Reinforcement Learning for Joint Detection & Mapping using Dynamic UAV Networks

Anna Guerra, *Member, IEEE,* Francesco Guidi, *Member, IEEE,*
Davide Dardari, *Senior, IEEE,* and Petar M. Djurić, *Fellow, IEEE,*

***Abstract*—Dynamic radar networks, usually composed of flying unmanned aerial vehicles (UAVs), have recently attracted great interest for time-critical applications, such as search-and-rescue operations, involving reliable detection of multiple targets and situational awareness through environment radio mapping. Unfortunately, the time available for detection is often limited, and in most settings, there are no reliable models of the environment, which should be learned quickly. One possibility to guarantee short learning time is to enhance cooperation among UAVs. For example, they can share information for properly navigating the environment if they have a common goal. Alternatively, in case of multiple and different goals or tasks, they can exchange their available information to fitly assign tasks (e.g., targets) to each network agent. In this paper, we consider ad-hoc approaches for task assignment and a multi-agent reinforcement learning (RL) algorithm that allow the UAVs to learn a suitable navigation policy to explore an unknown environment while maximizing the accuracy in detecting targets. The obtained results demonstrate that cooperation at different levels accelerates the learning process and brings benefits in accomplishing the team's goals.**

***Index Terms*—Autonomous navigation, Task assignment, Reinforcement learning, Unmanned aerial vehicles.**

I. Introduction

WIRELESS sensor networks, either with terrestrial fixed [1] and dynamic [2] sensors, are widely used for data gathering, sensing, and communications. Among all possible applications, their monostatic and multistatic deployments have been investigated for radar localization and target detection [3].

A step forward has been introducing flying dynamic sensor networks where sensors are integrated onboard unmanned aerial vehicles (UAVs) [4], [5]. A recent review on the use of UAVs for remote sensing, spanning from precision agriculture (e.g., forest monitoring), urban environment and management (e.g., air traffic control) to disaster hazards and rescue (e.g., post-disaster assessment), can be found in [6] and the references therein. In all such situations, networks of UAVs can offer privileged views for gathering radio and vision-based data. Compared to terrestrial fixed networks, the advantages of using UAV-based networks lie in their flexibility, robustness to single-point failure, reconfigurability, and ability to maintain a line-of-sight (LOS) condition with users and other destination points. For example, in [7], [8], swarms of coordinated UAVs equipped with ad-hoc radar sensors are deployed to track a malicious target. Or alternatively, UAVs have been used as a network infrastructure for localization, communications, and other applications [9]–[11].

In this domain, an important line of research is the optimization of the UAV trajectory [12], [13]. In fact, unlike terrestrial sensors, all tasks and navigation must be optimized so as not to waste time flying over areas of little interest from the mission perspective because of the frequent need to recharge batteries [4]. Moreover, UAVs must complete their mission within a finite time horizon, especially if they operate for time-critical applications. For example, in post-disaster situations, targets (e.g., victims) must be detected and localized as quickly as possible by rescuers aided by networks of autonomous UAVs [14]. Several recent papers have studied the UAV trajectory optimization for wireless communication purposes where UAVs are used either as flying base stations (BSs) or users [15]. For example, in [16], [17], the navigation goal was maximizing the communication rates of multiple concurrent cellular users' transmissions. Other contributions focus on localization [8], [18] or minimization of the electromagnetic exposure [19].

Traditionally, the navigation control problem is solved by adopting model-based optimization, e.g., nonlinear programming or dynamic programming [33]. Thanks to the availability of a statistical model, the navigation problem can be written as the minimization (maximization) of a cost (information) function and is solved by relying on classic optimization tools. Usually, the formulated optimization problem also considers constraints for anti-collision, obstacle avoidance, and energy consumption. For example, in [8], [18], the UAV navigation problem is described as the minimization of the uncertainty of target positioning.

Unfortunately, empirical system models are often unavailable or, in some situations, unreliable due to highly fast-changing environments. To this purpose, machine learning (ML) based approaches are of interest to learn a policy that achieves the desired objectives efficiently and in a data-driven fashion [20], [30], [34]–[38]. Among different ML

A. Guerra (corresponding author, e-mail: anna.guerra@cnr.it) and F. Guidi are with the National Research Council of Italy, CNR-IEIIT. D. Dardari is with the WiLAB - DEI "Guglielmo Marconi" - CNIT, University of Bologna, Italy. P. M. Djurić is with ECE, Stony Brook University, Stony Brook, NY 11794, USA. E-mail: petar.djuric@stonybrook.edu.

Table I: Some examples of deep and standard Q-learning based applications in UAV Networks.

Applications	**Optimization Objective**	**Techniques**
Enhancing cellular communications (e.g., flying base-stations)	• Maximize the sum–rate • Maximize the SNR • Balancing UAV power consumption and coverage	• Tabular Q–learning [20] • Double deep Q-network (DDQN) [21] • Dueling Deep Q-network [22]
Detection and/or localization and/or tracking of targets	• Minimize the positioning error • Maximize the detection rate • Maximize the number of detected targets	• Tabular Q–learning [4], [11], [14], [23] • SARSA and Q-learning [24] • Enhanced tabular Q–learning [25] • Deep Q-Networks [23] • Deep RL [26], [27]
Environmental radio mapping	• Minimize the entropy of the map • Maximize the mapping coverage	• Tabular Q-learning [4] • Dueling double deep Q network [28]
Smart sensing (e.g., for wildfires)	• Minimize the traveling time	• Review of deep learning for remote sensing [29]
Item delivery	• Minimize the traveling time	• Recurrent policy gradient algorithm [30]
Internet of Things (IoT)	• Learn the radio channel • Localize unknown nodes • Maximize data collection • Maximize the number of addressed tags.	• Model-aided Deep Reinforcement Learning (Q Learning) [31] • MARL with Deep Q-Learning [32]

approaches, RL and deep RL have been used for UAV policy navigation because of their ability to learn directly by interactions with the surrounding environments [24], [39]–[43]. When the environment has a grid-world representation (e.g., indoors), Q-learning represents a simple and optimal solution because state-action pairs can be represented by a tractable Q-table that is updated at each time instant according to the received rewards [4], [14], [44]. Table I summarizes the use of tabular and deep Q-learning applied to UAV networks. The main disadvantage of tabular Q-learning is the *curse of dimensionality* that occurs for large state and action spaces (e.g., large environments) and leads to increased computational complexity and a slow convergence [30]. The combination of deep learning with reinforcement learning (deep RL) overcomes this issue by relying on neural networks (NNs) for Q-function representations [45]. However, most applications treat NNs as black boxes, and understanding and interpreting deep learning models remains challenging [46]. The lack of interpretability still requires comprehensive treatment, especially for dual-use technologies like those based on UAVs and for safe-critical applications. Moreover, having a tabular representation can help analyze the impact of different parameters and schemes on performance.

