidar sensor (e.g., the RPLidar laser scanner), the Ray-cast sensor casts rays into the surrounding environment, and the feedback is a vector that provides the position of all detected objects and their distances. A UGV's Ray-cast sensor detects only in the horizontal direction (to identify obstacles on the floor), while a UAV casts rays towards the horizon, and upward and downward within 45 vertical degrees. The maximum detection range of all Ray-cast sensors is set to 20 meters with a 20-Hz refresh rate. A UGV-UAV coalition's observation, $\mathbf{o}$, is created by concatenating all of the observation vectors of its UGV and UAV agents to form a new vector. The UGV's action is represented by $a_1 = [a_x, a_y]$, and the UAV's action is represented by $a_2 = [a_x, a_y, a_z]$, where a_x, a_y, and a_z are accelerations in the x, y, and z direction, respectively. The UGV-UAV coalition's action, $\mathbf{a} = (a_1, a_2)$, is also created by concatenating a_1 and a_2 to form a new vector.

2) EXTRINSIC REWARDS

The extrinsic rewards configuration is summarized in Table 1. They are designed to capture basically every condition that could be experienced when deploying UGV-UAV coalitions for item scanning tasks. Considering that the average battery life of a UGV is 5 to 10 times that of a UAV, we set the UAV's cost of each step to be 6 times that of the UGV. Thus, the UAV tends to ride on the UGV when transiting between positions, while simultaneously finding the best trajectory to reach the destination by trading-off from the ride-on to fly state. To encourage coalitions to complete tasks, we set the reward of reaching each target to 100 times that of the step cost for UAVs. Our intention is for the UGV to successfully scan all the targets within its reachable vertical height and define them as bad targets for the UAV. If the UAV mistakenly reaches a bad target, a penalty as big as the reward (i.e., 60) will be issued. Targets that are too high and out of the UGV's reach are considered good targets for the UAV.

TABLE 1. Extrinsic rewards configuration.

Reward Items	*Reward Value*
UGV's step cost	-0.1
UAV's step cost	-0.6
UGV reaches a target	+60
UAV reaches a bad target	-60
UAV reaches a good target	+60
UAV collides with an obstacle	-60
UAV collides with another agent	-60
UGV collides with an obstacle	-30
UGV collides with another agent	-30
Final reward	+30

To ensure that the UGV and UAV avoid colliding into obstacles and other agents, the penalty for a collision is equal to the target reward (i.e., 60) for the UAV and half of that for the UGV. The reason for setting a lower penalty for the UGV is that UGVs are usually protected with anti-collision sensors or bumpers. When the coalition reaches all targets (or the given number of targets), it has completed the task and wins a final reward. We set the confidence weight β in (5) to 0.1 for the remaining experiments.

3) DEMONSTRATION DATA COLLECTION

The demonstration dataset τ_E is collected by manually controlling a UGV-UAV coalition through several simple scenarios that are displayed in Fig. 4. The dataset τ_E consists of 40 total episodes of completed tasks (10 tasks per scenario)

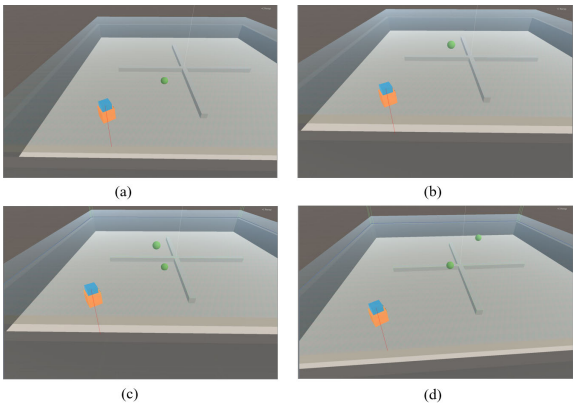

FIGURE 4. The scenarios that allow for collection of demonstration data τ_E: (a) a target (green ball) within reachable height of the UGV, (b) a target reachable only by the UAV, (c) one target each for the UAV and UGV to reach within the same sub-zone, (d) one target each for the UAV and UGV to reach, but within different sub-zones.

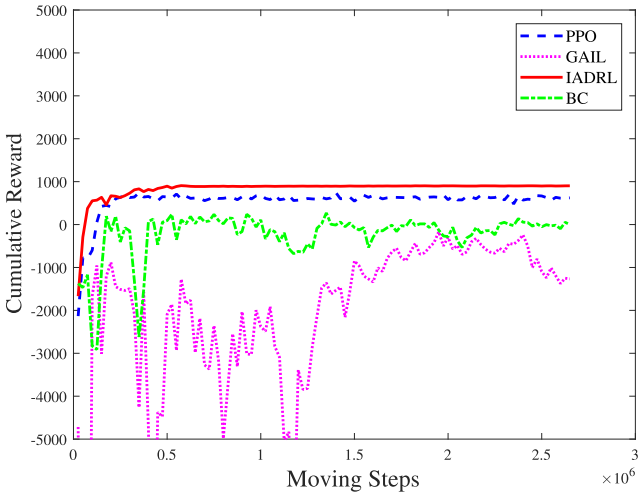

FIGURE 5. Accumulated training rewards values for PPO, GAIL, IADRL, and BC methods.

according to the scenarios described in Fig. 4 (10 for each scenario). For the scenario shown in Fig. 4(a), a target is created within the reachable height of the UGV, and we controlled the coalition in a way that allowed the UGV to reach the target. In Fig. 4(b), a target at a higher place is generated for the UAV to reach. The UAV first rides on the UGV to move closer to the target, and then flies to the target to scan it. In Fig. 4(c), we create two targets, one for the UAV and the other for the UGV, in the same sub-zone. Again, we navigated the coalition so the UGV and UAV could reach their targets cooperatively. Fig. 4(d) is a scenario similar to the scenario in Fig. 4(c), but we place the two targets in different sub-zones. Note that the targets in each scenario are generated randomly.

