Learning-Based UAV Trajectory Optimization With Collision Avoidance and Connectivity Constraints

Xueyuan Wang and M. Cenk Gursoy, *Senior Member, IEEE*

***Abstract*— Unmanned aerial vehicles (UAVs) are expected to be an integral part of wireless networks, and determining collision-free trajectories for multiple UAVs while satisfying requirements of connectivity with ground base stations (GBSs) is a challenging task. In this paper, we consider non-cooperative multi-UAV scenarios, in which multiple UAVs need to fly from initial locations to destinations, while satisfying collision avoidance, wireless connectivity, and kinematic constraints. We aim to find trajectories for the UAVs with the goal to minimize their mission completion time. We first formulate the multi-UAV trajectory optimization problem as a sequential decision making problem. We, then, propose a decentralized deep reinforcement learning approach to solve the problem. More specifically, a value network is developed to obtain values given the agent's joint state (including the agent's information, the nearby agents' observable information, and the locations of the nearby GBSs). A signal-to-interference-plus-noise ratio (SINR)-prediction neural network is also designed, using accumulated SINR measurements obtained when interacting with the cellular network, to map the GBSs' locations into the SINR levels in order to predict the UAV's SINR. Numerical results show that with the value network and SINR-prediction network, real-time navigation for multi-UAVs can be efficiently performed in various environments with high success rate.**

***Index Terms*— Collision avoidance, decentralized algorithms, deep reinforcement learning, multi-UAV trajectory design, wireless connectivity.**

I. Introduction

UNMANNED aerial vehicles (UAVs), also commonly known as drones, are aircrafts piloted by remote control or embedded computer programs without human onboard [2]. Recently, UAVs have found numerous applications, such as aerial inspection, photography, precision agriculture, traffic control, search and rescue, package delivery, and telecommunications. UAVs in certain applications will be regarded as aerial user equipments (UEs) that need to be supported by the ground communication infrastructure, which brings both opportunities and challenges to cellular communications. These aerial UEs can be referred to as *cellular-connected UAVs* that access the cellular network from the sky for data communications [3]. As aerial UEs, the UAVs need efficient trajectories and also should keep connected with ground base stations (GBSs) during their flights. Therefore, the trajectory of cellular-connected UAVs need to be carefully designed to meet their mission specifications, while at the same time ensuring that the communication requirements are satisfactorily met.

In scenarios involving multiple UAVs or more generally multiple autonomous systems, a fundamental challenge is to safely control the interactions with other dynamic agents in the environment. Specifically, it is important for the autonomous devices (e.g., robots and drones) to navigate in an environment with or without obstacles, and stay free of collisions with each other and the obstacles, based on local observations of the environment. Finding solutions to this problem is challenging, since one robot's action is based on others' motions (intents) and policies which are in general unknown, and, furthermore, explicit communication of such hidden quantities is often impractical due to physical limitations. Earlier works have largely leveraged well-engineered interaction models to enhance the social awareness in robot navigation, e.g. [4] and [5], where the same policy is applied to all agents. The key challenge for these models is that they heavily rely on hand-crafted functions and cannot generalize well to various scenarios for crowd-like cooperation. As an alternative, reinforcement learning frameworks have been used to train computationally efficient policies that implicitly encode the interactions and cooperation among agents. Recent works, e.g., [6]–[9], have shown the power of deep reinforcement learning techniques to learn socially cooperative policies.

A. Related Prior Work

Trajectory design for cellular-connected UAVs has been extensively investigated in the literature. For instance, the authors in [10] studied the trajectory design for a single cellular connected UAV under delay-limited communication. The authors in [11] applied convex optimization and linear programming to find the optimal set of waypoints and speed for a UAV to ensure the minimum connection time constraint with the ground terminals. A circular trajectory with optimized flight radius and speed for a UAV was considered in [12] to maximize the energy efficiency. [13] aimed to find UAV path planning strategy to optimize the wireless coverage for the UAV. In [14], the UAV trajectory optimization was

Manuscript received November 25, 2020; revised June 9, 2021 and August 27, 2021; accepted November 8, 2021. Date of publication November 30, 2021; date of current version June 10, 2022. An earlier version of this paper was presented in part at the IEEE International Conference on Communications (ICC), Montreal, QC, in 2021, [DOI: 10.1109/ICC42927.2021.9500368]. The associate editor coordinating the review of this article and approving it for publication was P. Li. *(Corresponding author: M. Cenk Gursoy.)*

The authors are with the Department of Electrical Engineering and Computer Science, Syracuse University, Syracuse, NY 13244 USA (e-mail: xwang173@syr.edu; mcgursoy@syr.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TWC.2021.3129226.

Digital Object Identifier 10.1109/TWC.2021.3129226

studied to minimize the total propulsion related power consumption while satisfying a cellular-connectivity constraint. A connectivity-aware UAV path planning problem was formulated in [15] to find the shortest path subject to connectivity constraints. Three-dimensional (3D) path planning for a cellular-connected UAV was studied in [16] to minimize its flying distance from initial to final locations, while satisfying an expected signal-to-interference-plus-noise ratio (SINR) requirement, and also an SINR map was constructed. In [17], the authors formulated a problem to minimize the UAV mission completion time by jointly optimizing the UAV trajectory and UAV-GBS association order. In addition, the authors in [18], [19] considered how to determine the optimal path for the UAV to minimize its mission completion time, subject to wireless connectivity constraint. Optimization techniques, graph theory and dynamic programming were used to solve the single-UAV path planning problems formulated in these prior studies.

Reinforcement learning (RL) has also been utilized to obtain solutions to trajectory optimization for cellular-connected UAVs in the literature. The authors in [20] proposed a double Q-learning method to solve the UAV trajectory optimization problem under a maximum continuous disconnection time constraint or a total disconnection time constraint. In [21], a dueling double deep Q network with multi-step learning algorithm was formulated as a solution to the UAV trajectory optimization problem to minimize the weighted sum of its mission completion time and expected communication outage duration. Additionally, an interference-aware path planning scheme for a network of cellular-connected UAVs was proposed in [22] to achieve a trade-off between maximizing energy efficiency and minimizing both wireless latency and the interference. A deep reinforcement learning algorithm, based on echo state network cells, was developed to solve the problem. However, none of these prior works considered collision avoidance constraints.

Multi-UAV control with reinforcement learning techniques has also been investigated in the literature. For example, the authors in [23] studied the joint problem of dynamic multi-UAV altitude control and multi-cell wireless channel access management of Internet of Things (IoT) devices. Online model-free constrained deep reinforcement learning (CDRL) algorithm based on Lagrangian primal-dual policy optimization was proposed to solve the problem. In [24], the authors developed a deep reinforcement learning (DRL)-based self-regulation approach to maximize the accumulated UE satisfaction score in multi-UAV networks with UAV and UE dynamics. Moreover, authors in [25] proposed learning-based algorithms to solve the problem of joint trajectory design and power control for multiple UAVs with the goal to maximize the instantaneous sum transmission rate of mobile UEs. However, collision avoidance constraint was not taken into account in these papers.

Different approaches for the collision avoidance of multiple UAVs have also been developed in the literature. For instance, a rolling horizon approach using dynamic programming was used to solve the problem in a multi-agent cooperative system in [26]. A neuro-dynamic programming algorithm was proposed in [27] for multi-UAV cooperative path planning. A mixed integer linear programming method was used in [28]. Partially observable Markov decision process based methods were applied in [29]–[31] for UAV collision avoidance. In addition, authors in [32] used reachable sets to represent the collection of possible trajectories of the obstacle aircraft. Once a collision was detected, a sampling-based method was used to generate a collision avoidance path for the UAV. In [33], predictive state space was utilized to present the waypoints of the UAVs, with which initial collision-free trajectories were generated and then improved by a rolling optimization algorithm to minimize the trajectory length. Artificial potential field method with an additional control force was proposed for multi-UAV path planning in [34]. The authors in [35] presented path planning algorithms using rapidly-exploring random trees to generate paths for multiple UAVs in obstacles rich environment. Moreover, the authors in [36] used DDQN algorithm to solve the UAV path planning problem for data collection from distributed IoT nodes. However, considering collision avoidance in multiple cellular-connected UAV navigation with wireless communication requirements, addressing these challenges via deep reinforcement learning methods, and obtaining decentralized solutions have not been adequately explored yet.

B. Contributions

Motivated by these facts, we propose a decentralized deep reinforcement learning algorithm as a solution to the multi-UAV trajectory optimization problem with collision avoidance and wireless connectivity constraints. The contributions of the paper are summarized as follows:

- We study multi-UAV trajectory optimization to minimize the UAVs' mission completion time under realistic constraints, e.g., collision avoidance, wireless connectivity, and kinematic constraints, while also taking into account antenna patterns and interference levels.
- We formulate the problem as a sequential decision making problem, and develop a decentralized deep reinforcement learning algorithm to solve it. More specifically, we formulate the problem as a Markov decision process (MDP) with properly designed state space, action space, and reward function. We optimize the value function of the MDP to find the optimal policy, and design a value neural network to approximate the value function.
- Due to the fact that the UAVs do not communicate with each other in the considered network, uncertainty exists in the UAVs' unobservable intents. Thus, we employ a velocity filter to estimate the UAVs' intentions to address this uncertainty. In addition, we further design an SINR-prediction neural network to estimate the SINRs experienced at the UAVs.
- We delineate the initialization, refining, and training steps of the algorithm and describe the real-time navigation process. We extensively evaluate the proposed decentralized deep reinforcement learning algorithm. We demonstrate that with the introduction of the SINR-prediction network, testing environment is not restricted to be the

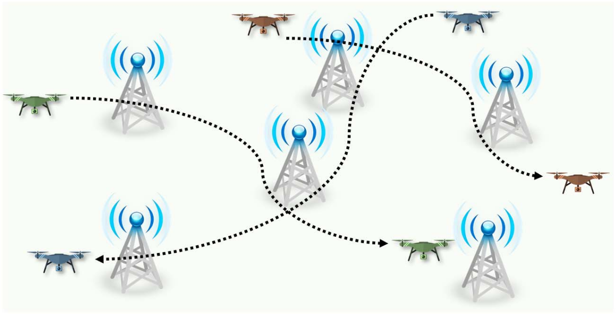


Fig. 1. An illustration of multi-UAV multi-GBS cellular networks.

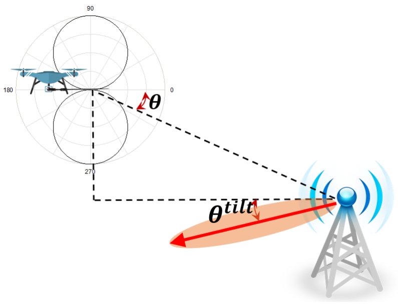


Fig. 2. Illustrations of the antenna patterns of the UAVs and the GBSs.

same as the training environment. Furthermore, we show that the real-time decentralized navigation of multiple UAVs can be efficiently performed with high success rate in various environments, e.g., environments with different antenna patterns, environments with obstacles or no-fly zones. Moreover, we demonstrate that the proposed algorithm with its collision awareness can significantly reduce the collision rates.