A way to accelerate the training of large Q-tables without relying on NNs is having agents cooperating with each other [47]. Different techniques in the literature have been proposed for cooperation, accounting for centralized and decentralized solutions [41], [48]. While centralized solutions usually permit a global view of the environment, decentralized solutions are more flexible but require more intelligent agents. According to Q-learning, several cooperative approaches for sharing the learned experience among the agents' network have already been proposed in [49], [50]. In [50], [51], the authors considered distributed Q-learning approaches for multiple device access in massive machine-type communications scenarios, whereas in [48] the authors analyze a setting where some agents are more expert than others (thus, being more informative) in cooperation.

Given this background, and differently from the optimization objectives evidenced in Table I, in this paper, we aim to study cooperative RL in a dynamic radar network (DRN) of UAVs whose tasks are to detect targets accurately and to enhance their ambient awareness by estimating its occupancy radio map. Cooperation among UAVs will be tackled in a twofold manner: for task assignment when agents within a DRN share different goals, e.g., detect multiple targets, and for UAV navigation when only a single target is present. Through such cooperation, UAVs take actions based on a "global" (network) shared knowledge and reduce the overall learning time, thus improving the network's performance.

The main contributions of this paper can be summarized as follows.

- We propose an ad-hoc DRN architecture, composed of UAVs, for solving joint target detection and environment mapping tasks;
- We investigate cooperative multi-agent Q-learning approaches for solving autonomous navigation of UAVs when agents share the same mission goal so that the required mission time is reduced;
- We investigate different approaches for task assignment when multiple and competing tasks are required during the mission. In particular, we consider either a simple received signal strength indicator (RSSI) based solution, or a random assignment or a multi-armed bandit (MAB)

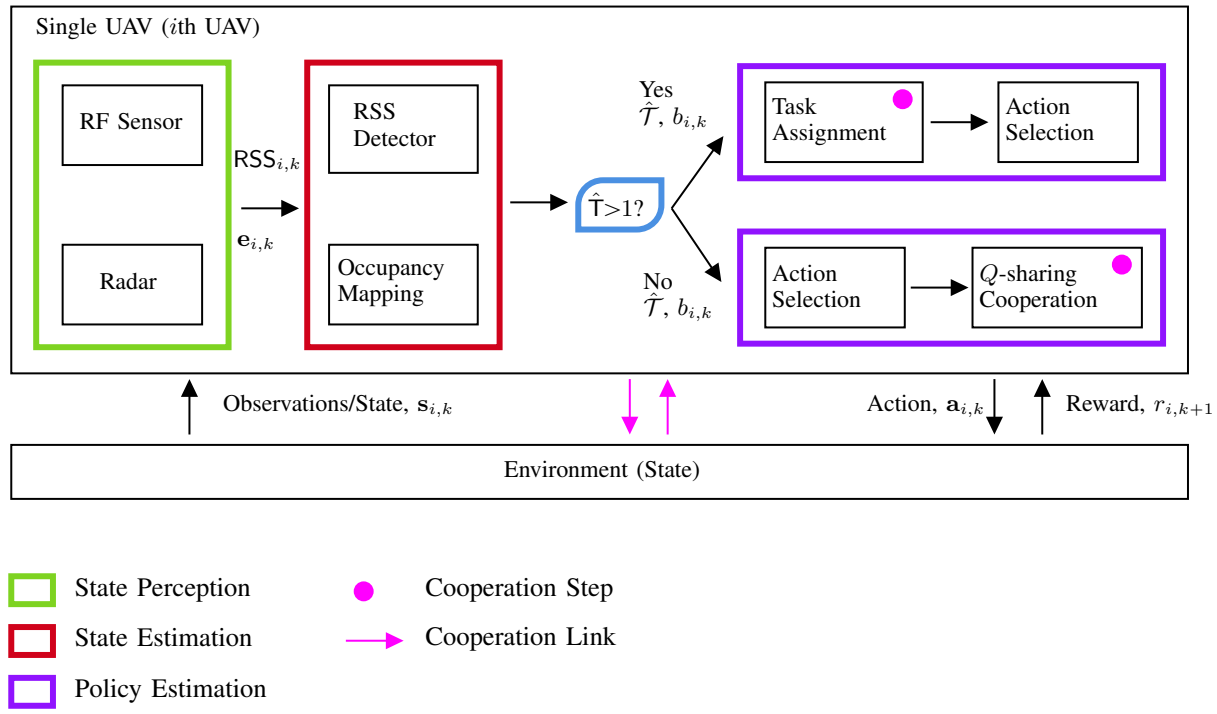


Figure 1: A block diagram for decentralized joint detection, mapping, and navigation. Green, red, purple blocks indicate state perception, state estimation and policy estimation, respectively. The environment is estimated by processing radar observations. More specifically, an RF sensor is used to gather RSS measurements for target detection, indicated as $\mathsf{RSS}_{i,k}$, and a scanning radar permits to collect an Angle-Range matrix, denoted with $\mathbf{e}_{i,k}$, for environment mapping. The true state at time instant k is indicated with $\mathbf{s}_{i,k}$ for the ith UAV. The set and the estimated number of targets are denoted by $\hat{\mathcal{T}}$ and $\hat{\mathsf{T}}$, respectively, whereas the belief of the environment map with $b_{i,k}$. The policy estimation allows the UAV to select an optimized action $\mathbf{a}_{i,k}$. Such an action leads to a new state where the UAV collects a reward indicated with $r_{i,k+1}$.

scheme for properly managing the UAVs-targets assignment at specific time instants of the mission.
- We demonstrate through a comprehensive case study, including Terahertz (THz) mapping, the feasibility of the proposed DRN in various settings. In particular, we investigate the trade-off between the mapping and detection performance while reducing the learning process, highlighting the benefits carried out by cooperation.

The rest of the paper is organized as follows. Sec. II describes the problem formulation for the DRN, whereas Secs. III- IV overview the considered navigation and cooperation approaches. Then, Sec. V reports the considered case study, and final conclusions are drawn in Sec. VI.

II. Problem Formulation

In this paper, we consider a DRN composed of UAVs that, by either collaborating together or acting as independent learners, navigate for detecting active targets in an unknown environment while reconstructing a probabilistic map of it.[1]

[1]In this paper, we focus on active targets sending beacons that can be detected by radio frequency (RF) sensors. Passive targets are instead considered objects whose probabilistic presence is inferred by a mapping procedure.