During this process, we manually controlled the coalition with some non-optimized strategies. For example, we do not optimize the route when moving towards any target. For the scenarios in Figs. 4(c) and 4(d), we do not consider the order of targets for optimizing the moving trajectory. Thus, τ_E serves as an instructor that guides all agents to learn complimentary behavior patterns rather than only copying the sample actions provided in the training stage.

B. EXPERIMENTAL RESULTS

1) TRAINING PROCESS RESULTS

In the training process, the maximum number of steps, st_{max}, for one episode is 1×10^5, which includes the steps of the UGV-UAV coalition. If the coalitions reach all of the targets, the training episode is terminated immediately and the final reward is received. Otherwise, it will keep tasking until st_{max} is reached. As a baseline scheme for performance comparison, we implemented three existing models, including:

- the original GAIL model, termed GAIL, introduced in [18];
- the PPO model, termed PPO, presented in [37]; and
- a supervised learning method, termed BC (Behavior Cloning) from [16].

Moreover, to guarantee a fair comparison, we used the same training parameter (i.e., number of targets achieved, learning rate, maximum number of steps, etc.) for the three approaches.

First, we conducted several experiments with five coalitions using the four models. As shown in Fig. 5, the accumulated rewards of IADRL and PPO are convergent, while the GAIL and BC curves do not converge. Obviously, compared to the other three algorithms, the cumulative reward value of the IADRL approach is the highest and it is the most stable given the same reward settings. This result is consistent with our preliminary theoretical conjecture that GAIL only replicates the behaviors and policy offered by the demonstration dataset τ_E, rather than by the optimal policy for achieving higher rewards. Although the cumulative reward obtained by the PPO model is high and convergent, it cannot successfully complete all the cooperative tasks. This is because PPO is incapable of learning the complementary model between the UGV and UAV. Fig. 6 shows that all episodes of the PPO model are terminated when they reach the maximum number of steps, $st_{max} = 10^5$, and, thus, are incapable of successfully reaching all the targets. The task completion rate for the PPO model is consistently zero, indicating that the model is not able to provide an optimized policy that enables the UGV-UAV coalition to complete tasks exploiting complementary cooperation. Therefore, in the following section, we will not discuss the performance of PPO. Furthermore, Fig. 7 shows the training loss values during the training process. It is clear that the loss values of IADRL, GAIL, and BC are significantly minimized after $st_{max} = 10^5$ steps are completed.

Note that every training episode will be terminated if all targets are reached before the maximum number of steps are taken. Thus, the average steps to complete an episode varies for each models. To compare the models and better present the training process, the results in Figs. 5 and 7 are obtained with different numbers of steps for the models.

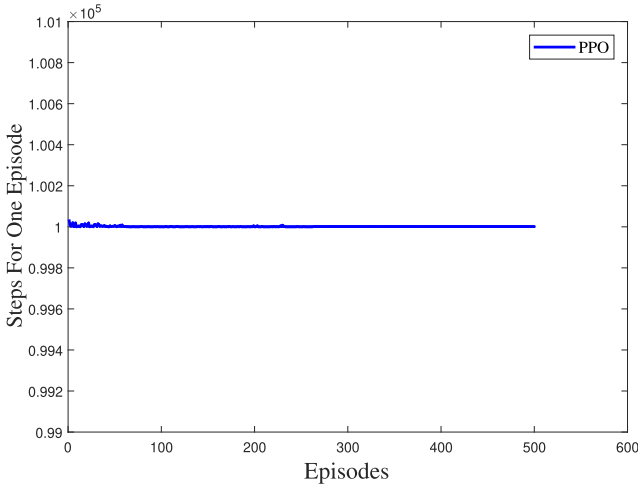

FIGURE 6. The number of steps taken before episodes are terminated for the PPO model.

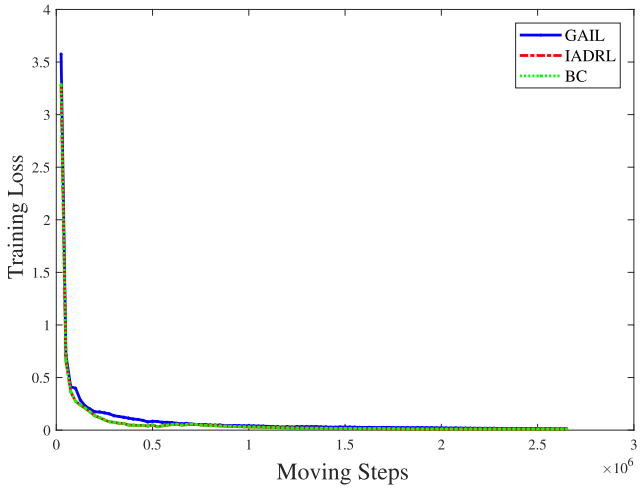

FIGURE 7. Training loss values of GAIL, BC, and IADRL methods.