The remainder of the paper is organized as follows: System model is introduced in Section II. Section III describes the multi-UAV trajectory optimization problem, including the considered constraints. Section IV focuses on the reinforcement learning framework for solving the proposed problem and the approaches used to tackle the uncertainty in the environment. The decentralized deep reinforcement learning algorithm is presented in Section V in detail. In Section VI, numerical and simulation results are provided to evaluate the performance of the proposed algorithm. Finally, concluding remarks are given in Section VII.

II. System Model

In this section, we introduce the system model of the multi-UAV and multi-GBS cellular networks in detail. Note that in this section, unless specified otherwise, we remove the time index e.g., in the position vector $\mathbf{p}(t) \rightarrow \mathbf{p}$, and the index for UAVs or GBSs, e.g., $\mathbf{p}_i \rightarrow \mathbf{p}$.

A. Deployment

We consider multi-UAV multi-GBS cellular networks as displayed in Fig. 1, in which J UAVs, with potentially different missions, need to fly from starting locations to destinations over an area containing K GBSs. Without loss of generality, we assume that the area of interest is a cubic volume, which can be specified by $C : \mathcal{X}\times\mathcal{Y}\times\mathcal{Z}$ and $\mathcal{X} \triangleq [x_{\min}, x_{\max}]$, $\mathcal{Y} \triangleq [y_{\min}, y_{\max}]$, and $\mathcal{Z} \triangleq [z_{\min}, z_{\max}]$. Each UAV is modeled as disc-shaped with radius r. Let $\mathbf{p} = [p_x, p_y, H_V]$ denote the 3D position of the UAV, which is the center of the disc. H_V is the altitude of the UAVs, and is assumed to be fixed. $\mathbf{p}^S = [p_{sx}, p_{sy}, H_V] \in \mathbb{R}^3$ and $\mathbf{p}^D = [p_{gx}, p_{gy}, H_V] \in \mathbb{R}^3$ are used to denote the coordinates of the starting points and destinations, respectively.

Each UAV's state is composed of an observable information vector and an unobservable (hidden) information vector, $\mathbf{s} = [\mathbf{s}^o, \mathbf{s}^h]$, where the observable state can be observed by other UAVs, while the unobservable state cannot. In the global frame, observable state includes the UAV's position, velocity $\mathbf{v} = [v_x, v_y]$, and radius r, i.e., $\mathbf{s}^o = [\mathbf{p}, \mathbf{v}, r] \in \mathbb{R}^6$.The unobservable state consists of the destination $\mathbf{p}^D$, maximum speed $v_{\max}$, and orientation ϕ, i.e., $\mathbf{s}^h = [\mathbf{p}^D, v_{\max}, \phi] \in \mathbb{R}^5$. It is worth noting that the UAVs do not communicate with other UAVs. Hence, we address a more challenging non-communicating scenario.

In this cellular network, there are K GBSs providing wireless coverage simultaneously. The k^{th} GBS has transmit power P_{B_k}, and it is located at position $\mathbf{p}_{B_k} = [p_{x_{B_k}}, p_{y_{B_k}}, H_B]$, where H_B is the height of the GBS and is assumed to be the same for all GBSs.

B. Antenna Configuration

The GBSs and the UAVs are equipped with directional antennas with fixed radiation patterns, which are shown in Fig. 2.

1) GBS: We assume that the antenna elements of the GBSs are only directional along the vertical dimension but omni-directional horizontally [2]. Along the vertical dimension, the signal is usually downtilted toward the ground to cover the ground users and suppress the intercell interference. Therefore, the antenna gain can be expressed as [37]

$$\begin{aligned} G_B(d) &= G_h + G_v \text{ (dB)} \\ &= 10^{-\min\left(-1.2\left(\frac{\arctan(\frac{H_B-H_V}{d})-\theta^{tilt}}{\theta^{3dB}}\right)^2, \frac{G_m}{10}\right)} \end{aligned} \tag{1}$$

where

$$G_h = 0 \text{ (dB)} \tag{2}$$

$$G_v(d) = -\min\left(12\left(\frac{\arctan(\frac{H_B-H_V}{d})-\theta^{tilt}}{\theta^{3dB}}\right)^2, G_m\right) \text{ (dB)}. \tag{3}$$

Above, G_m is the maximum attenuation of the antennas, d is the horizontal distance between the UAV and the GBS, θ^{tilt} and θ^{3dB} represent antenna downtilting angle and the vertical 3dB beamwidth of the antennas at the GBSs, respectively.

2) UAV: The UAVs are assumed to be equipped with a receiver with a horizontally oriented antenna, and a simple analytical approximation for antenna gain provided by UAVs can be expressed as [38]

$$G_V(d) \ = \sin(\theta) = \frac{H_V - H_B}{\sqrt{d^2 + (H_V - H_B)^2}} \tag{4}$$

where θ is elevation angle between the UAV and GBS, H_V is the UAV altitude, and H_B is the height of the GBS.

C. Path Loss

For cellular-connected UAVs, due to high UAV altitude, UAV-to-GBS channels usually constitute strong line-of-sight (LOS) links, and LOS links are dominant [2], [39]. In addition, if the UAV altitude H_V is greater than a threshold, 3GPP specifications suggest a LOS link with probability one. For example, in the 3GPP specifications in [40], the altitude threshold is suggested to be 40m for RMa (Rural Macro) deployment, and 100m for UMa (Urban Macro) deployment. Therefore, we assume that all links between the UAVs and GBSs are LOS. The path loss can be expressed as

$$L(d) = \left(d^2 + (H_B - H_V)^2\right)^{\alpha/2} \tag{5}$$

where α is the path loss exponent.

D. SINR and Connectivity

The UAVs receive signals from all GBSs, among one of which is the serving BS, and others contribute to the interference. The received signal from the k^{th} GBS to the i^{th} UAV can be expressed as $P_k G_{B_k}(d_{ik}) G_{V_i}(d_{ik}) L^{-1}(d_{ik})$. The experienced SINR at the i^{th} UAV if it is associated with the k^{th} GBS can be expressed as

$$\mathcal{S}_{r_{i,k}} \triangleq \frac{P_k G_{B_k}(d_{ik}) G_{V_i}(d_{ik}) L^{-1}(d_{ik})}{\mathcal{N}_s + \sum_{k' \neq k} P_{k'} G_{B_{k'}}(d_{ik'}) G_{V_i}(d_{ik'}) L^{-1}(d_{ik'})} \tag{6}$$

where $\mathcal{N}_s$ is the noise power. If the experienced SINR at a UAV is larger than a threshold $\mathcal{T}_s$, then the UAV is regarded as connected with the cellular network, and disconnected otherwise.

E. SINR Measurement

Along the path to destination, UAVs interact with the cellular network, measure the raw signal from GBSs, and obtain the instantaneous SINR $\mathcal{S}'_r([\mathbf{p}, \mathbf{s}_B]; h)$, where h includes the random small-scale fading coefficients with all GBSs, and $\mathbf{p}$ and $\mathbf{s}_B = [\mathbf{p}_{B_k}, \forall k]$ are the position of the UAV and positions of all GBSs, respectively. These measurements can be obtained by leveraging the existing soft handover mechanisms with continuous reference signal received power (RSRP) and reference signal received quality (RSRQ) [21]. At each time t, over a very short time interval, during which the agents' locations can be approximately considered to be unchanged, it is assumed that the UAV performs N_m SINR measurements. Then the empirical SINR can be obtained as

$$\widehat{\mathcal{S}}_{r_{(t)}} = \frac{1}{N_m} \sum_{n=1}^{N_m} \mathcal{S}'_{r_{(t)}}([\mathbf{p}(t), \mathbf{s}_B]; h_{(t),n}). \tag{7}$$

To average over the randomness arising from small-scale fading, we can consider large N_m and have $\lim_{N_m \to \infty} \widehat{\mathcal{S}}_{r_{(t)}} = \mathcal{S}_{r_{(t)}}$ by applying the law of large numbers. Therefore, as long as the UAV performs signal measurements sufficiently frequently so that $N_m \gg 1$, $\mathcal{S}_{r_{(t)}}$ can be evaluated by its empirical value $\widehat{\mathcal{S}}_{r_{(t)}}$.

III. Multi-UAV Trajectory Optimization

In this section, we first introduce the constraints and then formulate the multi-UAV trajectory optimization problem.

A. Constraints

1) Collision Avoidance: Collision avoidance is central to many autonomous systems. During flight, the UAVs should not collide with others, which means that the distance between two UAVs should be larger than the sum of their radii all the time, i.e.,

$$||\mathbf{p}_i(t) - \mathbf{p}_j(t)||_2 > r_i + r_j \quad \forall j \neq i, \forall t \tag{8}$$

where $\mathbf{p}_i(t)$ is the location of the i^{th} UAV at time t, and r_i is its radius. Note that this radius can also include a buffer zone in which no other UAV should be present.

2) Wireless Connectivity Constraint: To support the command, control and also data flows, UAVs have to maintain a reliable communication link to the GBSs. To achieve this goal, we consider the connectivity constraint for the UAVs, i.e., the maximum contiguous time duration that the UAV is disconnected should not be longer than $\mathcal{T}_t$. The maximum continuous disconnected time duration can be mathematically expressed as

$$T_O^{\max} = \max_{t \in [0,T]} t - T_L(t) \tag{9}$$

where T is the total travel time, and $T_L(t)$ is the last time that the UAV is connected with the cellular network before time t, i.e.,

$$\begin{aligned} T_L(t) = \max \quad & \tau \\ \text{s.t.} \quad & \tau \in [0, t] \\ & \mathcal{S}_r(\tau) \geq \mathcal{T}_s. \end{aligned} \tag{10}$$

Therefore, the connectivity constraint can be written as

$$\left(\max_{t \in [0,T]} t - T_L(t) \right) \leq \mathcal{T}_t. \tag{11}$$

3) Initial and Final Locations: Each UAV starts its mission from a given initial location and completes its flight at a given destination, i.e.,

$$\mathbf{p}(0) = \mathbf{p}^S \text{ and } \mathbf{p}(T) = \mathbf{p}^D \tag{12}$$

4) Kinematic Constraints: Kinematic constraints need to be considered for operating UAVs. We impose the speed and rotational constraints as follows:

$$\mathbf{v}(t) = [v_s(t), \phi(t)] \tag{13}$$

$$\text{Speed limit: } v_s(t) \leq v_{\max} \tag{14}$$

$$\text{Rotation limit: } |\phi_(t) - \phi(t - \Delta t)| \leq \Delta t \cdot \mathcal{T}_r \tag{15}$$

where $\mathbf{v}(t)$, $v_s(t)$ and $\phi(t)$ are the UAV's velocity, speed and orientation at time t, respectively. $v_{\max}$ is the maximum speed of the UAV, and $\mathcal{T}_r$ is the maximum angle that a UAV can rotate in unit time period. This constraint limits the direction that a UAV can travel at a given time.