More specifically, UAVs have the following two tasks:

(i) *Primary (Extrinsic) Task*: High-quality detection of multiple active targets. Practical examples are cooperative users that need to be rescued or hidden malicious targets whose unwanted communication is sniffed within a certain frequency band;

(ii) *Secondary (Intrinsic) Task*: Estimation of an occupancy map of the explored area.

To accomplish them, each UAV performs the following steps: *sensing*, *state estimation*, *task allocation* (for scenarios with multiple targets) and *policy estimation*, depicted in Fig. 1 and described in the following.

a) Sensing: Each UAV is considered equipped with ad-hoc low-cost and low-complexity sensors. A radar working at high frequencies (e.g., millimeter-waves or THz) can be accommodated in a small space despite the adoption of many antennas for accurate beam-steering operations. Such radars are useful for mapping as they can collect a range-angle energy matrix to be processed by a mapping algorithm [52].

On the other hand, different sensors can be used for target detection spanning from vision-based systems to radars. In the following, we will consider a RF sensor able to measure

the RSSI from a target that can be discriminated from other targets through the detected packet ID [53].[2]

Generally speaking, if other sensors gathering different types of data are on board, data-fusion techniques can be used to process heterogeneous information.

b) State Estimation: The state comprises the UAV positions, an occupancy map of the environment, and the ID of an associated target. In our investigated scenario, the map is estimated using a Bayesian filtering approach, namely an occupancy grid (OG). Appendix A shows the basic principles of the adopted OG algorithm. The environment is discretized in cells, and each cell has a binary status (1 if occupied and 0 if free). The goal of the estimation is to infer the a-posteriori probability mass function of the occupancy of each cell based on the history of radar measurements.

In addition, we assume to be able to distinguish the signals coming from different targets, as they use different tones of an orthogonal frequency-division multiplexing (OFDM) signaling scheme, provided that the signal-to-noise ratio (SNR) is above a certain threshold that allows to decode the received signals and extrapolate the sources' IDs.[3]

c) Task Allocation: To avoid situations where more UAVs are likely to get closer to the same target, with the risk of missing other targets, a UAV-target allocation algorithm can be run either at each UAV or at a network level to distribute the available resources better. In the next, we consider the following three solutions described in Sec. IV-B: (i) *Random*: where the allocation is randomly chosen; (ii) *Independent RSSI*: where each UAV is assigned to the target corresponding to the maximum received power; (iii) *Cooperative MAB*: where a MAB formulation is used to describe the problem, and an upper confidence bound (UCB) solution is considered.

d) Policy Estimation: Starting from the estimated state, each UAV should decide where to navigate to maximize joint detection and mapping performance and global network behavior. The functions that map states into actions are called policies. As a first step, each UAV acts as an independent learner, estimates its own policy, and takes a navigation decision. The navigation action drives the UAV to the next position, where an instantaneous reward is collected according to the goodness of the chosen action. Such a reward permits a first update of the policy. In collaborative settings, UAVs can share their knowledge with neighbors or with more expert UAVs (e.g., by exchanging Q-values). After such an exchange, the policy can be further updated. In this sense, in the rest of the paper, we will focus on the capability to make informative navigation decisions, independently or cooperatively, according to multi-agent Q-learning.

[2]Such a system works properly only with active targets. If passive targets are also of interest, the UAVs could use a dual-functional radar for joint detection and mapping, providing easy on-board integrability [54], [55]. However, this setting is beyond the scope of this paper. Our primary aim is to develop decision-making strategies for the navigation of autonomous agents.

[3]Such values were numerically set to $\mathsf{SNR} = 10\,\mathrm{dB}$ in our case study [56].

A. System Model

We consider a set of $\mathcal{M} = \{1, \ldots, i, \ldots, \mathsf{M}\}$ UAVs employed in the environment, and a set of $\mathcal{T} = \{1, \ldots, n, \ldots, \mathsf{T}\}$ targets' IDs to be discovered with certain reliability, i.e., the measured SNR should overcome a desired threshold. We divide the time into a sequence $\bar{K}$ of discrete time instants upper bounded by K to take into account the limited UAV endurance.

1) State-Action Model: In our scenario, the state vector for each UAV $\mathbf{s}_{i,k}$ at time k contains the UAV location, the map of the environment and a detection variable, i.e., $\mathbf{s}_{i,k} = \left[\mathbf{p}_{i,k}, \mathbf{m}_k, n_{i,k}\right]^T$, where $\mathbf{p}_{i,k} = \left[x_{i,k}, y_{i,k}, h\right]^T \in \mathbb{R}^3$ is the true UAV position, $\mathbf{m}_k \in \mathbb{B}^{N_{\mathsf{cell}}}$ is the true map at time k described as a vector of N_{cell} cells in which the map is discretized, and $n_{i,k}$ contains the target ID associated to the ith UAV, and that can be empty in case no target is associated to the considered UAV. The environment is assumed stationary, so that $\mathbf{m}_k = \mathbf{m}, \forall\, k$, with $\mathbf{m} = \left[m_1, \ldots, m_j, \ldots, m_{N_{\mathsf{cell}}}\right]^T$, containing the occupancy value of each cell, i.e., $m_j \in \mathbb{B}$.

In the navigation algorithm, we consider that the state coincides with the UAV positions, that is $\mathbf{s}_{i,k} = \mathbf{p}_{i,k} \in \mathbb{R}^3$. We also assume that the UAVs move in a grid so that $\mathbf{s}_{i,k} = (\mathbf{p}_{i,k} \mod \Delta)$, where Δ is the grid step. Consequently, for a single-agent case, the state space considered for RL navigation purposes is $|\mathcal{S}| = N_{\mathsf{cell}}$.[4] Similarly, the UAV navigation actions can be defined as $\mathbf{a}_{i,k} = \Delta\mathbf{p}_{i,k} = \left[\Delta x_{i,k}, \Delta y_{i,k}, 0\right]^T \in \mathbb{R}^3$ where $\Delta\mathbf{p}_{i,k}$ is a position displacement in a continuous space. As before, since the UAVs are constrained to move in a grid with only 4 available actions, we have $\mathbf{a}_{i,k} = (\Delta\mathbf{p}_{i,k} \mod \Delta)$ and the action space is here defined as $\mathcal{A} = \left\{ [\Delta, 0], [-\Delta, 0], [0, \Delta], [0, -\Delta] \right\}$ corresponding with right, left, up and down directions.