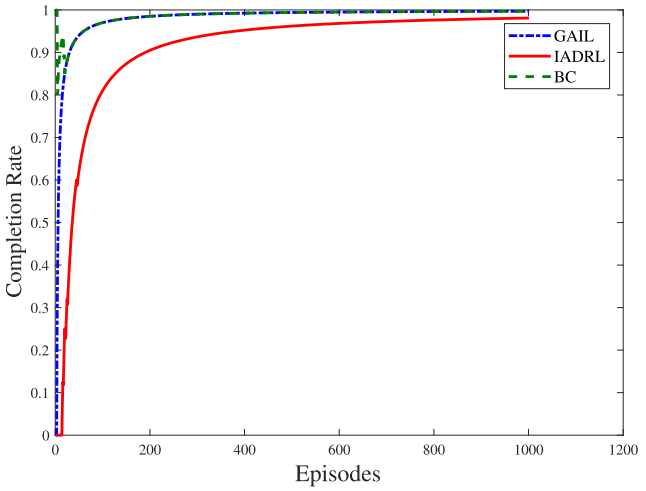

FIGURE 8. Task completion rate, $\Re_{task}$, for GAIL, BC, and IADRL methods.

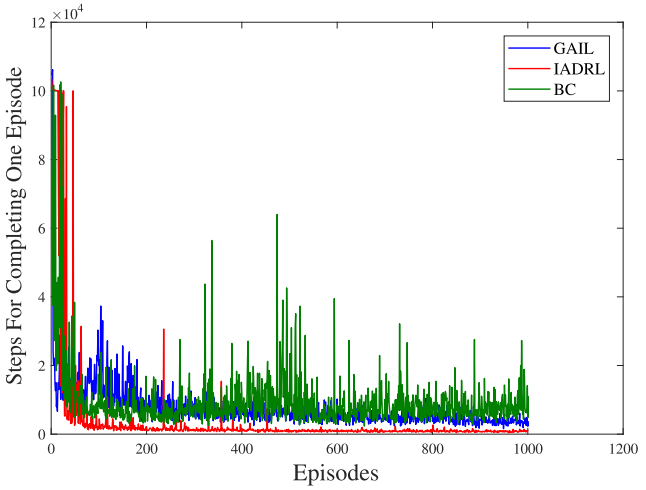

FIGURE 9. Number of steps needed to complete each training episode using IADRL, GAIL, and BC methods.

2) PERFORMANCE ANALYSIS

To further prove the superiority of IADRL, we evaluate three additional indicators: (i) number of collisions in one episode, (ii) steps needed for completing one episode, and (iii) the overall task completion rate. Fig. 8 describes the task completion rate, denoted by $\Re_{task}$, for IADRL, GAIL, and BC. We defined the task completion rate as:

$$\Re_{task} = \frac{N_{failed}}{N_{total}}, \tag{11}$$

where N_{failed} is the number of episodes that the UGV-UAV coalitions fail to reach all targets, and N_{total} is the total number of completed episodes. For our task setting, the key point towards completing a mission is the complementary cooperation between UGVs and UAVs. Fig. 8 shows that the task completion rate $\Re_{task}$ of IADRL quickly converges to 1, which indicates that after it fails in the first several episodes, IADRL quickly learns the complementary model from τ_E and succeeds in all the subsequent episodes. The three curves close to each other illustrates that IADRL has a similar capability of learning a model from τ_E to that of GAIL and BC, which are designed to directly replicate the policy from demonstration data.

To evaluate the efficiency of tasking, we compare the number of steps taken to reach all the targets with these three schemes, and the results are presented in Fig. 9. Obviously, IADRL achieves the given tasks within 600 steps for each episode, which is far less than the number of steps GAIL and BC take given the same mission. Furthermore, the number of steps required for IADRL training is much more sustainable than that of GAIL and BC, as it reaches the optimized policy within fewer episodes (around 200 training episodes). Furthermore, the BC method not only uses the most steps to complete tasks, but even at the end of the training, no convincing task completion strategies have been determined, as shown by the large fluctuations at the tail end of the BC curve.

Collision avoidance is a key factor when deploying UGV-UAV coalitions for many applications, and, therefore, the number of collisions for all agents is a critical gauge for measuring the quality of our work. According to Fig. 10,

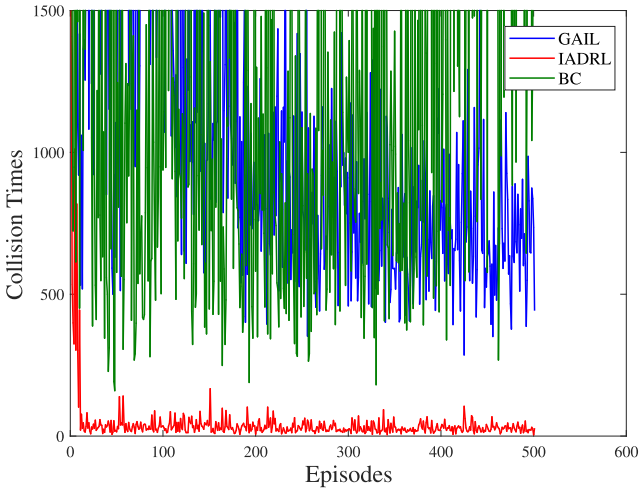

FIGURE 10. Total number of agent collisions with GAIL, BC, and IADRL methods.