5) Association Constraint: Each UAV is associated with one GBS at a time, and the associated GBS is denoted by $a(t) \in \{1, \ldots, K\}$. Largest received signal power based association is adopted in this paper, where

$$a(t) = \underset{k}{\operatorname{argmax}} P_k G_{B_k}(d_k(t)) G_V(d_k(t)) L^{-1}(d_k(t)). \tag{16}$$

B. Problem Formulation in Continuous Time Domain

The goal of this work is to find trajectories for all UAVs in the network such that the travel/flight time of each UAV between the initial and final locations is minimized, while the constraints are satisfied. In the considered decentralized setting, the trajectory optimization problem for the i^{th} UAV can be formulated as

$$\begin{aligned} (\text{P0}): \ & \underset{\{\mathbf{p}_i(t), \forall t\}}{\operatorname{argmin}} \ T_i \\ & s.t. \ (8), (11), (12), (14), (15) \\ & \quad a(t) \in \{1, \ldots, K\}, \forall t \end{aligned}$$

C. Problem Formulation in Discrete Time Domain

Since the UAV is not permitted to be disconnected continuously for more than $\mathcal{T}_t$ time units, it is sufficient to consider $\Delta t = \mathcal{T}_t / n_t$ as one time step and address the problem every n_t time steps. If, at these specific time instances, the experienced SINRs at all UAVs are higher than $\mathcal{T}_s$, we can guarantee that the connectivity constraint is satisfied. Now, the optimization problem can be represented in discrete time domain as follows:

$$\begin{aligned} (\text{P1}): & \underset{\{\mathbf{p}_{i,t}, \forall t\}}{\operatorname{argmin}} \ T_i \\ & s.t. \ ||\mathbf{p}_{i,t} - \mathbf{p}_{j,t}||_2 > r_i + r_j, \quad \forall j \neq i, \forall t && \text{(P1.a)} \\ & \quad \mathcal{S}_{r_{i,t}} \geq \mathcal{T}_s, \ \text{ if } t \mid n_t && \text{(P1.b)} \\ & \quad \mathbf{p}_{i,0} = \mathbf{p}_i^S, \mathbf{p}_{i,T_i} = \mathbf{p}_i^D, \quad \forall i && \text{(P1.c)} \\ & \quad v_{s_{i,t}} \leq v_{\max_i}, \quad \forall t && \text{(P1.d)} \\ & \quad |\phi_{i,t} - \phi_{i,t-1}| \leq \Delta t \cdot \mathcal{T}_r, \quad \forall t && \text{(P1.e)} \\ & \quad a_t \in \{1, \ldots, K\}, \quad \forall t && \text{(P1.f)} \end{aligned}$$

where the integer-valued discrete time index t indicates time increments by Δt, and $t \mid n_t$ signifies that t is divisible by n_t.

The non-communicating multi-agent navigation task can be formulated as a sequential decision making problem in a reinforcement learning framework [6]. The objective then is to develop policies, $\{\pi_i : \mathbf{s}_{i,t}^{jn} \mapsto \mathbf{v}_{i,t}, \forall i\}$ that select actions to minimize the expected time to destination while satisfying all the constraints, where $\mathbf{s}_{i,t}^{jn}$ and $\mathbf{v}_{i,t}$ are the joint state and the action of the agent, respectively. Now, the optimization problem can be reformulated as

$$\begin{aligned} (\text{P2}): & \underset{\pi_i}{\operatorname{argmin}} \ \mathbb{E}[T_i | \mathbf{s}_i^{jn}, \pi_j, \forall j \neq i] \\ & s.t. \ (\text{P1}.a) - (\text{P1}.c), (\text{P1}.f) \\ & \quad \mathbf{p}_{i,t} = \mathbf{p}_{i,t-1} + \Delta t \cdot \pi_i(\mathbf{s}_{i,t-1}^{jn}), \quad \forall t && \text{(P2.d)} \end{aligned}$$

where the expectation in the objective function in (P2) is with respect to other agents' unobservable states and policies, and (P2.d) is the agent's kinematics, which satisfy the kinematic constraints in (P1.d) and (P1.e). Further, since the agents in the considered networks have the same objective function and constraints, we use the common assumption that each agent would follow the same policy [5], [6] [41], i.e., $\pi = \pi_i$.

IV. Reinforcement Learning Based Approach

In this section, we first introduce reinforcement learning (RL) formulation for the multi-UAV navigation problem. Then, we present the approaches used to tackle the uncertainty in the UAVs' unobservable intents, and the interaction between the UAVs and the cellular network.

A. Reinforcement Learning

RL is a class of machine learning methods for solving sequential decision making problems with unknown state-transition dynamics [6] [9]. Typically, a sequential decision making problem can be formulated as an MDP, which is described by the tuple $\langle S, A, P, R, \gamma \rangle$, where S is the state space, A is action space, P is the state-transition model, R is the reward function, and γ is a discount factor.

Since the action space in this work is continuous and the set of permissible velocity vectors depends on the agent's state, we choose to optimize the value function $V_\pi(\mathbf{s}^{jn})$ as in [6], instead of optimizing the commonly used action-value function $Q(\mathbf{s}^{jn}, \mathbf{v})$ (where $\mathbf{v}$ denotes the action). The state value function of an MDP is the expected return starting from time t following policy π, i.e.,

$$V_\pi(\mathbf{s}_t^{jn}) = \sum_{t'=t}^{T} \gamma^{t'-t} R_{t'}(\mathbf{s}_{t'}^{jn}, \pi(\mathbf{s}_{t'}^{jn})). \tag{17}$$

The optimal policy is to maximize the expected return:

$$\begin{aligned} \pi^*(\mathbf{s}_t^{jn}) = & \underset{\mathbf{v}_t}{\operatorname{argmax}} R(\mathbf{s}_t^{jn}, \mathbf{v}_t) \\ & + \gamma \int_{\mathbf{s}_{t+1}^{jn}} P(\mathbf{s}_{t+1}^{jn} | \mathbf{s}_t^{jn}, \mathbf{v}_t) V^*(\mathbf{s}_{t+1}^{jn}) \mathrm{d}\mathbf{s}_{t+1}^{jn}, \end{aligned} \tag{18}$$

where $V^*(\mathbf{s}_t^{jn}) = \sum_{t'=t}^{T} \gamma^{t'-t} R_{t'}(\mathbf{s}_{t'}^{jn}, \pi^*(\mathbf{s}_{t'}^{jn}))$ is the optimal value function, $R(\mathbf{s}_t^{jn}, \mathbf{v}_t)$ is the reward received at time t, $P(\mathbf{s}_{t+1}^{jn} | \mathbf{s}_t^{jn}, \mathbf{v}_t)$ is the transition probability from time t to time $t+1$.

B. Reinforcement Learning Formulation

To estimate the high-dimensional, continuous value function, it is common to approximate it with a deep neural network (DNN) parameterized by weights and biases, $\boldsymbol{\xi}$. For notational simplicity, we drop the DNN parameters from the value function notation, i.e., $\mathcal{V}(\mathbf{s};\boldsymbol{\xi}) = \mathcal{V}(\mathbf{s})$. And $\mathbf{s}$ is the joint state of an agent which is also the input of the DNN, and $\mathcal{V}(\mathbf{s})$ is the output of the value network given $\mathbf{s}$.

By detailing each of these elements and relating to (P1.a)-(P1.c) and (P2.d), the following provides an RL formulation for the multi-UAV navigation problem. Each UAV is an independent agent, and in the discussions below, we use agent instead of UAV.

1) State Space: In multi-agent multi-GBS cellular networks, the agents are able to observe the following information from the environment: 1) its own information vector $\mathbf{s}_{i,t}$ (for the i^{th} agent at time step t); 2) the observable state of the nearest $J_n < J$ agents $\mathbf{s}_{i,t}^{jno} = [\mathbf{s}_{j,t}^{o} : j \in \{1, 2, \ldots, J_n\}]$; 3) the location information of the nearest $K_n \le K$ GBSs, which is assumed to be observed by the agents, and is denoted by $\mathbf{s}_B^o = [\mathbf{p}_{B_k} : k \in \{1, \ldots, K_n\}]$. All the information observed by the agent constitutes its joint state $\mathbf{s}_{i,t}^{jn} = [\mathbf{s}_{i,t}, \mathbf{s}_{i,t}^{jno}, \mathbf{s}_B^o], \forall t$.

2) Action Space: The action space is a set of permissible velocity vectors. Ideally, the agent can travel in any direction at any time. However, in reality the kinematic constraints in (13)-(15) restrict the agent's movement and should be taken into account. Then, based on the agent's current speed, orientation $[v_{s,i,t}, \phi_{s,i,t}]$ and the kinematic constraints, permissible actions $\mathbf{v} = [v_s, \phi]$ are sampled to build the action space $A_{i,t}$.

3) Reward Function: Similar to the formulation of the reward function defined in [42], [6], and [9], we define a sparse reward function, which awards the agent for reaching its goal, and penalizes the agent for getting too close or colliding with other agents, and also penalizes for getting close to be disconnected or already being disconnected from the cellular network. The reward function consists of four parts: the reward, R_c, that penalizes close encounters with other agents; the reward, R_s, that encourages keeping connectivity with the cellular network; the reward, R_d, that encourages arrival at the destination; and a step penalty, R_t, that encourages fast arrival. For instance, at time step t, the reward functions for the i^{th} agent can be expressed as follows:

$$
\begin{aligned}
&R_{c_{i,t}}(\mathbf{s}_{i,t}^{jn}, \mathbf{v}_{i,t}) \\
&= \begin{cases} -\alpha_1, & \text{if } d_{t_{\min}} \le r_i + r_j, \\ -\alpha_1 \times \left(1 - \dfrac{d_{t_{\min}} - r_i - r_j}{d_b}\right), & \text{if } r_i + r_j < d_{t_{\min}} \\ & \le d_b + r_i + r_j, \\ 0, & \text{otherwise}, \end{cases}
\end{aligned} \tag{19}
$$

$$
\begin{aligned}
&R_{s_{i,t}}(\mathbf{s}_{i,t}^{jn}, \mathbf{v}_{i,t}) \\
&= \begin{cases} -\alpha_2, & \text{if } t \mid n_t \text{ and } \mathcal{S}_{r_{i,t+1}} < \mathcal{T}_s, \\ -\alpha_2/2, & \text{if } t \mid n_t \text{ and } \mathcal{T}_s \le \mathcal{S}_{r_{i,t+1}} < \mathcal{T}_s + \mathcal{S}_{r_b}, \\ 0, & \text{otherwise}, \end{cases}
\end{aligned} \tag{20}
$$

$$
R_{d_{i,t}}(\mathbf{s}_{i,t}^{jn}, \mathbf{v}_{i,t}) = \begin{cases} \alpha_3, & \text{if } \mathbf{p}_{i,t+1} = \mathbf{p}_i^D, \\ 0, & \text{otherwise}, \end{cases} \tag{21}
$$

$$
R_t = -\alpha_4, \tag{22}
$$

where $d_{t_{\min}}$ is the minimum distance to other agents within the next time step duration. $\alpha_{1\sim4}$ are positive constants that can be varied to adjust the weight or emphasis of each reward term, d_b is a distance buffer between two agents, and $\mathcal{S}_{r_b}$ is an SINR buffer. Therefore, the overall reward function can be expressed as the sum

$$
\begin{aligned}
&R_{i,t}(\mathbf{s}_{i,t}^{jn}, \mathbf{v}_{i,t}) \\
&\quad = R_{c_{i,t}}(\mathbf{s}_{i,t}^{jn}, \mathbf{v}_{i,t}) + R_{s_{i,t}}(\mathbf{s}_{i,t}^{jn}, \mathbf{v}_{i,t}) + R_{d_{i,t}}(\mathbf{s}_{i,t}^{jn}, \mathbf{v}_{i,t}) + R_t.
\end{aligned} \tag{23}
$$