2) Observation Model for Target Detection: The UAVs initially sense the environment through a detection module whose intent is to reveal the presence of a collaborative target that periodically broadcasts a beacon in the environment. Then, if the received packets are correctly demodulated, the UAVs collect a vector with the RSSIs that, for time instant k, is

$$\mathsf{RSS}_{i,k} = \Big\{ \mathsf{RSS}_{1,i,k}, \ldots, \mathsf{RSS}_{n,i,k}, \ldots, \mathsf{RSS}_{\hat{\mathsf{T}}_i,i,k} \big| \quad \text{s.t. } \mathsf{RSS}_{n,i,k} \geqslant \xi, \forall n \in \hat{\mathcal{T}}_i \Big\}, \tag{1}$$

where $\hat{\mathcal{T}}_i$ refers to the set containing the target IDs detected by the ith UAV with cardinality $\hat{\mathsf{T}}_i$, and $\mathsf{RSS}_{n,i,k}$ is the RSSI measured from the nth target at time instant k, where we assume that the duration of the beacon is less than the interval

[4]When the dimension of the state space is large (e.g., for large outdoors), policy iteration might suffer from the "curse of dimensionality" [57].

between k and $k+1$. Here the $\mathsf{RSS}_{n,i,k}$ (in dBm) is modeled according to a log-normal power loss model as follows [58]:

$$\mathsf{RSS}_{n,i,k}\ [\mathrm{dBm}] = 30 + k_0 - 10\,\alpha_{\mathsf{pl}} \log_{10}\frac{d_{n,i,k}}{\mathsf{d}_1} + S_{\mathsf{h}}, \quad (2)$$

where α_{pl} is the path-loss exponent, $d_{n,i,k}$ is the distance between the ith UAV and the nth target at time k, S_{h} models the shadowing effect, and it is here considered normally distributed as $S_{\mathsf{h}} \sim \mathcal{N}\left(0, \sigma_s^2\right)$, with σ_s being the shadowing spread, and where k_0 is defined as

$$k_0 = 10\log_{10}\left[P_n\, G_n\, G_i \frac{\lambda^2}{\left(4\,\pi\right)^2 \mathsf{d}_1^2}\right] - \mathbb{1}_{i,n,k}\, L_{\mathrm{NLOS}}\,, \quad (3)$$

is the received power at $\mathsf{d}_1 = 1\,\mathrm{m}$, where λ is the wavelength, P_n is the nth target's transmitted power, G_n (G_i) refers to the transmitting (receiving) antenna gain, $\mathbb{1}_{i,n,k}$ is an indicator function set to one if there is a non line-of-sight (NLOS) between the target and the UAV at time instant k, and L_{NLOS} is the additional attenuation due to the blockages creating the NLOS condition [14], [59], [60]. Note that $\mathsf{RSS}_{n,i,k}$ is a function of the UAV and target positions and distance $d_{n,i,k}$, and in the following it is used for defining the rewards. Typical values of the shadowing standard deviation can be of $1.70\ \mathrm{dB}$ and $3\ \mathrm{dB}$ in LOS and NLOS situations respectively [58]. The path-loss exponent was set to 2. Moreover, in presence of NLOS, in our case study, we considered an attenuation of $L_{\mathrm{NLOS}} = 30\,\mathrm{dB}$. Next, we describe RL in DRNs for navigation, and then we discuss cooperation in Sec. IV

III. Navigation Policy

A. Single-Agent Markov Decision Process

A Markov decision process (MDP) is defined by the tuple comprising the state space $\mathcal{S}$, the action space $\mathcal{A}$, the reward space $\mathcal{R}$, and the probability of transitioning from one state $\mathbf{s}_k$, at time instant k, to the next state $\mathbf{s}_{k+1}$ [57].[5]

The actions of the ith agent are selected according to a policy $\pi_i\left(\mathbf{a}_{i,k}|\mathbf{s}_{i,k}\right)$ that is the conditional probability mass function of the action. The optimal policy selects an action according to

$$\mathbf{a}^*_{i,k} = \arg\max_{\mathbf{a}} Q_{\pi_i}\left(\mathbf{s}_{i,k}, \mathbf{a}\right), \quad (4)$$

where the Q-function, $Q_{\pi_i} = Q_{\pi_i}\left(\mathbf{s}_{i,k}, \mathbf{a}_{i,k}\right)$, is the expected sum of discounted rewards over all possible policies and is given by

$$Q_{\pi_i} = \mathbb{E}_{\pi_i}\left\{\sum_{l=0}^{\infty} \gamma^l R_{i,k+l+1} \middle| S_{i,k} = \mathbf{s}_{i,k}, A_{i,k} = \mathbf{a}_{i,k}\right\}, \quad (5)$$

with $0 \leqslant \gamma \leqslant 1$ being the discount rate, and where $\{R_{k,i}, S_{k,i}, A_{k,i}\}$ are the random variables for the ith agent related, respectively, to rewards, states, and actions at time instant k taking values in $\{\mathcal{R}, \mathcal{S}, \mathcal{A}\}$. The expected reward at time instant $k+1$ for the state-action pair is

$$r_{i,k+1}(\mathbf{s}_{i,k}, \mathbf{a}_{i,k}) = \mathbb{E}\left[R_{i,k+1} | S_{i,k} = \mathbf{s}_{i,k}, A_{i,k} = \mathbf{a}_{i,k}\right]. \quad (6)$$

Optimal policies share the same *optimal action-value function* for policy π defined as [57]

$$Q_i^*\left(\mathbf{s}_{i,k}, \mathbf{a}_{i,k}\right) = \max_{\pi} Q_{\pi}\left(\mathbf{s}_{i,k}, \mathbf{a}_{i,k}\right), \quad (7)$$

$\forall \mathbf{s}_{i,k} \in \mathcal{S}, \forall \mathbf{a}_{i,k} \in \mathcal{A}$.

B. Q-learning for UAV Navigation

Q-learning is an off-policy temporal-difference (TD) control algorithm where the policy is learned run-time while the UAV navigates the environment. It is a model-free tabular approach with the possibility of choosing a random action. The simplest solution is represented by the ϵ-greedy approach [57], [61], [62], where a random action is selected with a probability given by ϵ. Other variants of these approaches account for exploration only at the beginning (ϵ-first strategy) or for a time-decaying exploration (ϵ-decaying strategy) to converge to a quasi-optimal solution. The advantages of using TD methods instead of Monte Carlo or dynamic programming is that there is no need for a model, and an update of the return (i.e., cumulative rewards) is made at each time step.