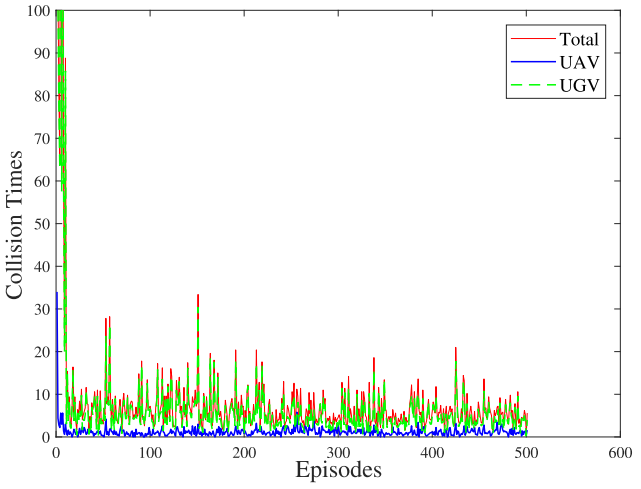

FIGURE 11. A composition of the number of collisions by UAVs and UGVs using the IADRL model, where the UGV collisions are dominant and UAV collisions are minimal.

GAIL and BC perform poorly when avoiding collisions. To better present results, we limited the range of the y-axis to [0,1500]. In the early training stages of GAIL and BC, there are many poor performance results, and some even exceed 4000 times that of IADRL. After training convergence, the number of collisions of IADRL for all agents in each episode is reduced to very low levels compared to that of GAIL and BC.

Additionally, we plotted the total number of collisions, the number of UGV collisions, and the number of UAV collisions in Fig. 12. As shown, UGVs experience the most collisions, and the more vulnerable UAVs work safely in a majority of cases. This is consistent with our initial design that the penalty for UGV collision is only half of that of a UAV's, as established in Table 1. Note that in real deployment scenarios, UGVs are more robust to collisions than UAVs, as most UGVs are equipped with bumpers and bumper sensors that help them protect against and avoid collision. Furthermore, UGVs utilize collisions to detect and navigate around the surrounding environment (e.g., iRobot Roomba Vacuums).

After a further analysis, we find that the collisions are mainly caused by the sparse observation of UGV and UAV, as agents in IADRL are not able to detect obstacles and other agents. Although this result is already acceptable for many real-world robotic applications, we are confident that the addition of more sensory information to our system would allow for a much better performance on avoiding collisions.

To illustrate the path planning performance of each scheme, we designed a simple test with two targets, one located at $(-5, -5, 5)$ and the other at $(-5, -5, 1)$[1]. The planned paths for the three schemes obtained in five trials are plotted in Fig. 12. An optimized strategy for the coalition would have the lowest cost associated with reaching both targets. The UAV should ride on the UGV as close to the

[1]Otherwise, the planned paths would be hard to plot and see.

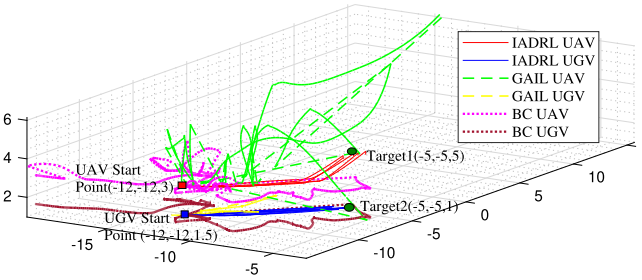

FIGURE 12. Planned paths for the UGV and UAV to reach two objects in five trials as computed by IADRL, GAIL, and BC schemes.

first target as possible, and then fly to reach the first target. The UGV should then continue on to reach the second target. Due to the physical size differences of UAVs and UGVs, we ploted their trajectories individually. For the trajectories generated by IADRL, we can see that the initial parts of the red lines (UAV) are parallel to the blue lines (UGV) because the UGV carries the UAV during this interval. The five lines for the UAV and UGV are for each of the five trials. Obviously, the path planning of our proposed algorithm enables the UGV-UAV coalition to reach targets with an optimized route at a greatly reduced cost than that of GAIL or BC methods. Additionally, each IADRL planned route is almost identical in every trial, further proof of its stable performance.