C. Estimation of the Agents' Unobservable Intents

The probabilistic state transition model in (18) is determined by the agents' kinematics as defined in (P2.d), other agents' hidden states, and the other agents' choices of action. Since the other agents' hidden intents are unknown, the system's state transition model is unknown as well. In addition, it is difficult to evaluate the integral, because the other agents' next state has an unknown distribution (that depends on their unobservable intents). We approximate this integral by assuming that the other agent would be traveling at a filtered velocity for a short duration Δt, which is regarded as a one-step lookahead procedure [4] [5] [6] [43]. This propagation step amounts to predicting the other agent's motion with a simple linear model, i.e., $\hat{\mathbf{v}}_{j,t} = \text{filter}(\mathbf{v}_{j,0:t})$. For the i^{th} agent, other agents' filtered velocities are included in the vector $\hat{\mathbf{v}}_{i,t}^{jno} = [\hat{\mathbf{v}}_{j,t} : j \in \{1, 2, \ldots, J_n\}]$. Then, the estimated next state of the i^{th} agent will be

$$
\hat{\mathbf{s}}_{i,t+1,\mathbf{v}}^{jn} = [f(\mathbf{s}_{i,t}, \Delta t, \mathbf{v}), f(\mathbf{s}_{i,t}^{jno}, \Delta t, \hat{\mathbf{v}}_{j,t}^{jno}), \mathbf{s}_B^o] \tag{24}
$$

where $f(\cdot)$ is the kinematic model. Then, we can select the action that has the highest value with respect to other agents' estimated state, which can be formulated as

$$
\underset{\mathbf{v} \in A_{i,t}}{\operatorname{argmax}} R_{i,t}(\mathbf{s}_{i,t}^{jn}, \mathbf{v}) + \gamma V(\hat{\mathbf{s}}_{i,t+1,\mathbf{v}}^{jn}). \tag{25}
$$

D. SINR Prediction

Model-free RL requires no prior knowledge about the environment. This usually leads to slow learning process and requires a large number of agent-environment interactions, which is typically costly or even risky to obtain [21]. Actually, each real experience obtained from the agent and cellular network interaction not only can be used to get reward and refine the value network, but also can be used for model learning in order to predict the agent's SINR experienced at certain positions. More specifically, when flying in the environment, agents interact with the cellular network and obtain the empirical SINR $\widehat{\mathcal{S}}_r$. Since there is no need to use the exact SINR for connectivity measurement, this work uses the quantized SINR level, $L_w(\widehat{\mathcal{S}}_r)$, to check the agent's connectivity. With a finite set of measurements $\{\langle \mathbf{s}_B^{jn}, L_w(\mathbf{s}_B^{jn})\rangle\}$, where $\mathbf{s}_B^{jn} = [\mathbf{p}, \mathbf{s}_B^o]$, a DNN can be trained to predict the SINR level $L_w(\mathbf{s}_B^{jn})$.

A fully connected DNN with parameters $\boldsymbol{\xi}_w$ can be used to predict the agent's SINR level, i.e., $\boldsymbol{\xi}_w$ is trained so that $L_w(\mathbf{s}_B^{jn}) \approx \mathcal{L}_w(\mathbf{s}_B^{jn}; \boldsymbol{\xi}_w)$. The data measurement $\langle \mathbf{s}_B^{jn}, L_w(\mathbf{s}_B^{jn})\rangle$ only arrives incrementally as the agent flies to new locations and can be saved in a database (e.g., replay memory), and a minibatch is sampled at random from the database to update the network parameter $\boldsymbol{\xi}_w$. Note that the prediction of SINR levels might be highly inaccurate initially, but can be continuously improved as more real experience is accumulated.

V. Decentralized Deep Reinforcement Learning Algorithm

In this section, we present the proposed decentralized deep reinforcement learning algorithm as a solution to multi-UAV navigation with collision avoidance and wireless connectivity constraints, including the SINR-prediction neural network. The proposed algorithm is presented in Algorithm 1, and is referred to as RLTCW-SP (RL for Trajectory optimization with Collision avoidance and Wireless connectivity constraint and with SINR Prediction).

A. Parametrization

Since the optimal policy should be invariant to any coordinate plane, we follow the agent-centric parameterization as in [6], [42] and [9], where the agent is located at the origin and the x-axis is pointing toward the agent's destination. The states of the i^{th} agent after transformation is

$$\widetilde{\mathbf{s}}_i = [d_{g_i}, v_{\max_i}, \tilde{v}_{x_i}, \tilde{v}_{y_i}, r_i, \tilde{\phi}_i] \tag{26}$$

$$\widetilde{\mathbf{s}}_i^{jno} = [[\tilde{p}_{x_j}, \tilde{p}_{y_j}, H_V, \tilde{v}_{x_j}, \tilde{v}_{y_j}, r_j, d_j] : j \in \{1, 2, \ldots, J_n\}] \tag{27}$$

where d_g is the agent's distance to the goal, d_j is the agent's distance to the j^{th} agent, and $\tilde{p}$ and $\tilde{v}$ denote p and v in the new coordinate, respectively.

In addition, SINR experienced at an agent depends on the distance and the relative angular direction from the agent to the GBSs, while it does not depend on the positions in global coordinates. To remove this redundant dependence, the location information vector of all GBSs can be parameterized as

$$\widetilde{\mathbf{p}}_{B_k} = [\tilde{p}_{x_{B_k}}, \tilde{p}_{y_{B_k}}, d_{B_k}, \phi_{B_k}, \theta_{B_k}] \tag{28}$$

$$\widetilde{\mathbf{s}}_{B_i} = [\widetilde{\mathbf{p}}_{B_k} : k \in \{1, \ldots, K_n\}] \tag{29}$$

where $d_{B_k} = ||\mathbf{p}_{B_k} - \mathbf{p}_i||$ is the distance from the agent to the k^{th} BS, ϕ_{B_k} and θ_{B_k} are the horizontal and vertical angles of the k^{th} BS with respect to the agent.

Therefore, the joint state of the i^{th} agent after transformation is

$$\widetilde{\mathbf{s}}_i^{jn} = [\widetilde{\mathbf{s}}_i, \widetilde{\mathbf{s}}_i^{jno}, \widetilde{\mathbf{s}}_{B_i}]. \tag{30}$$

Algorithm 1: RLTCW-SP Algorithm

```
Input: State-value pairs $D$
Initialize state-value pairs $D$
Initialize location-SINR pairs $D_w$
Initialize value network $\boldsymbol{\xi}$ with $D$
Initialize SINR-prediction network $\boldsymbol{\xi}_w$
for episode = 0: total episode do
    for n random training cases do
        Initialize $\mathbf{s}_{i,0} \forall i$
        while not all reached destinations do
            for each agent $i$ do
                if not reached destination then
                    $\mathbf{s}_{i,t}^{jn} \leftarrow$ observeEnvironment()
                    $A_{i,t} \leftarrow$ sampleActionSpace()
                    $c \leftarrow$ randomSample(Uniform (0,1))
                    if $c \leq \epsilon$ then
                        $\mathbf{v}_{i,t} \leftarrow$ randomSample($A_{i,t}$)
                    else
                        $\hat{\mathbf{v}}_{i,t}^{jno} \leftarrow$ filter($\mathbf{v}_{0:t-1}^{jn}$)
                        $\hat{\mathbf{s}}_{i,t+1}^{jno} \leftarrow$ propagate($\mathbf{s}_{i,t}^{jno}, \hat{\mathbf{v}}_{i,t}^{jno}$)
                        for every $\boldsymbol{a}$ in $A_{i,t}$ do
                            $\hat{\mathbf{s}}_{i,t+1} \leftarrow$ propagate($\mathbf{s}_{i,t}, \mathbf{a}$)
                            $\hat{L}_{w_{i,t+1}} = \mathcal{L}_w(\hat{\mathbf{s}}_{B_{i,t+1}}^{jn})$
                            $R_{i,t} \leftarrow$ getReward($\hat{\mathbf{s}}_{i,t+1}^{jn}, \hat{L}_{w_{i,t+1}}$)
                            $V_p = R_{i,t} + \gamma \mathcal{V}(\hat{\mathbf{s}}_{i,t+1}^{jn})$
                        $\mathbf{v}_{i,t} \leftarrow \text{argmax}_{\mathbf{a} \in A_{i,t}} V_p$
                    $R_{i,t}, \mathbf{s}_{i,t+1}, \mathcal{S}_{r_{i,t+1}} \leftarrow$ executeAction($\mathbf{v}_{i,t}$)
        for each agent $i$ do
            $V_{i,0:T_i} \leftarrow$ updateValue($\mathbf{s}_{i,0:T_i}^{jn}, R_{i,0:T_i}, \boldsymbol{\xi}$)
            $L_{w_{i,0:T_i}} \leftarrow$ getSINRlevel($\mathcal{S}_{r_{i,0:T_i}}$)
            Update state-value pairs $D$ with $\langle \mathbf{s}_{i,0:T_i}^{jn}, V_{i,0:T_i} \rangle$
            Update location-SINR pairs $D_w$ with
              $\langle \mathbf{s}_{B_{i,0:T_i}}^{jn}, L_{w_{i,0:T_i}} \rangle$
    Sample random minibatch from $D$, and update value
      network $\boldsymbol{\xi}$ by gradient descent.
    Sample random minibatch from $D_w$, and update
      SINR-prediction network $\boldsymbol{\xi}_w$ by gradient descent.
return $\boldsymbol{\xi}, \boldsymbol{\xi}_w$
```

And the input of the SINR-prediction network becomes

$$\widetilde{\mathbf{s}}_{B_i}^{jn} = [[d_{B_k}, \phi_{B_k}, \theta_{B_k}] : k \in \{1, \ldots, K_n\}]. \tag{31}$$

B. Initialization

The value network $\boldsymbol{\xi}$ can be first initialized with imitation learning using a set of experiences to accelerate the convergence. More specifically, in this work, we use optimal reciprocal collision avoidance (ORCA) [5] to generate a number of trajectories that contain a large set of state-value pairs $\{\langle \mathbf{s}^{jn}, V\rangle\}^{N_1}$, where $V = \gamma^{t_g}$ and t_g is the time to reach the destination. The experiences are saved in memory D

(line 1 in Algorithm 1). Then, the value network is initialized by supervised training on D (line 3). The value network is trained by back-propagation to minimize a quadratic regression error

$$\boldsymbol{\xi} = \underset{\boldsymbol{\xi}'}{\operatorname{argmin}} \sum_{k=1}^{N_1} \left(V_k - \mathcal{V}(\mathbf{s}_k^{jn}; \boldsymbol{\xi}') \right)^2 . \tag{32}$$

If a set of location-SINR experiences can be downloaded from the cloud, we can save the downloaded dataset in memory D_w (line 2), $\{\langle \mathbf{s}_B^{jn}, L_w \rangle\}^{N_2}$, where L_w is the scaled SINR level that the agent experienced. Then, the SINR-prediction network can be initialized with $\boldsymbol{\xi}_w = \operatorname{argmin}_{\boldsymbol{\xi}_w'} \sum_{k=1}^{N_2} \left(L_{w_k} - \mathcal{L}(\mathbf{s}_B^{jn}; \boldsymbol{\xi}_w') \right)$, which is trained by back-propagation (line 4). If no dataset is available, D_w is initialized with an empty list, and the SINR-prediction network is initialized with random network parameters.