For discrete states and actions, the Q-value in (5) can be represented by a Q-table that, at each time instant and for each agent, is updated by [57]

$$Q_i(\mathbf{s}_{i,k}, \mathbf{a}_k) \leftarrow Q_i(\mathbf{s}_{i,k}, \mathbf{a}_{i,k}) + \alpha\left[r_{i,k+1} + \gamma \max_{\mathbf{a}} Q(\mathbf{s}_{i,k+1}, \mathbf{a}) - Q_i(\mathbf{s}_{i,k}, \mathbf{a}_{i,k})\right], \quad (8)$$

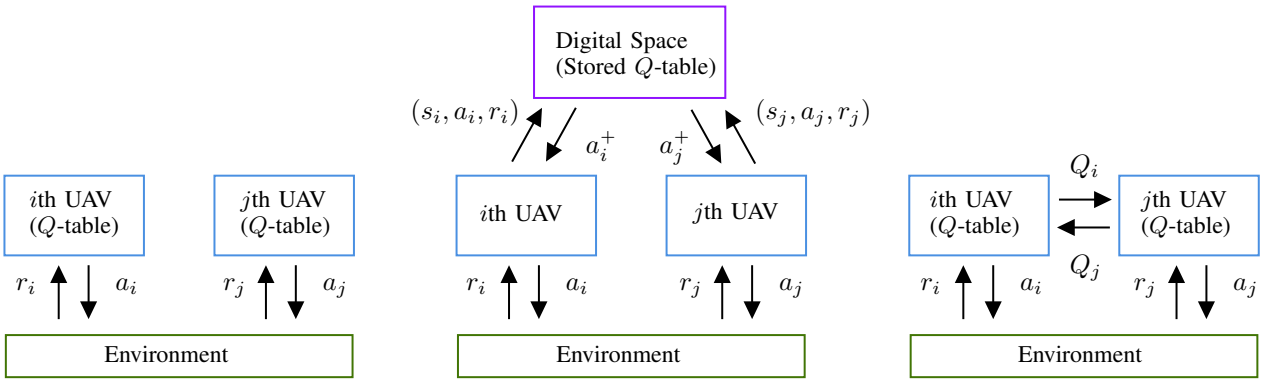
where α is the learning rate, and the max operator is used to have a greedy policy. In this case, the learned action-value function directly approximates the optimal action-value function in (7), independently from the policy being followed.

C. Navigation Rewards

One of the most important aspects when adopting RL is the reward shaping that drives the agents' behavior in the desired manner [63], [64]. To this purpose, we recall that the UAV network has a primary (extrinsic) and a secondary (intrinsic) task and associated rewards.

The extrinsic rewards are usually task-specific, and they associate a state-action pair into a real-valued reward, whereas the intrinsic rewards only indirectly depend on the world's state through the beliefs estimated by UAV about such a state [63].

[5]The problem can also be formulated as a constrained MDPs to account for anti-collision and safety constraints. In our paper, we include such constraint information in terms of penalties. As soon as a UAV senses an obstacle through its proximity sensors, then it selects another action.



a. Independent Learning b. Centralized Cooperative Learning c. Distributed Cooperative Learning

Figure 2: From the left to the right: Independent, Centralized, and Distributed Cooperative Learning schemes.

Thus, as in [62], we can write the reward as a weighted sum of extrinsic and intrinsic rewards as[6]

$$r_{i,k+1} = \eta\, r^{\mathsf{e}}_{i,k+1} + r^{\mathsf{i}}_{i,k+1}, \tag{9}$$

where

$$r^{\mathsf{e}}_{i,k+1} = r^{\mathsf{d}}_{i,k+1}, \qquad r^{\mathsf{i}}_{i,k+1} = \xi_1\, r^{\mathsf{c}}_{i,k+1} + \xi_2\, r^{\mathsf{m}}_{i,k+1}, \tag{10}$$

where $r^{\mathsf{d}}_{i,k+1}$, $r^{\mathsf{c}}_{i,k+1}$, and $r^{\mathsf{m}}_{i,k+1}$ relate to the target detection, the mapping coverage, and accuracy, respectively, and η, ξ_1, and ξ_2 are the related weight coefficients whose impact will be studied and discussed in the numerical results.

The detection reward is expressed as a function of the RSSI measured from the assigned target as

$$r^{\mathsf{d}}_{i,k+1} \triangleq \frac{\mathsf{RSS}_{n_i,i,k+1}}{\mathsf{RSS}_{\max}}, \tag{11}$$

where $\mathsf{RSS}_{n_i,i,k+1}$ is the measured RSSI from the $n_{i,k+1}$th target associated to the ith UAV at time $k+1$, and $\mathsf{RSS}_{\max}$ is the RSSI a UAV would experience at a distance of a single cell (i.e., the minimum possible distance) from a target.[7]

The mapping coverage reward is given by

$$r^{\mathsf{c}}_{i,k+1} \triangleq \frac{\sum_{j\in\mathcal{I}_{i,k+1}} \mathbb{I}(j \in \mathcal{D}_{i,k+1})}{N_{\mathsf{cell}}}, \tag{12}$$

where $\mathbb{I}$ is the indicator function, i.e., $\mathbb{I}(x) = 1$ if x is true and 0 otherwise, $\mathcal{D}_{i,k+1} \subseteq \mathcal{I}_{i,k+1}$ indicates the cells visited for the first time, at time $k+1$, whereas $\mathcal{I}_{i,k+1}$ represents the set of indices of all the cells illuminated by the ith UAV at the same $k+1$ th instant. In other words, the higher the number of cells visited for the first time, the higher the reward. Finally, $r^{\mathsf{m}}_{i,k+1}$ is defined as follows:

$$r^{\mathsf{m}}_{i,k+1} = -\sum_{j\in\mathcal{I}_{i,k+1}} b_{i,k+1}(m_j)\, \log_2\left(b_{i,k+1}(m_j)\right), \tag{13}$$

where $b_{i,k+1}(m_j)$ is the belief of the occupancy state of the j th cell as predicted by the i th agent at time slot $k+1$. Notably, this reward aims to push actions that minimize the uncertainty of the map in the shortest possible time.

Note that obstacles are assumed to be detected with (proximity) sensors that allow the agent to avoid them by including numerical penalties in the Q-table.

IV. Multi-Agent Cooperation

According to Fig. 1, cooperation can be intended in a twofold manner. In the first one, if a group of UAVs shares a common (detection) task, an exchange of Q-values can speed up the mission completion. In the second one, if multiple (detection) tasks should be completed by the network, then the cooperation can be implemented through task assignment between UAVs.

A. Common Task: Q-sharing

When UAVs are networked and share a common (detection) task, they coordinate together for navigation to achieve the mission goal more rapidly than if they operate independently. In this section, we extend the single-agent framework of Sec. III-A to the multi-agent case.