3) ROBUSTNESS IN DIFFERENT ENVIRONMENTS

The proposed IADRL scheme is robust to changes in the environment and can be directly deployed in an environment different from where it was trained. As such, we train the model in an environment similar to Fig. 3 and deploy it in a more complex environment shown in Fig. 13. We add more obstacles, marked in red in Fig. 13, to simulate a warehouse with higher obstacle density. The same UGV-UAV coalitions with the well-trained IADRL model are deployed in this new environment to complete the same missions. We then compare the previous results with that in Fig. 3 using the three

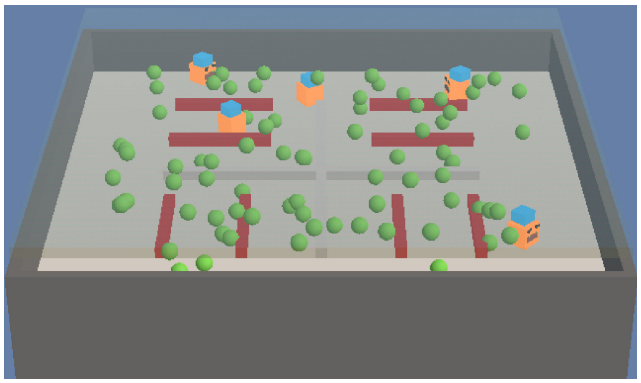

FIGURE 13. A complex simulation environment representing a high-density warehouse.

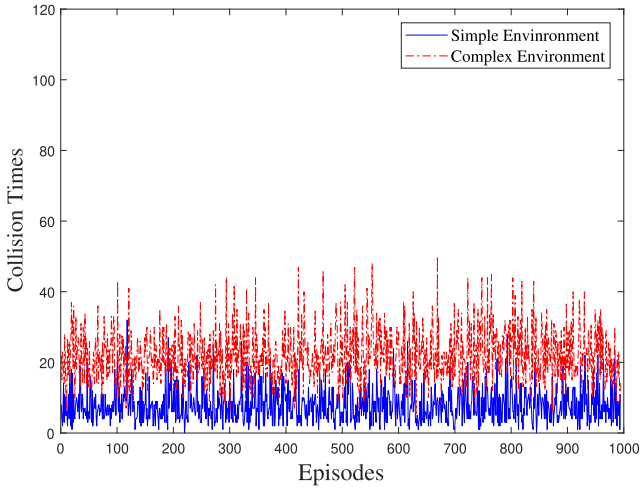

FIGURE 14. Number of collisions in the simple and complex environments.

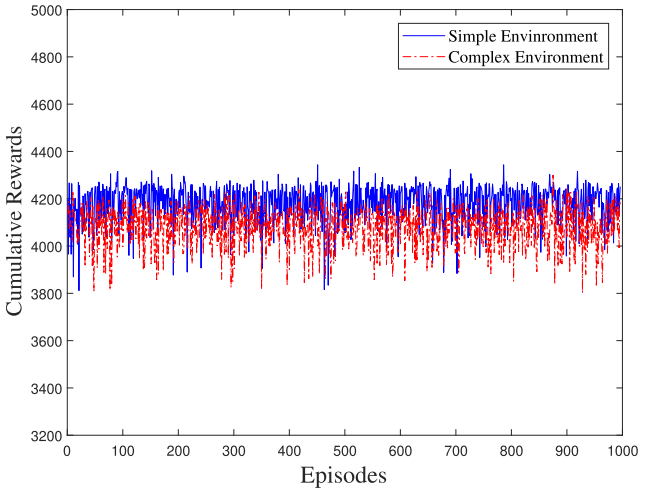

FIGURE 15. Accumulated rewards in the simple and complex environments.

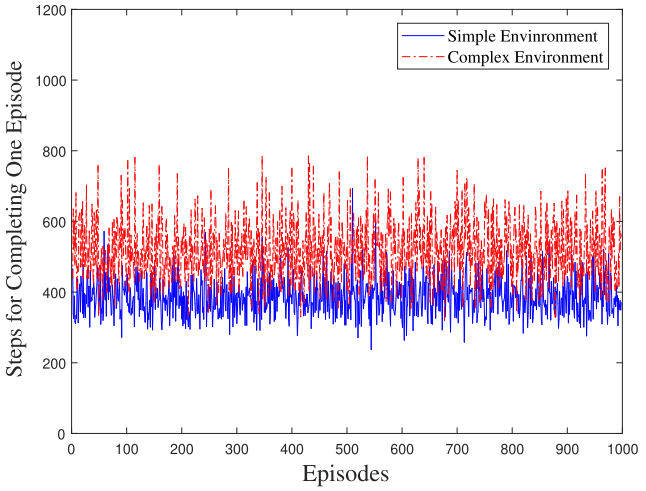

FIGURE 16. Number of steps needed to complete one episode in the simple and complex environments.

measurements introduced in the section IV-B2. To guarantee the credibility of comparison results, all parameters, including reward settings, materials, and shape of agents, are kept identical in these two environments. In the rest of this section, we will refer to the results of experiments in Fig. 13 as the "Complex Environment," whereas the result in Fig. 3 will be referred to as the "Simple Environment."

Fig. 14 depicts the number of collisions during the testing process. Even challenged by higher environmental complexity, the number of collisions for each episode is only slightly increased due to the increased complexity of the environment. We also note that there is only a slight decrease in the accumulated reward values in the complex environment as compared to the simple environment, as illustrated in Fig. 15. Additionally, we investigate the amount of steps needed to complete the tasks in each episode. The results displayed in Fig. 16 show that it takes about 200 more steps for the coalitions to accomplish all tasks in the complex environment. These observations meet our expectations, as coalitions require more steps to bypass extra obstacles in the complex environment, and, thus, have a higher step cost and a decrease in accumulated rewards.