C. Refining Process

After initialization, a refining process is performed using RL. Particularly, a set of random training cases is generated in each episode (line 6). In each training case, each agent navigates around others to arrive at its destination, while interacting with the cellular network (line 10- line 25). It is worth noting that the agents navigate simultaneously and with no communication among each other. At each time step t, each agent first observes the environment, obtains the observable states of other nearby agents and the location information of the GBSs, and then obtains its joint state $\mathbf{s}_t^{jn}$ (line11). Then, based on its current velocity and kinematic constraints, each agent builds an action space A_t (line 12). Using an ϵ-greedy policy, each agent selects a random action with probability ϵ from A_t (line 15), or follows the value network greedily otherwise (lines 17-24). When following the value network to choose actions, each agent performs the following: 1) estimate other nearby agents' motion by filtering their velocities, and estimate their observable states $\hat{\mathbf{s}}_{t+1}^{jno}$ following equation (24) (lines 17-18); 2) predict its next SINR level $\mathcal{L}_{w_{t+1}}$ using the SINR-prediction network $\boldsymbol{\xi}_w$ (line 21); and 3) choose the best action in A_t which has the maximum V_p (line 22-24). Then, each agent executes the chosen action $\mathbf{v}_t$ and moves to the next position, while obtaining reward R_t from the environment (line 25).

When all agents have arrived their destinations in each training case, trajectories $\mathbf{s}_{i,0:T_1} \forall i$ are then processed to generate a set of state-value pairs $\langle \mathbf{s}_{i,0:T_i}^{jn}, V_{i,0:T_i} \rangle$, where

$$V_{i,t} = \begin{cases} R_{i,t} + \gamma \mathcal{V}(\mathbf{s}_{i,1:t+1}^{jn}) & \text{if } t+1 < T_i, \\ R_{i,t} & \text{if } t+1 = T_i, \end{cases}$$

and a set of location-SINR pairs $\langle [\mathbf{p}_{i,0:T_i}, \mathbf{s}_B^o], L_{w_{i,0:T_i}} \rangle$. The new pairs are used to update D and D_w.

D. Training

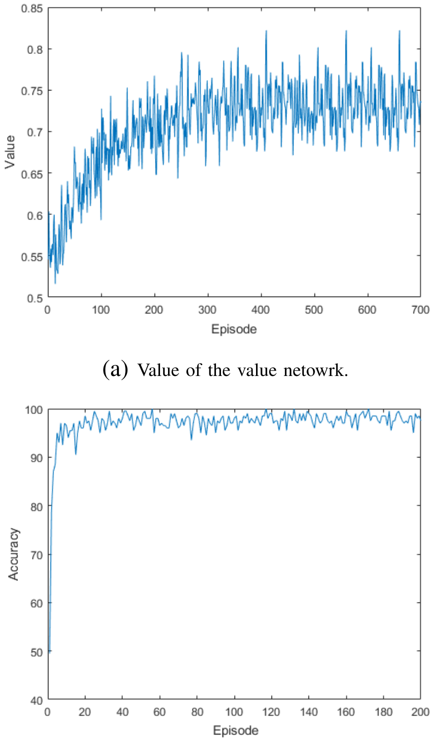
To train the value network and SINR-prediction network, a set of training points is randomly sampled from the experience set, which contains state-value pairs for $\boldsymbol{\xi}$ or location-SINR pairs for $\boldsymbol{\xi}_w$ from many different trajectories. Then, the networks are finally updated by stochastic gradient descent (back-propagation) on the sampled subsets of experience.

E. Real-Time Navigation

With the trained value network and SINR-prediction network, agent can execute real-time navigation. This process is provided in Algorithm 2.

Algorithm 2: Real-Time Navigation

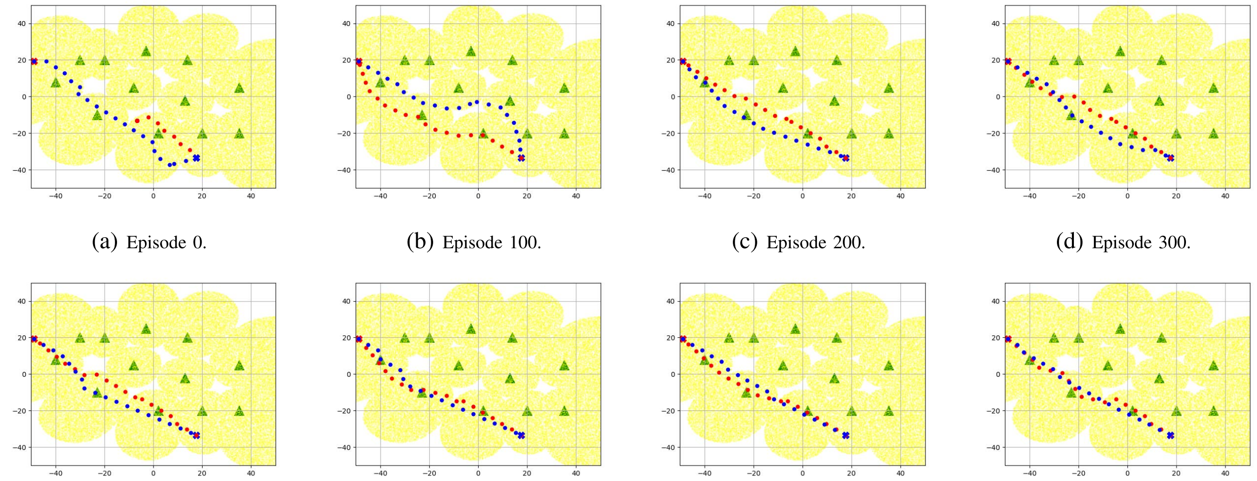
```
Input: $\boldsymbol{\xi}, \boldsymbol{\xi}_w$
Initialize $\mathbf{s}_0$
while not reached destination do
    $\mathbf{s}_t^{jn} \leftarrow$ observeEnvironment()
    $A_t \leftarrow$ sampleActionSpace()
    $\hat{\mathbf{v}}_t^{jn} \leftarrow$ filter($\mathbf{v}_{0:t-1}^{jn}$)
    $\hat{\mathbf{s}}_{t+1}^{jno} \leftarrow$ propagate($\mathbf{s}_t^{jno}, \hat{\mathbf{v}}_t^{jn}$)
    for every $\mathbf{a}$ in $A_t$ do
        $\hat{\mathbf{s}}_{t+1} \leftarrow$ propagate($\mathbf{s}_t, \mathbf{a}$)
        $\hat{L}_{w_{t+1}} = \mathcal{L}_w([\hat{\mathbf{p}}_{t+1}, \mathbf{s}_B])$
        $R_t \leftarrow$ getReward($\hat{\mathbf{s}}_{t+1}, \hat{\mathbf{s}}_{t+1}^{jno}, \hat{L}_{w_{t+1}}$)
        $V_p = R_t + \gamma \mathcal{V}(\hat{\mathbf{s}}_{t+1}^{jn})$
    $\mathbf{v}_t \leftarrow \operatorname{argmax}_{\mathbf{a} \in A_t} V_p$
    $\mathbf{s}_{t+1} \leftarrow$ executeAction($\mathbf{v}_t$)
return $\mathbf{v}_{0:T-1}$, $\mathbf{s}_{0:T}$
```



(a) Value of the value netowrk.

(b) Accuracy of the SINR-prediction network.

Fig. 3. Value of the value network and accuracy of the SINR-prediction network as functions of the number of episodes.



(a) Episode 0.

(b) Episode 100.

(c) Episode 200.

(d) Episode 300.

(e) Episode 400.

(f) Episode 500.

(g) Episode 600.

(h) Episode 700.

Fig. 4. Trajectory examples at different episodes during training.

VI. Numerical Results

In this section, we present the numerical results to evaluate the performance of the proposed algorithms. In the illustrations of environment and trajectories in this section, the GBSs are marked by green triangles, and the yellow areas indicate the communication coverage zones where the UAVs are able to connect with the cellular network (i.e., $\mathcal{S}_r \geq \mathcal{T}_s$). UAVs' trajectories are displayed as a list of dots in different colors, and the destinations are marked with crosses. In each flight trajectory, there are four possible outcomes for the agent/UAV: 1) success, if the UAV arrives at its destination successfully; 2) collision, if it collides with others; 3) disconnection, if the continuous disconnected time is larger than the threshold $\mathcal{T}_t$; 4) stuck, if the UAV freezes and stops moving and consequently does not reach the destination. In addition, we also compute the additional average time (referred to also as average more time) needed to reach the destination, when compared with the lower bound (attained when the UAV goes straight towards the destination at the maximum speed). Therefore, we use success rate (SR), collision rate (CR), disconnection rate (DR) and average more time (AMT) to show the performance of the algorithms.

A. Environment Setting and the Networks

Since the UAVs fly at the same altitude, the area of interest becomes two-dimensional. In the simulations, we consider an area with 12 GBSs deployed. The GBSs transmit with power $P_B = 1$ dBW, and have a height of $H_B = 32$m. The antenna patterns are set with $\theta^{tilt} = 10^\circ$ and $\theta^{3dB} = 15^\circ$. The UAVs are assumed to fly at a fixed altitude of $H_V = 50$ m. The noise power is $\mathcal{N}_s = 10^{-6}$, and the SINR threshold is $\mathcal{T}_s = -3$ dB. Each UAV, as an independent agent, is able to observe the nearest 8 GBSs' locations and at most 2 other UAVs' observable states.