According to Fig. 2, two types of UAVs are conceived: *(i)* independent learners; and *(ii)* cooperative learners that can work either in a distributed or in a centralized manner. More specifically, we have:

- *Independent Learning*: Each UAV finds the best policy in an independent way by solving the optimization problem in (4) for their local Q-table, i.e., $Q_i\left(\mathbf{s}_{i,k}, \mathbf{a}_{i,k}\right)$. Thanks to the first stage of single-agent Q-learning, they select their own action and move to the next position (i.e.,

[6]The reward model in (10) is also similar to the theory of RL multi-objectivation [65] where the most common approach is to employ a scalarization of the multi-objectives [66]. Setting these weights a priori to achieve a particular trade-off requires extensive parameter tuning [65].

[7]We suppose the positions of targets are unknown. For this reason, we cannot assume to have a cost function in the form of $p\left(\mathbf{p}_{i,k}\right) = -1 - d_{i,n_i,k}$ as in [67].

$\mathbf{p}_{i,k+1}$) where they collect an instantaneous (sample) reward $r_{i,k+1}$.

- *Centralized Cooperative Learning:* In this case, the UAVs share the same Q-table, that is, $Q_i = Q, \forall i \in \mathcal{M}$. Thus, each UAV indirectly knows what has been experienced by the others through the Q-table, which is updated by the ith agent as

$$Q(\mathbf{s}_{i,k}, \mathbf{a}_{i,k}) \leftarrow Q(\mathbf{s}_{i,k}, \mathbf{a}_{i,k}) + \alpha \left[r_{i,k+1} + \gamma \max_{\mathbf{a}} Q(\mathbf{s}_{i,k}, \mathbf{a}) - Q(\mathbf{s}_{i,k}, \mathbf{a}_{i,k}) \right]. \quad (14)$$

  Note that while actions are selected on-board by each UAV, the Q-table is shared among all of them (e.g., it can be stored in an edge or cloud), and this allows the UAVs to make more informed decisions. This approach has the same disadvantages as centralized architecture, i.e., a low degree of robustness and the need to communicate with a central node or share all the Q-tables.
- *Distributed Cooperative Learning*: In this case, UAVs share some learning information (e.g., Q-values) with other UAVs. This allows updating their own Q-tables by considering the knowledge acquired by the others. Each agent updates its own Q-table according to a specific function, i.e., $Q_i \leftarrow f\left(Q_i, \{Q_\nu\}_{\nu \in \mathcal{M}_{i,k}}\right)$ where $\mathcal{M}_{i,k}$ are the UAVs within the communication range of the ith agent (also indicated as "neighbors"). In the next, we consider $\mathcal{M}_{i,k} = \mathcal{M}_i$, $\forall k$.

Since there are different ways to conceive distributed cooperative approaches, in the following, we highlight different possible techniques and implementations of the cooperative function $f(\cdot)$ according with [68].

a) Distributed Cooperation with Maximum Q-values: Each agent i updates its Q-table by substituting each Q-value with the related best Q-value among all the Q-tables of neighboring agents. By omitting the temporal index, for each state-action pair, the Q-value is updated by

$$Q_i(\mathbf{s}, \mathbf{a}) \leftarrow f\left(Q_i, \{Q_\nu\}_{\nu \in \mathcal{M}_i}\right) = Q^{\max}(\mathbf{s}, \mathbf{a}) = \max_{\nu \in \{i, \mathcal{M}_i\}} Q_\nu(\mathbf{s}, \mathbf{a}), \quad (15)$$

$\forall\, i \in \mathcal{M}$, $\mathbf{s} \in \mathcal{S}$, $\mathbf{a} \in \mathcal{A}$. Each Q-table entry indicated by the pair $(\mathbf{s}, \mathbf{a})$ is substituted with the maximum Q-values among the corresponding entries of all neighbors.

In this type of cooperation, all agents must share the entire Q-table with the risk of slow and time-consuming communication operations.[8]

b) Distributed Cooperation with Absolute Q-Values: The previous approach might suffer because negative rewards are neglected since they are not considered useful, even if they are important, as they can prevent other agents from repeating the same mistakes.

Thus, a possible alternative is to account for the absolute value of the Q-table (BestAbs-Q). In this case, (15) becomes

$$Q_i(\mathbf{s}, \mathbf{a}) \leftarrow f\left(Q_i, \{Q_\nu\}_{\nu \in \mathcal{M}_i}\right) = Q^{\text{best}}(\mathbf{s}, \mathbf{a}) = \max_{\nu \in \{i, \mathcal{M}_i\}} |Q_\nu(\mathbf{s}, \mathbf{a})|. \quad (16)$$

c) Distributed Cooperation with Averaged Q-Values: Each agent averages the best Q-table with its current Q-table as follows:

$$Q_i(\mathbf{s}, \mathbf{a}) \leftarrow f\left(Q_i, \{Q_\nu\}_{\nu \in \mathcal{M}_i}\right) = \frac{Q^{\text{best}}(\mathbf{s}, \mathbf{a}) + Q_i(\mathbf{s}, \mathbf{a})}{2}, \quad (17)$$

$\forall\, i \in \mathcal{M}$, $\mathbf{s} \in \mathcal{S}$, $\mathbf{a} \in \mathcal{A}$. In this way, the best values of the Q-table are mixed with the current Q-values of the ith agent, such that new and past information is balanced.

B. Multiple Tasks: UAV-Target Association

We refer to the task allocation of the target-UAV association procedure that maximizes the number of discovered targets in a given mission time.

Let $\mathcal{L} = \{1, 2, \ldots, \ell, \ldots, \mathsf{L}\}$ be the set of all possible associations. The output of the assignment is a vector containing the estimated association. Such a vector is a part of a matrix $\mathbf{A}$ of size $\mathsf{L} \times \mathsf{M}$ where $\mathsf{L} = |\mathcal{L}|$ is the cardinality of $\mathcal{L}$. Each matrix entry is an index identifying the target associated with the considered UAV.

We recall that the term "discovered target ID" refers to a label associated with targets detected with a received power larger than a predefined threshold ξ as defined in (1). Moreover, the task of target discovery ends when the aforementioned targets are detected with a RSSI higher than $\mathsf{RSS}_{\mathsf{max}}$. In the following, we describe the proposed solutions.

1) Cooperative UAV-Target Association: The cooperative solution is based on MAB. MAB is a sequential decision process equivalent to one-state MDP whose objective is the maximization of the cumulative payoff/reward obtained in a sequence of decisions [69], [70]. Indeed, differently from Q-learning, it is defined by a reward and a set of actions (i.e., arms), but it does not entail the concept of state transitions. Among the possible solutions for the MAB problem, the UCB allows picking arms by solving the dilemma *exploration vs. exploitation* in a closed form [69], [71]. In the following, we propose an ad-hoc UCB-based approach such that the target-UAV association is performed in a fashion that it avoids assigning the same target to multiple UAVs.