V. CONCLUSIONS

This paper presented IADRL, a novel method that enables UGVs and UAVs to form a coalition for the complementary accomplishment of tasks that neither the UAV or UGV could not complete independently. IADRL learns the complementary behavior features of the UGV-UAV coalition from a demonstration dataset that can be readily collected from some simple and imperfect settings alike. It also optimizes the strategy to achieve given goals with minimum overall costs required to complete task in dynamic environments. We also extended the IADRL model to facilitate the cooperation of multiple UGV-UAV coalitions deployed together for complex tasks. The experimental results proved that the proposed IADRL approach was effective for solving intricate tasks requiring heterogeneous agents to complement each other in dynamic environments.

ACKNOWLEDGMENT

(Jian Zhang and Zhitao Yu are co-first authors.)

REFERENCES

[1] T. Roppel, Y. Lyu, J. Zhang, and X. Xia, "Corrosion detection using robotic vehicles in challenging environments," in *Proc. CORROSION*, New Orleans, LA, USA, Mar. 2017, pp. 1–14.
[2] J. Zhang, Y. Lyu, T. Roppel, J. Patton, and C. P. Senthilkumar, "Mobile robot for retail inventory using RFID," in *Proc. IEEE Int. Conf. Ind. Technol. (ICIT)*, Taipei, Taiwan, Mar. 2016, pp. 101–106.
[3] X. Xia, T. Roppel, J. Zhang, Y. Lyu, S. Mao, S. C. G. Periaswamy, and J. and Patton, "Enabling a mobile robot for autonomous RFID-based inventory by multilayer mapping and ACO-enhanced path planning," *J. Robot. Autom. Technol.*, vol. 1, no. 1, pp. 1–13, Sep. 2019.
[4] Q. Wu, Y. Zeng, and R. Zhang, "Joint trajectory and communication design for multi-UAV enabled wireless networks," *IEEE Trans. Wireless Commun.*, vol. 17, no. 3, pp. 2109–2121, Mar. 2018.
[5] J. Zhang, Z. Yu, X. Wang, Y. Lyu, S. Mao, S. C. Periaswamy, J. Patton, and X. Wang, "RFHUI: An intuitive and easy-to-operate human-UAV interaction system for controlling a UAV in a 3D space," in *Proc. MobiQuitous*, New York, NY, USA, Nov. 2018, pp. 69–76.
[6] J. Zhang, Z. Yu, X. Wang, Y. Lyu, S. Mao, S. C. Periaswamy, J. Patton, and X. Wang, "RFHUI: An RFID based human-unmanned aerial vehicle interaction system in an indoor environment," *Digit. Commun. Netw.*, vol. 6, no. 1, pp. 14–22, Feb. 2020.
[7] J. Zhang, X. Wang, Z. Yu, Y. Lyu, S. Mao, S. C. Periaswamy, J. Patton, and X. and Wang, "Robust RFID based 6-DoF localization for unmanned aerial vehicles," *IEEE Access*, vol. 7, pp. 77348–77361, 2019.
[8] H. Qie, D. Shi, T. Shen, X. Xu, Y. Li, and L. Wang, "Joint optimization of multi-UAV target assignment and path planning based on multi-agent reinforcement learning," *IEEE Access*, vol. 7, pp. 146264–146272, 2019.
[9] A. Koubâa, M.-F. Sriti, H. Bennaceur, A. Ammar, Y. Javed, M. Alajlan, N. Al-Elaiwi, M. Tounsi, and E. Shakshuki, *Coros: A Multi-Agent Software Architecture for Cooperative and Autonomous Service Robots*. Cham, Switzerland: Springer, 2015, pp. 3–30, ch. 1.
[10] M. Li, J. Harris, M. Chen, S. Mao, Y. Xiao, W. Read, and B. Prabhakaran, "Architecture and protocol design for a pervasive robot swarm communication networks," *Wireless Commun. Mobile Comput.*, vol. 11, no. 8, pp. 1092–1106, Aug. 2011.
[11] M. Li, K. Lu, H. Zhu, M. Chen, S. Mao, and B. Prabhakaran, "Robot swarm communication networks: Architectures, protocols, and applications," in *Proc. 3rd Int. Conf. Commun. Netw.*, Hangzhou, China, Aug. 2008, pp. 162–166.
[12] M. Mendonäa, I. R. Chrun, F. Neves, and L. V. R. Arruda, "A cooperative architecture for swarm robotic based on dynamic fuzzy cognitive maps," *Eng. Appl. Artif. Intell.*, vol. 59, pp. 122–132, Mar. 2017.
[13] G. P. Das, T. M. McGinnity, S. A. Coleman, and L. Behera, "A distributed task allocation algorithm for a multi-robot system in healthcare facilities," *J. Intell. Robot. Syst.*, vol. 80, no. 1, pp. 33–58, Nov. 2014.
[14] M. Erdelj and E. Natalizio, "UAV-assisted disaster management: Applications and open issues," in *Proc. Int. Conf. Comput., Netw. Commun. (ICNC)*, Kauai, HI, USA, Feb. 2016, pp. 1–5.
[15] M. Bain and C. Sammut, "A framework for behavioural cloning," in *Proc. Mach. Intell.*, Oxford, U.K., Jul. 1995, pp. 103–129.
[16] F. Torabi, G. Warnell, and P. Stone, "Behavioral cloning from observation," 2018, *arXiv:1805.01954*. [Online]. Available: http://arxiv.org/abs/1805.01954