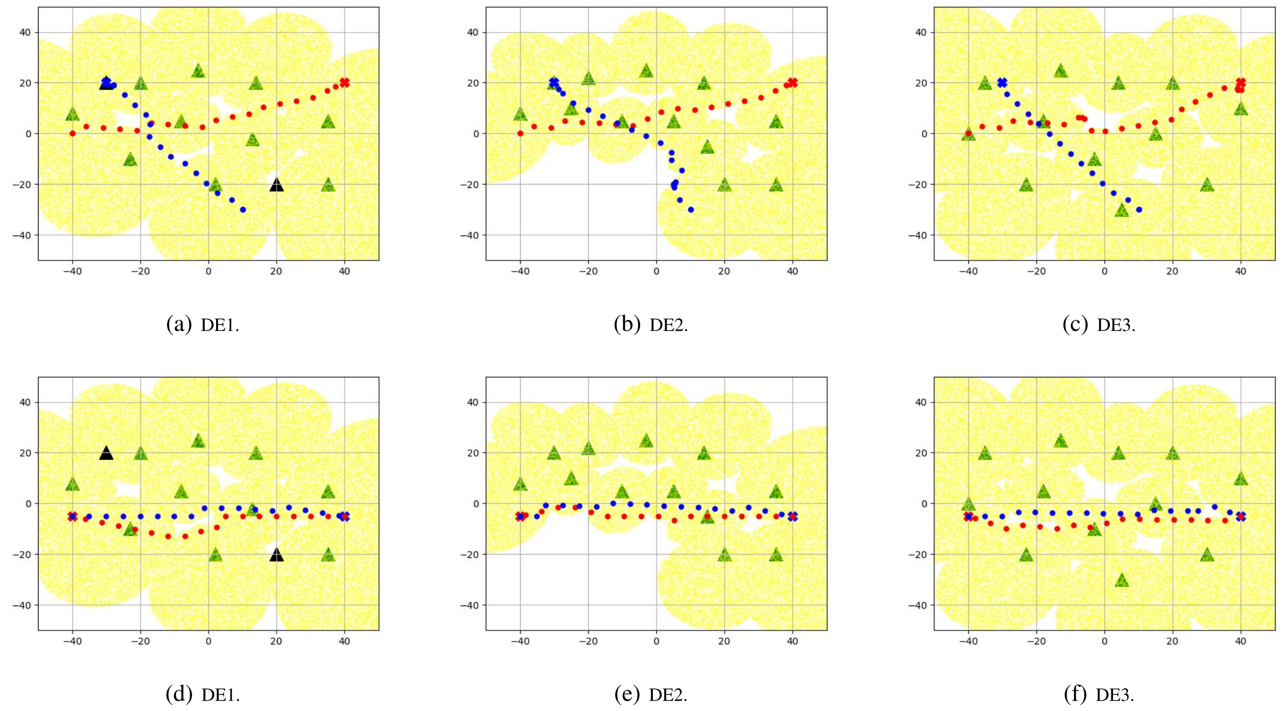
We construct the value network via a three-layered DNN of size (64,32,16). The exploration parameter ϵ linearly decays from 0.5 to 0.1. The replay memory capacity is 30000 for the 2-UAV scenario and 100000 for scenarios with more than two UAVs. The SINR-prediction network is constructed via a three-layered DNN of size (32,16,8). A standardization layer is utilized after the input layer of both networks. ReLU activation function is used for the input layer and two hidden layers for both networks. Both networks use Adam optimizer, and have learning rate 0.01, batch size 200, and a regularization parameter 0.0001.

To build the action spaces $A_{i,t}$, based on the UAV's current velocity $[v_{s,i,t}, \phi_{s,i,t}]$, 22 velocities are chosen, including: 1) combinations $[v_s, \phi_{s,i,t} + \phi]$, where $v_s \in \{0, \frac{1}{2}v_{\max}, v_{\max}\}$ and $\phi \in \{\pm\mathcal{T}_r, \pm\frac{2}{3}\mathcal{T}_r, \pm\frac{1}{3}\mathcal{T}_r, 0\}$; and 2) current velocity. The values for $\alpha_{1\sim4}$, d_b and $\mathcal{S}_b$ in reward function are selected as follows: $\alpha_1 = 1, \alpha_2 = 1, \alpha_3 = 2, \alpha_4 = 0.1, d_b = 0.2$ and $\mathcal{S}_{r_b} = 0.1$.

B. Convergence in Training

Fig. 3 shows the value of the value network and accuracy of the SINR-prediction network as functions of the number of episodes during training for a 2-UAV scenario. Fig. 3(a) shows that the value converges after around 200 episodes. From Fig. 3(b), we can see that the accuracy converges after around 20 episodes, since in each episode 50 random trajectories are generated for each UAV, during which more than 15000 location-SINR pairs are collected and used to train the SINR-prediction network.

The trajectory optimization process for two UAVs is displayed in Fig. 4. At episode 0, the SINR-prediction network is initialized with random weights and bias, and is not able to predict the accurate SINR level. Besides, the policy has



(a) DE1.

(b) DE2.

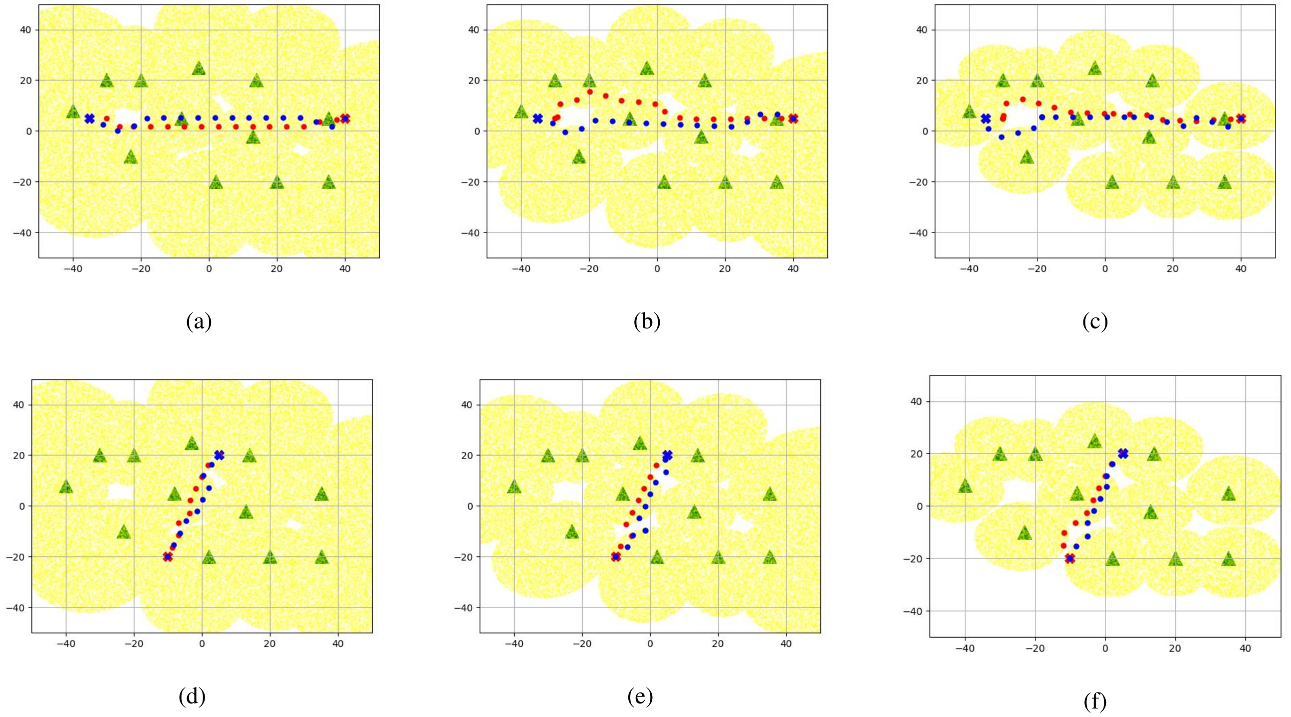
(c) DE3.

(d) DE1.

(e) DE2.

(f) DE3.

Fig. 5. Illustrations of different environments used in navigation testing, and trajectory examples when using proposed RLTCW-SP algorithm. In (a) and (d), the two BSs in black are not operational for the UAVs.



(a)

(b)

(c)

(d)

(e)

(f)

Fig. 6. Trajectory examples in environments with different settings, i.e., in (a) and (d), $H_V = 50$ m, $\theta^{tilt} = 10^\circ$ and $\theta^{3dB} = 15^\circ$; in (b) and (e), $H_V = 100$ m, $\theta^{tilt} = 10^\circ$ and $\theta^{3dB} = 15^\circ$; and in (c) and (f), $H_V = 50$ m, $\theta^{tilt} = 15^\circ$ and $\theta^{3dB} = 35^\circ$.

not been refined by RL. As a result, the two UAVs are easily getting disconnected or stuck. After 100 episodes of training, the SINR-prediction network is well-trained and able to predict the SINR levels with 97% accuracy. Also, the value network is trained with refined state-value pairs. Thus, the UAVs can reach their destinations, but with long trajectories to avoid collisions and disconnection. As the training proceeds, the policy improves, leading to shorter expected trajectories.

TABLE I

PERFORMANCE OF DIFFERENT ALGORITHMS IN DIFFERENT ENVIRONMENTS IN TERMS OF SUCCESS RATE (SR), COLLISION RATE (CR), AND DISCONNECTION RATE (DR) (**ALL RATES ARE IN %**)

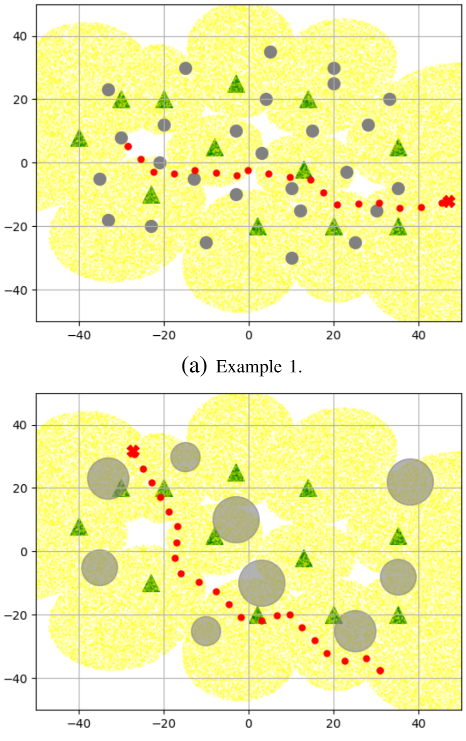
	Same Environment			DE1			DE2			DE3		
	SR	CR	DR	SR	CR	DR	SR	CR	DR	SR	CR	DR
RLTCW-AW	99.15	0.85	0	98.95	0.95	0	98.25	1.42	0.17	99.1	0.9	0
RLTCW-SP	99.1	0.85	0.05	99.05	0.85	0.07	98.08	1.5	0.42	99	0.92	0.08
RLTCW	64.8	0.6	34.6	74.85	0.15	25	60.7	0.8	38.5	69	0.33	30.67

After 700 episodes of training, the AMT (for the successful trajectories) is 0.2662s. Separately, we also compute the AMT in two different scenarios for comparison: 1) the connectivity constraint is not considered for the two-UAV trajectory design (CADRL [6]); and 2) the collision avoidance constraint is not considered if there is only one UAV. The AMT in these two scenarios are 0.195s, and 0.204s, respectively. In addition, after 700 episodes of training, the proposed algorithm can achieve a success rate (SR) of 99%, while the SR is 3.9% with the random selection method.

C. Testing of Navigation in Different Environments

In the proposed RLTCW-SP algorithm, an SINR-prediction network is trained to predict the SINR level. In an ideal scenario, the antenna pattern information of GBSs may be available to the UAVs, and then the UAVs are able to predict the SINR with that information. In this subsection, we compare the performances of the following three algorithms: 1) the UAVs are able to get the antenna pattern information of the GBSs, and then predict the SINR directly (referred to as RLTCW-AW); 2) the proposed RLTCW-SP algorithm, which uses the location-SINR memory, and trains an SINR-prediction network to predict the SINR level; 3) the UAVs do not predict the SINR and only use the value network to make decisions (referred to as RLTCW). The navigation test is done in three types of environments: 1) the same environment as in the training; 2) the same environment but two BSs are not operational for the UAV (due to congestion, malfunction, resource allocation to ground UEs, or GBS activation schedule), an illustration of which is presented in Fig. 5(a) and (d); 3) different environments with different GBS deployments, illustrations of which are presented in Figs. 5(b), (c), (e) and (f). Environments displayed in Figs. 5 (a) (b) and (c) are referred to as DE1, DE2 and DE3, respectively (using DE as the abbreviation for different environment). Fig. 5 also presents examples of trajectories that the UAVs perform using the proposed RLTCW-SP algorithm, and Fig. 5 (d) (e) and (f) present a challenging scenario in which the destination of one UAV is the starting point of the other UAV.