The UCB-based algorithm consists of four phases: *initial*, *explore-all*, *training* and *association* phases, and it works as follows [72]:

a) Initial Phase: The UAVs performs an initial measurement campaign with the intent to reveal the presence of a target through its transmitted beacon, as described in

[8] A simplified version is to restrict the shared information to only the UAV's state at time instant k, i.e., $Q_i(\mathbf{s}_{i,k}, \mathbf{a}) \leftarrow f\left(Q_i, \{Q_\nu\}_{\nu \in \mathcal{M}_i}\right) = Q^{\max}(\mathbf{s}_{i,k}, \mathbf{a}) = \max_{\nu \in \{i, \mathcal{M}_i\}} Q_\nu(\mathbf{s}_{i,k}, \mathbf{a})$, $\forall\, i \in \mathcal{M}$, $\mathbf{a} \in \mathcal{A}$.

Sec. II-A2. At the end of this phase, all the UAVs combine the gathered information and a set with $\hat{\mathcal{T}} = \bigcup_{i\in\mathcal{M}} \hat{\mathcal{T}}_i$ detected ID targets is created with cardinality $\hat{\mathsf{T}} = |\hat{\mathcal{T}}|$. The number of arms (i.e., of possible associations), indicated with $\mathsf{L} = |\mathcal{L}|$, is determined as follows:

$$\mathsf{L} = \begin{cases} \mathsf{M}! & \text{if} \quad \mathsf{M} = \hat{\mathsf{T}}, \\ \frac{\mathsf{M}!}{(\mathsf{M}-\hat{\mathsf{T}})!} & \text{if} \quad \mathsf{M} > \hat{\mathsf{T}}, \\ \frac{\hat{\mathsf{T}}!}{(\hat{\mathsf{T}}-\mathsf{M})!} & \text{if} \quad \mathsf{M} < \hat{\mathsf{T}}, \end{cases} \tag{18}$$

where $x!$ indicates the factorial of x. Note that if the number of targets is below the number of agents, some UAVs will remain without targets and dedicate their efforts only to environment radio mapping. On the contrary, if the number of targets is larger than the number of UAVs, some targets will be served later.

b) Explore-All Phase: After the initial procedure, with the creation of $\hat{\mathcal{T}}$, there is a training phase where all the arms $\ell \in \mathcal{L}$ are played once. For each arm, played at instant $t = \ell$, we observe a global reward $\bar{r}_\ell$, and we update the average reward due to the choice of arm ℓ

$$\bar{r}_\ell = \frac{1}{\mathsf{M}} \sum_{i=1}^{\mathsf{M}} r_{\ell,i}, \qquad \hat{\mu}_\ell = \frac{\bar{r}_\ell}{N_\ell}, \tag{19}$$

where N_ℓ is the number of times the ℓ-th arm is selected that, for this phase, is equal to 1 (i.e., $\hat{\mu}_\ell = \bar{r}_\ell$), and $r_{\ell,i}$ is evaluated according to (11).

c) Training Phase: After the *Explore-All* phase, the UCB solves the trade-off between exploration (e.g., choosing a random action) and exploitation (e.g., choosing an action according to the collected information) in a closed form. By assuming Gaussian distributions and a known standard deviation of rewards, the objective reduces to compute the best estimate for the mean value of the reward to pick the best arm. By accounting for τ training instants, at the tth training time, with $\mathsf{L} < t \leqslant \tau$, the choice of an UAV-target allocation action is performed by picking the ℓ-th arm as

$$\ell_t = \arg\max_{\ell} \left(\hat{\mu}_\ell + \sqrt{\frac{2\ln(t)}{N_\ell}} \right), \tag{20}$$

where, in this case, it holds $\hat{\mu}_\ell$ as defined in (19) which is the estimated average cumulative reward collected so far for arm ℓ and $\bar{r}_\ell$ is updated as

$$\bar{r}_\ell = \sum_{\nu=1}^{t} \left[\frac{1}{\mathsf{M}} \sum_{i=1}^{M} r_{\ell,i}(\nu) \cdot \mathbb{1}(\ell,\nu) \right], \tag{21}$$

where $r_{\ell,i}(\nu)$ is the reward associated to agent i due to choice ℓ at time instant k, and $\mathbb{1}(x)$ is the indicator function defined as in (12) as

$$\mathbb{1}(\ell,\nu) = \begin{cases} 1, & \text{if } \ell\text{th arm selected at time instant } \nu \\ 0, & \text{otherwise.} \end{cases} \tag{22}$$

The rationale behind the utility function in (20) is to find a trade-off between exploitation and exploration based on time and the number of times an arm has been chosen. The exploration part is included with the term $\sqrt{\frac{2\ln(t)}{N_\ell}}$: when N_ℓ is large, then this term goes to zero, and the agent can rely on its already acquired knowledge. Concurrently, the exploitation term described by $\hat{\mu}_\ell$ will be more accurate as time passes.

d) Association Phase: Finally, the arm selected for the association is $\hat{\ell} = \ell_\tau$ where ℓ_τ is computed as in (20).

To avoid the need to have the UCB running continuously, intermittent MAB mode can be used to enable UAVs making assignments only in a few instants to save power.

2) Independent UAV-Target Association: The previous approach allows the coordination of multiple UAVs for assigning separate tasks, but it might entail the exchange of several messages between each UAV and a central node, which can be a UAV of the network or an edge/cloud. This, however, can imply extra power consumption.

To overcome this limitation, a viable and simple solution is that each UAV decides on its own to which target to associate according to the measured environmental characteristics. In this sense, the simplest and most intuitive solution is that each UAV picks the target it reveals with the highest RSSI, even though other agents might have the same goal. Such association rule aligns with [67], [73].

In operating like this, a twofold aspect merits attention: the first is that the entire swarm of agents might go toward the same goal, neglecting other targets that need support. On the other hand, this solution can also be adopted as a backup plan if the network's connectivity prevents using the MAB approach.

Thus, omitting the time index k, in this case, we have that the association is defined by a scalar[9] given by

$$\{\mathbf{A}\}_{1,i} = \arg\max_{n} \mathsf{RSS}_{n,i}, \qquad \forall i \in \mathcal{M}, \tag{23}$$

where $\mathsf{RSS}_{n,i}$ is defined as in (1), and $\mathbf{A}$ contains a single row.