[17] S. Arora and P. Doshi, "A survey of inverse reinforcement learning: Challenges, methods and progress," 2018, *arXiv:1806.06877*. [Online]. Available: http://arxiv.org/abs/1806.06877
[18] J. Ho and S. Ermon, "Generative adversarial imitation learning," in *Proc. NIPS*, Barcelona, Spain, Dec. 2016, pp. 4565–4573.
[19] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, J. Zhao, and K. Zieba, "End to end learning for self-driving cars," 2016, *arXiv:1604.07316*. [Online]. Available: http://arxiv.org/abs/1604.07316
[20] F. Codevilla, E. Santana, A. Lopez, and A. Gaidon, "Exploring the limitations of behavior cloning for autonomous driving," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Seoul South Korea, Oct. 2019, pp. 9329–9338.
[21] Y. Sun, M. Peng, Y. Zhou, Y. Huang, and S. Mao, "Application of machine learning in wireless networks: Key techniques and open issues," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3072–3108, 4th Quart., 2019.
[22] X. Wang, Z. Yu, and S. Mao, "Indoor localization using smartphone magnetic and light sensors: A deep LSTM approach," *Mobile Netw. Appl.*, vol. 25, no. 2, pp. 819–832, Apr. 2020.
[23] L. Wang, S. Mao, B. Wilamowski, and R. M. Nelms, "Ensemble learning for load forecasting," *IEEE Trans. Green Commun. Netw.*, vol. 4, no. 2, pp. 616–628, Jun. 2020, doi: 10.1109/TGCN.2020.2987304.
[24] Y. Wang, Y. Shen, S. Mao, X. Chen, and H. Zou, "LASSO and LSTM integrated temporal model for short-term solar intensity forecasting," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2933–2944, Apr. 2019.
[25] A. Y. Ng, "Algorithms for inverse reinforcement learning," in *Proc. ICML*, Stanford, CA, USA, Jun./Jul. 2000, pp. 663–670.
[26] S. Sariel-Talay, T. R. Balch, and N. Erdogan, "A generic framework for distributed multirobot cooperation," *J. Intell. Robotic Syst.*, vol. 63, no. 2, pp. 323–358, May 2011.
[27] J. Wang, Y. Gu, and X. Li, "Multi-robot task allocation based on ant colony algorithm," *J. Comput.*, vol. 7, no. 9, pp. 2160–2167, Apr. 2012.
[28] K. A. Ghamry, Y. Dong, M. A. Kamel, and Y. Zhang, "Real-time autonomous take-off, tracking and landing of UAV on a moving UGV platform," in *Proc. 24th Medit. Conf. Control Autom.*, Athens, Greece, Jun. 2016, pp. 1236–1241.
[29] K. A. Ghamry, M. A. Kamel, and Y. Zhang, "Cooperative forest monitoring and fire detection using a team of UAVs-UGVs," in *Proc. Int. Conf. Unmanned Aircr. Syst. (ICUAS)*, Arlington, VA, USA, Jun. 2016, pp. 1206–1211.
[30] A. M. Khaleghi, D. Xu, S. Minaeian, M. Li, Y. Yuan, J. Liu, Y.-J. Son, C. Vo, and J.-M. Lien, "A dddams-based UAV and UGV team formation approach for surveillance and crowd control," in *Proc. Winter Simulation Conf.*, Savanah, GA, USA, Dec. 2014, pp. 2907–2918.
[31] C. E. Pippin and H. Christensen, "A Bayesian formulation for auction-based task allocation in heterogeneous multi-agent teams," in *Proc. Ground/Air Multisensor Interoperability, Integr., Netw.*, Orlando, FL, USA, May 2011, Art. no. 804710.
[32] F. A. Oliehoek, M. T. J. Spaan, and N. Vlassis, "Optimal and approximate Q-value functions for decentralized POMDPs," *J. Artif. Intell. Res.*, vol. 32, pp. 289–353, May 2008.
[33] T. Rashid, M. Samvelyan, C. Schroeder de Witt, G. Farquhar, J. Foerster, and S. Whiteson, "QMIX: Monotonic value function factorisation for deep multi-agent reinforcement learning," 2018, *arXiv:1803.11485*. [Online]. Available: http://arxiv.org/abs/1803.11485
[34] P. Sunehag, "Value-decomposition networks for cooperative multi-agent learning based on team reward," in *Proc. Int. Conf. Auton. Agents MultiAgent Syst. (AAMAS)*, Stockholm, Sweden, Jul. 2018, pp. 2085–2087.
[35] I. Goodfellow, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, Montreal, QC, Canada, Dec. 2014, pp. 2672–2680.
[36] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *Proc. ICML*, Lille, France, Jul. 2015, pp. 1889–1897.
[37] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017, *arXiv:1707.06347*. [Online]. Available: http://arxiv.org/abs/1707.06347
[38] M. Bloem and N. Bambos, "Infinite time horizon maximum causal entropy inverse reinforcement learning," in *Proc. 53rd IEEE Conf. Decision Control*, Los Angeles, CA, USA, Dec. 2014, pp. 4911–4916.
[39] A. Juliani, V.-P. Berges, E. Teng, A. Cohen, J. Harper, C. Elion, C. Goy, Y. Gao, H. Henry, M. Mattar, and D. Lange, "Unity: A general platform for intelligent agents," 2018, *arXiv:1809.02627*. [Online]. Available: http://arxiv.org/abs/1809.02627