The performance of the three algorithms in different environments are presented in Table I. As expected, the RLTCW-AW with the perfect knowledge of antenna patterns has the best performance, and the RLTCW-SP algorithm has slightly lower performance which is due to the potential inaccuracies in the SINR prediction, while the performance of RLTCW is substantially lower compared to the other two, due to very high DR (disconnection rate). In addition, the SR (success rate) performance of the proposed RLTCW-SP algorithm decreases only slightly in different environments, and how large the decrease is depends on which environment is used in testing. When there are large and wide out-of-coverage zones in the environment (as shown in Fig. 5(b)), the SR performance decreases relatively a bit more. The reason is that the wide out-of-coverage zones are more likely to make the UAV get stuck at the edge and not be able to decide which direction to go. Overall, in the three different environments in testing, the proposed RLTCW-SP can achieve above 98% of SR in 2-UAV scenarios.



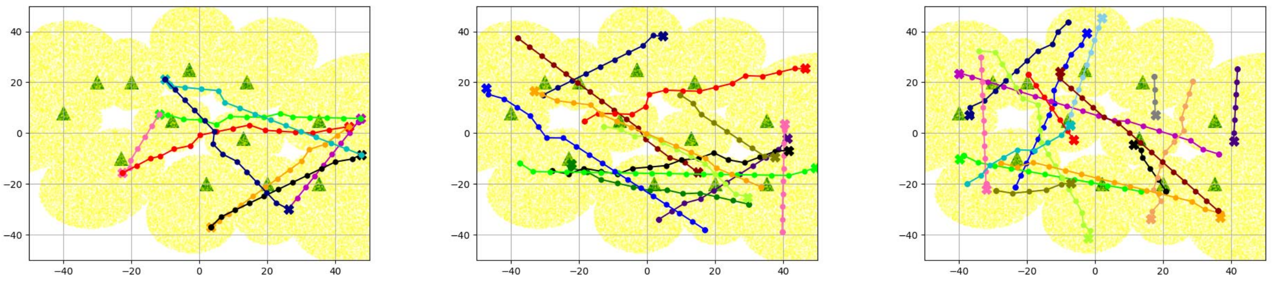
(a) Example 1.

(b) Example 2.

Fig. 7. Trajectory examples in environments with obstacles/no-fly zones.

D. Navigation in Different Settings

In this subsection, we present simulation results on the trajectories when the GBSs have different antenna patterns and when the UAVs fly at different heights. The SINR threshold is $\mathcal{T}_s = -4$ dB in this subsection. In Figs. 6 (a) and (d), we provide two different trajectory examples when we have $H_V = 50$ m, $\theta^{tilt} = 10°$ and $\theta^{3dB} = 15°$. In Figs. 6 (b) and (e),



(a) 8-UAV scenario

(b) 12-UAV scenario

(c) 16-UAV scenario

Fig. 8. Trajectory examples for multi-UAV navigation.

TABLE II
PERFORMANCE FOR MULTI-UAV NAVIGATION

Number of UAVs		2	4	6	8	10	12	14	16	18	20
d_b=0.2	SR(%)	99.1	98.65	97.37	97	96.9	95.55	94.83	93.9	91.9	91.74
	CR(%)	**0.85**	**1.32**	**2.18**	**2.73**	**3.68**	**4.38**	**5.12**	**6.31**	**7.06**	**8.18**
	DR(%)	0.05	0.03	0.1	0.06	0.05	0.07	0.05	0.1	0.04	0.08
	AMT(s)	0.266	0.306	0.319	0.322	0.336	0.346	0.364	0.369	0.381	0.441
d_b=1	SR(%)	99.33	98.92	98.58	97.67	97.53	97.28	97	96.19	95.32	95
	CR(%)	**0.56**	**1**	**1.25**	**2.17**	**2.13**	**2.67**	**2.5**	**3.62**	**4.37**	**4.7**
	DR(%)	0.11	0.08	0.17	0.17	0.33	0.06	0.5	0.19	0.11	0.3
	AMT(s)	0.299	0.359	0.365	0.406	0.470	0.481	0.510	0.525	0.572	0.675

UAV altitudes are increased to $H_V = 100$ m, and we notice that due to larger path loss and smaller antenna gains, coverage zones shrink, which in turn potentially increases the length of the trajectories. In Figs. 6 (c) and (f), GBSs have larger downtilting angle and 3dB beamwith of the main lobe. In this case, the UAVs experience smaller received power from the main link and potentially larger interference, leading to substantially smaller SINR levels. Therefore, the coverage zones in Figs. 6 (c) and (f) are smaller than those in Figs. 6 (a) and (d) and even Figs. 6 (b) and (e). In all cases, we note that UAVs successfully find different trajectories to meet the connectivity requirements and adapt to different coverage zones.

E. Navigation in Environments With Obstacles/No-Fly Zones

The proposed RLTCW-SP algorithm can also be used for navigation in environment with obstacles that are regarded as non-moving agents. For instance, the trained networks for the 2-UAV scenario can be used for one UAV navigation in an environment with obstacles or no-fly zones. More specifically, the UAV can observe the nearest obstacle, and takes the obstacle's location in the joint state for choosing actions. Fig. 7 displays two illustrations. In this setting, obstacles can be considered as actual obstacles (e.g., tall buildings or structures) or no-fly zones modeled for the UAVs.

F. Navigation With More Than Two UAVs

Fig. 8 provides the illustrations for 8-UAV, 12-UAV, 16-UAV navigation scenarios, respectively, where dotted lines are used to display the trajectories. The SR, CR, DR and AMT performances in scenarios with 2 to 20 UAVs are presented in Table II, when the distance buffer is $d_b = 0.2$ (default) or 1. We note that the CR increases when more UAVs are in the environment. As mentioned before, the UAVs can observe a maximum of 2 nearest UAVs in the environment. Therefore, for more crowded scenarios, several UAVs are non-observable and as a result the CR can increase when compared with scenarios involving smaller number of UAVs. Additionally, when there are more UAVs in the same area, the interactions become more complex and challenging for the algorithm to handle. However, we note that the performance regarding the SR is still above 90% for the 20-UAV scenario. Table II further shows that the UAVs need more time to reach their destinations when there are more UAVs in the environment. Moreover, when we compare the performances between the scenario with $d_b = 0.2$ and the scenario with $d_b = 1$, we have the following observations for larger d_b: 1) the CR is reduced, since larger d_b encourages the UAVs to stay relatively farther away from each other and therefore leads to lower collision risk; and 2) the AMT is increased, since staying further away from each other generally leads to longer trajectories and correspondingly longer mission completion time. In addition, the CR is less than 5% for the 20-UAV scenario when we set $d_b = 1$.

Fig. 9 compares the collision rates between the proposed algorithm (RLTCW-SP) and the trajectory optimization algorithm with connectivity constraint only, i.e., collision avoidance is not considered. It can be observed from the figure that as the number of UAVs in the environment grows, CR increases accordingly. If collision avoidance is not considered in the algorithm, CR is not only much higher (e.g., three to six times higher) but also increases much faster than the proposed algorithm, which takes into account the collision

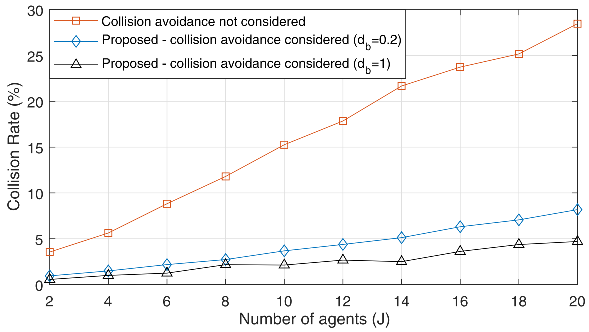


Fig. 9. Collision rate comparison between two methods: 1) the proposed approach, in which the collision avoidance is taken into account in training; 2) the approach in which collision with other UAVs is not considered in training.

avoidance. Therefore, collision avoidance is critical in multi-UAV scenarios, especially in crowded environments, and the proposed algorithm can significantly decrease the collision risk and hence achieve much lower collision rates.

VII. Conclusion

In this work, we have studied multi-UAV trajectory optimization with collision avoidance and wireless connectivity constraints. In establishing the wireless connectivity, we have taken into account the antenna radiation patterns, path loss, and SINR levels. We have formulated trajectory optimization as a sequential decision making problem and proposed a decentralized deep reinforcement learning algorithm. In particular, a value neural network has been developed to obtain the values from the agent/UAV's joint states. An SINR-prediction neural network has been designed, using accumulated SINR measurements obtained when interacting with the cellular network, to map the GBS locations into the SINR levels in order to predict the UAV's SINR levels. We have investigated the performance in terms of success rate, collision rate, disconnection rate, and average more time. In the numerical results, we have considered various scenarios (e.g., with different GBS deployments, different UAV heights, different antenna patterns, and obstacles/no-fly zones) and we have shown that with the value network and SINR-prediction network, real-time navigation for multi-UAVs can be efficiently performed in different environments with high success rates.

References

[1] X. Wang and M. C. Gursoy, "Learning-based UAV trajectory optimization with collision avoidance and connectivity constraints," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2021, pp. 1–6.

[2] Y. Zeng, Q. Wu, and R. Zhang, "Accessing from the sky: A tutorial on UAV communications for 5G and beyond," *Proc. IEEE*, vol. 107, no. 12, pp. 2327–2375, Dec. 2019.

[3] M. M. Azari, F. Rosas, and S. Pollin, "Cellular connectivity for UAVs: Network modeling, performance analysis, and design guidelines," *IEEE Trans. Wireless Commun.*, vol. 18, no. 7, pp. 3366–3381, Jul. 2019.

[4] J. van den Berg, M. Lin, and D. Manocha, "Reciprocal velocity obstacles for real-time multi-agent navigation," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2008, pp. 1928–1935.

[5] J. Van Den Berg, S. J. Guy, M. Lin, and D. Manocha, "Reciprocal n-body collision avoidance," in *Robotics Research*. Berlin, Germany: Springer, 2011, pp. 3–19.

[6] Y. F. Chen, M. Liu, M. Everett, and J. P. How, "Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 285–292.

[7] Y. F. Chen, M. Everett, M. Liu, and J. P. How, "Socially aware motion planning with deep reinforcement learning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 1343–1350.

[8] M. Everett, Y. F. Chen, and J. P. How, "Motion planning among dynamic, decision-making agents with deep reinforcement learning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 3052–3059.

[9] M. Everett, Y. F. Chen, and J. P. How, "Collision avoidance in pedestrian-rich environments with deep reinforcement learning," *IEEE Access*, vol. 9, pp. 10357–10377, 2021.

[10] S. Zhang and R. Zhang, "Trajectory optimization for cellular-connected UAV under outage duration constraint," *J. Commun. Inf. Netw.*, vol. 4, no. 4, pp. 55–71, Dec. 2019.

[11] Y. Zeng, X. Xu, and R. Zhang, "Trajectory design for completion time minimization in UAV-enabled multicasting," *IEEE Trans. Wireless Commun.*, vol. 17, no. 4, pp. 2233–2246, Apr. 2018.

[12] Y. Zeng and R. Zhang, "Energy-efficient UAV communication with trajectory optimization," *IEEE Trans. Wireless Commun.*, vol. 16, no. 6, pp. 3747–3760, Jun. 2017.

[13] S. De Bast, E. Vinogradov, and S. Pollin, "Cellular coverage-aware path planning for UAVs," in *Proc. IEEE 20th Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Jul. 2019, pp. 1–5.

[14] B. Khamidehi and E. S. Sousa, "Power efficient trajectory optimization for the cellular-connected aerial vehicles," in *Proc. IEEE 30th Annu. Int. Symp. Pers., Indoor Mobile Radio Commun. (PIMRC)*, Sep. 2019, pp. 1–6.

[15] H. Yang, J. Zhang, S. H. Song, and K. B. Lataief, "Connectivity-aware UAV path planning with aerial coverage maps," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2019, pp. 1–6.

[16] S. Zhang and R. Zhang, "Radio map-based 3D path planning for cellular-connected UAV," *IEEE Trans. Wireless Commun.*, vol. 20, no. 3, pp. 1975–1989, Mar. 2021.

[17] X. Mu, Y. Liu, L. Guo, and J. Lin, "Non-orthogonal multiple access for air-to-ground communication," *IEEE Trans. Commun.*, vol. 68, no. 5, pp. 2934–2949, May 2020.

[18] S. Zhang, Y. Zeng, and R. Zhang, "Cellular-enabled UAV communication: A connectivity-constrained trajectory optimization perspective," *IEEE Trans. Commun.*, vol. 67, no. 3, pp. 2580–2604, Mar. 2019.

[19] E. Bulut and I. Guevenc, "Trajectory optimization for cellular-connected UAVs with disconnectivity constraint," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, May 2018, pp. 1–6.

[20] B. Khamidehi and E. S. Sousa, "A double Q-learning approach for navigation of aerial vehicles with connectivity constraint," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2020, pp. 1–6.

[21] Y. Zeng, X. Xu, S. Jin, and R. Zhang, "Simultaneous navigation and radio mapping for cellular-connected UAV with deep reinforcement learning," *IEEE Trans. Wireless Commun.*, vol. 20, no. 7, pp. 4205–4220, Jul. 2021.

[22] U. Challita, W. Saad, and C. Bettstetter, "Interference management for cellular-connected UAVs: A deep reinforcement learning approach," *IEEE Trans. Wireless Commun.*, vol. 18, no. 4, pp. 2125–2140, Apr. 2019.

[23] S. Khairy, P. Balaprakash, L. X. Cai, and Y. Cheng, "Constrained deep reinforcement learning for energy sustainable multi-UAV based random access IoT networks with NOMA," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 4, pp. 1101–1115, Apr. 2021.

[24] R. Zhang, M. Wang, L. X. Cai, and X. Shen, "Learning to be proactive: Self-regulation of UAV based networks with UAV and user dynamics," *IEEE Trans. Wireless Commun.*, vol. 20, no. 7, pp. 4406–4419, Jul. 2021.

[25] X. Liu, Y. Liu, Y. Chen, and L. Hanzo, "Trajectory design and power control for multi-UAV assisted wireless networks: A machine learning approach," *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 7957–7969, Aug. 2019.

[26] R. W. Beard and T. W. McLain, "Multiple UAV cooperative search under collision avoidance and limited range communication constraints," in *Proc. 42nd IEEE Int. Conf. Decis. Control*, vol. 1, 2003, pp. 25–30.

[27] D. Bauso, L. Giarre, and R. Pesenti, "Multiple UAV cooperative path planning via neuro-dynamic programming," in *Proc. 43rd IEEE Conf. Decis. Control (CDC)*, vol. 1, Dec. 2004, pp. 1087–1092.

[28] B. D. Song, J. Kim, and J. R. Morrison, "Rolling horizon path planning of an autonomous system of UAVs for persistent cooperative service: MILP formulation and efficient heuristics," *J. Intell. Robotic Syst.*, vol. 84, nos. 1–4, pp. 241–258, Dec. 2016.

[29] T. B. Wolf and M. J. Kochenderfer, "Aircraft collision avoidance using Monte Carlo real-time belief space search," *J. Intell. Robot. Syst.*, vol. 64, no. 2, pp. 277–298, Nov. 2011.

[30] S. Temizer, M. Kochenderfer, L. Kaelbling, T. Lozano-Pérez, and J. Kuchar, "Collision avoidance for unmanned aircraft using Markov decision processes," in *Proc. AIAA Guid., Navigat., Control Conf.*, 2010, p. 8040.

[31] H. Bai, D. Hsu, M. J. Kochenderfer, and W. S. Lee, "Unmanned aircraft collision avoidance using continuous-state POMDPs," in *Robotics: Science and Systems VII*, vol. 1. Cambridge, MA, USA: MIT Press, 2012, pp. 1–8.

[32] Y. Lin and S. Saripalli, "Collision avoidance for UAVs using reachable sets," in *Proc. Int. Conf. Unmanned Aircr. Syst. (ICUAS)*, Jun. 2015, pp. 226–235.

[33] T. Yu, J. Tang, L. Bai, and S. Lao, "Collision avoidance for cooperative UAVs with rolling optimization algorithm based on predictive state space," *Appl. Sci.*, vol. 7, no. 4, p. 329, Mar. 2017.

[34] Y. B. Chen, J. Q. Yu, X. L. Su, and G. C. Luo, "Path planning for multi-UAV formation," *J. Intell. Robot. Syst.*, vol. 77, no. 1, pp. 229–246, 2015.

[35] M. Kothari, I. Postlethwaite, and D.-W. Gu, "Multi-UAV path planning in obstacle rich environments using rapidly-exploring random trees," in *Proc. 48th IEEE Conf. Decis. Control (CDC) Held Jointly With 28th Chin. Control Conf.*, Dec. 2009, pp. 3069–3074.

[36] H. Bayerlein, M. Theile, M. Caccamo, and D. Gesbert, "Multi-UAV path planning for wireless data harvesting with deep reinforcement learning," *IEEE Open J. Commun. Soc.*, vol. 2, pp. 1171–1187, 2021.

[37] J. Ikuno, M. Wrulich, and M. Rupp, *Evolved Universal Terrestrial Radio Access (E-UTRA); Further Advancements for E-UTRA Physical Layer Aspects*, document 3GPP TR 36.814 (v9. 0.0), 2010.

[38] J. Chen, D. Raye, W. Khawaja, P. Sinha, and I. Guvenc, "Impact of 3D UWB antenna radiation pattern on air-to-ground drone connectivity," in *Proc. IEEE 88th Veh. Technol. Conf. (VTC-Fall)*, Aug. 2018, pp. 1–5.

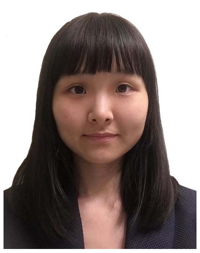
[39] Y. Zeng, J. Lyu, and R. Zhang, "Cellular-connected UAV: Potential, challenges, and promising technologies," *IEEE Wireless Commun.*, vol. 26, no. 1, pp. 120–127, Feb. 2019.

[40] *Study on Enhanced LTE Support for Aerial Vehicles*, document 3GPP TR 36.777 V15.0.0, Dec. 2017.

[41] H. Kretzschmar, M. Spies, C. Sprunk, and W. Burgard, "Socially compliant mobile robot navigation via inverse reinforcement learning," *Int. J. Robot. Res.*, vol. 35, no. 11, pp. 1289–1307, 2016.

[42] C. Chen, Y. Liu, S. Kreiss, and A. Alahi, "Crowd-robot interaction: Crowd-aware robot navigation with attention-based deep reinforcement learning," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 6015–6022.

[43] J. Snape, J. van den Berg, S. J. Guy, and D. Manocha, "The hybrid reciprocal velocity obstacle," *IEEE Trans. Robot.*, vol. 27, no. 4, pp. 696–706, Aug. 2011.



Xueyuan Wang received the B.S. degree in electrical and electronics engineering from the Beijing University of Posts and Telecommunications, Beijing, China, in 2013, and the M.S. degree in electrical engineering and the Ph.D. degree in electrical and computer engineering from Syracuse University, Syracuse, NY, USA, in 2016 and 2021, respectively. Her primary research interests include autonomous systems, heterogeneous millimeter wave communications, the Internet of Things networks, unmanned aerial vehicles-enabled networks, wireless information and power transfer, and machine learning.

M. Cenk Gursoy (Senior Member, IEEE) received the B.S. degree (Hons.) in electrical and electronics engineering from Boğaziçi University, Istanbul, Turkey, in 1999, and the Ph.D. degree in electrical engineering from Princeton University, Princeton, NJ, USA, in 2004. He is currently a Professor with the Department of Electrical Engineering and Computer Science, Syracuse University. His research interests are in the general areas of wireless communications, information theory, communication networks, signal processing, and machine learning. He was a recipient of the Gordon Wu Graduate Fellowship from Princeton University from 1999 to 2003. He received an NSF CAREER Award in 2006. More recently, he received the *EURASIP Journal on Wireless Communications and Networking* Best Paper Award, the 2020 IEEE Region 1 Technological Innovation (Academic) Award, the 38th AIAA/IEEE Digital Avionics Systems Conference Best of Session (UTM-4) Award in 2019, the 2017 IEEE PIMRC Best Paper Award, the 2017 IEEE Green Communications and Computing Technical Committee Best Journal Paper Award, the UNL College Distinguished Teaching Award, and the Maude Hammond Fling Faculty Research Fellowship. He has been the Co-Chair of the 2017 International Conference on Computing, Networking and Communications (ICNC)—Communication QoS and System Modeling Symposium, the 2019 IEEE Global Communications Conference (GLOBECOM)—Wireless Communications Symposium, the 2019 IEEE Vehicular Technology Conference Fall—Green Communications and Networks Track, and the 2021 IEEE GLOBECOM—Signal Processing for Communications Symposium. He is the Aerospace/Communications/Signal Processing Chapter Co-Chair of IEEE Syracuse Section. He is a member of the editorial boards of IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, IEEE TRANSACTIONS ON COMMUNICATIONS, and IEEE TRANSACTIONS ON GREEN COMMUNICATIONS AND NETWORKING; and an Area Editor of IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY. He also served as an Editor for IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS from 2010 to 2015, IEEE COMMUNICATIONS LETTERS from 2012 to 2014, IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS—SERIES ON GREEN COMMUNICATIONS AND NETWORKING from 2015 to 2016, *Physical Communication* (Elsevier) from 2010 to 2017, and IEEE TRANSACTIONS ON COMMUNICATIONS from 2013 to 2018.