JIAN ZHANG (Member, IEEE) received the B.Sc. and M.Sc. degrees in applied physics from Sichuan University, Chengdu, China, in 2001 and 2008, respectively, and the Ph.D. degree in electrical and computer engineering from Auburn University, Auburn, AL, USA, in 2016. He is currently an Assistant Research Professor with the RFID Laboratory, Auburn University. His main research interests include RFID technologies and applications, the Internet of Things, indoor localization, UAV, and collaborative robotics. His work focuses on improving the efficiency of supply chain management for industry and business.

ZHITAO YU (Graduate Student Member, IEEE) received the B.Sc. degree in electrical engineering from the Nanjing University of Posts and Telecommunication, Nanjing, China, in 2016, and the M.Sc. degree in electrical and computer engineering from Auburn University, Auburn, AL, USA, in 2018, where he is currently pursuing the Ph.D. degree in ECE. His research interests include indoor localization, deep learning, and indoor UAV navigation.

SHIWEN MAO (Fellow, IEEE) received the Ph.D. degree in electrical and computer engineering from Polytechnic University, Brooklyn, NY, USA (now New York University Tandon School of Engineering).

He joined the Department of Electrical and Computer Engineering, Auburn University, Auburn, AL, USA, as an Assistant Professor, in 2006, and held the McWane Professorship, from 2012 to 2015. He is currently the Samuel Ginn Endowed Professor and the Director of the Wireless Engineering Research and Education Center and the NSF IUCRC FiWIN Center Site, Auburn University. His research interests include wireless networks, multimedia communications, and smart grid.

Dr. Mao is a member of ACM. He received the IEEE ComSoc TC-CSR Distinguished Technical Achievement Award, in 2019; the IEEE ComSoc MMTC Distinguished Service Award, in 2019; the Auburn University Creative Research and Scholarship Award, in 2018; the 2017 IEEE ComSoc ITC Outstanding Service Award; the 2015 IEEE ComSoc TC-CSR Distinguished Service Award; the 2013 IEEE ComSoc MMTC Outstanding Leadership Award; and the NSF CAREER Award, in 2010. He was a co-recipient of the IEEE ComSoc MMTC 2018 Best Journal Paper Award, the IEEE ComSoc MMTC 2017 Best Conference Paper Award, the Best Demo Award from the IEEE SECON 2017, the Best Paper Awards from the IEEE GLOBECOM 2019, 2016, and 2015, the IEEE WCNC 2015, and the IEEE ICC 2013, and the 2004 IEEE Communications Society Leonard G. Abraham Prize in the field of communications systems. He was a Distinguished Speaker, from 2018 to 2021. He was a Distinguished Lecturer of the IEEE Vehicular Technology Society, from 2014 to 2018. He is an Area Editor of the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, the IEEE OPEN JOURNAL OF THE COMMUNICATIONS SOCIETY, the IEEE INTERNET OF THINGS JOURNAL, the IEEE/CIC CHINA COMMUNICATIONS, and *ACM GetMobile*. He is an Associate Editor of the IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING, the IEEE TRANSACTIONS ON MOBILE COMPUTING, the IEEE MULTIMEDIA, and the IEEE NETWORKING LETTERS.

SENTHILKUMAR C. G. PERIASWAMY received the Ph.D. degree in computer science from the University of Arkansas, Fayetteville, in 2010. He is currently the Director of technology with the RFID Laboratory, Auburn University, a unique collaboration platform that involves end users, suppliers, technology providers, standards organizations, industry groups, and academic institutions on a global scale. He has researched, advised, and executed projects that enable the efficient adoption of RFID and sensor fusion in retailing, aerospace, manufacturing, and transportation. His work has focused on the common goal of making the adaptation of RFID and related sensor technologies more secure, efficient, reliable, and useful.

JUSTIN PATTON is currently the Director of the RFID Laboratory, Auburn University, a research institute focusing on the business case and technical implementation of emerging technologies in retail, supply chain, aerospace, and manufacturing. The RFID Laboratory is a unique private/academic partnership between users, technology vendors, standards organizations, and faculty. He has participated in business case research for advanced technology with Walmart, Target, Amazon, FedEx, Dillard's, Macy's, Delta Air Lines, and Boeing, and is also researching upstream supply chain benefits of RFID in both retail and manufacturing. He is one of the primary developers of the ARC Program, the first and most widely utilized international performance validation system for RFID, and also working to standardize the process of testing and certifying RFID performance in all aspects of the supply chain.

XUE XIA (Graduate Student Member, IEEE) received the bachelor's degree in communication engineering from the University of Electronic Science and Technology of China, Chengdu, China, in 2013, and the master's degree in electrical engineering from Auburn University, Auburn, AL, USA, in 2016, where she is currently pursuing the Ph.D. degree in electrical and computer engineering. Her research interests include robotics, SLAM, computer vision, and path planning.