In particular, the $\arg\max$ operation in (23) searches the highest RSS value inside the vector RSS_i and returns the corresponding index.

3) Random UAV-Target Association: The third approach entails the adoption of a completely random target-UAV assignment procedure which can also decide that a UAV is assigned only for exploration. Each matrix element is set as $\{\mathbf{A}\}_{1,i} = \chi$ where $\chi \sim \left\lfloor \mathcal{U}\left(1, \hat{\mathsf{T}}\right) \right\rfloor$ is a uniformly distributed random variable considering integer values between 1 and $\hat{\mathsf{T}}_i$, and $\lfloor\cdot\rfloor$ is the rounding down operator.

From one side, this approach is not efficient, especially when an agent is close to a target but is assigned to another one. But on the other hand, it also exploits the concept of exploration in the case of resource allocation, which could be

[9]Note that each UAV only knows its target association, as the procedure is performed in a completely independent manner.

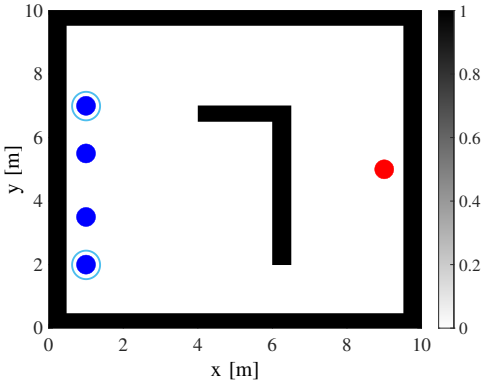


Figure 3: Reference occupancy map for the single-target case. The UAV initial positions are indicated with blue markers, and the source positions with red markers. Double circle markers indicate the positions of UAVs when only two are considered in simulations.

enlightening when the accumulated experience is insufficient to make informative decisions.

Note also that $\mathbf{A}$ is a single vector for the last two approaches, as the task assignment does not require any learning procedure but is essentially based on an independent choice.

V. Numerical Results

In this section, we provide a numerical investigation of the proposed techniques for task assignment and navigation of a network of agents exploring an unknown indoor environment while detecting the presence of targets. The mission time is described by episodes of duration K, so that the generic discrete-time temporal index is given by $t = (e-1)\,K + k \in \{1, 2, \ldots, K\,N_{\mathsf{ep}}\}$, where $k \in \{1, 2, \ldots, K\}$ is a discrete-time index of each episode and $e \in \{1, 2, \ldots, N_{\mathsf{ep}}\}$ is an index of a generic episode.

a) Onboard Sensors: For this case study, we used a sub-THz multiple-antenna radar for environment mapping and a radio receiver for active target detection.

The sub-THz radar was equipped with a planar squared array of 100 antennas working at $140\,\mathrm{GHz}$ with a maximum gain of $26\,\mathrm{dBi}$. The transmitted power and the signal bandwidth were set to $5\,\mathrm{dBm}$ and to $1\,\mathrm{GHz}$, respectively, whereas the observation window (time frame) was set to $50\,\mathrm{ns}$ in accordance to the size of the environments.[10] We considered 25 steering directions with angles between -60 and 60 degrees with a step of 5 degrees. Finally, the radar noise power was set to $-80\,\mathrm{dBm}$. Please refer to Appendix A for further details.

The RF receiver for target detection worked at a central frequency of $2.4\,\mathrm{GHz}$ with a bandwidth of $120\,\mathrm{MHz}$ [53]. The transmitted power of each target was $10\,\mathrm{dBm}$, and the receiver noise power was set to $-90\,\mathrm{dBm}$. Finally, we assumed that the UAV is equipped with a conventional omnidirectional antenna of $G = 5\,\mathrm{dBi}$ gain.

b) Figure of Merits: Next, we assess the source detection performance in terms of the rate of experiencing a certain SNR regime (Success Rate (SR)). Let's define the time instant at which the mission can be considered successfully completed as

$$\hat{t} = (\hat{e}-1)\,K + \hat{k} \tag{24}$$

$$\text{s.t.} \begin{cases} \mathsf{SNR}^{(\hat{e})}_{1,i,\hat{k}} \geqslant \zeta, \forall i \in \mathcal{M}, & \mathsf{T} = 1 \\ \exists\, i \;\middle|\; \mathsf{SNR}^{(\hat{e})}_{n,i,\hat{k}} \geqslant \zeta, \forall n \in \mathcal{T}, & \mathsf{T} > 1 \end{cases}, \tag{25}$$

with ζ being a threshold set to guarantee a reliable detection rate (i.e., the expected SNR at a distance of $2\,\mathrm{m}$ from the source). More specifically, the condition for $\mathsf{T} = 1$ in (24) is valid for the single-target scenario and indicates that all UAVs should experience a SNR above a certain threshold to reach the mission goal. Instead, the condition for $\mathsf{T} > 1$, i.e., valid for the multiple target case, indicates that the mission ends when all the targets are detected with an SNR over the threshold. Then, for each episode e, we define the SR as

$$\mathsf{SR}_e = \frac{1}{N_{\mathrm{MC}}} \sum_{m=1}^{N_{\mathrm{MC}}} \left(\frac{1}{K} \sum_{k=1}^{K} \mathbb{1}\left(t \geqslant \hat{t}\right) \right), \tag{26}$$

where N_{MC} is the number of Monte Carlo iterations.

To obtain a quantitative evaluation of the mapping performance, we consider the image similarity (IS) index defined as [52], [74]

$$\mathsf{IS}\left(\hat{\mathbf{m}}, \mathbf{m}\right) = \sum_{c \in \mathcal{C}} d\left(\hat{\mathbf{m}}, \mathbf{m}, c\right) + d\left(\mathbf{m}, \hat{\mathbf{m}}, c\right), \tag{27}$$

where $\mathbf{m}$ is the actual occupancy map taking value $m_j \in \mathcal{C} = \{0, 1\}, \forall j \in \mathcal{M}$ with $\mathcal{M}$ being the set of all grid cell indexes, $\hat{\mathbf{m}}$ is the estimated map with $\hat{m}_j = 0$ if $b_k\left(\mathbf{m}_k\right) \leqslant 0.4$ and $\hat{m}_j = 1$ if $b_k\left(\mathbf{m}_k\right) > 0.6$, and $d\left(\mathbf{m}_1, \mathbf{m}_2, c\right)$ is defined as

$$d\left(\mathbf{m}_1, \mathbf{m}_2, c\right) = \frac{\sum_{m_{1,i}=c} \min\left(d_{\mathsf{M}}\left(m_{1,i}, m_{2,j}\right) | m_{2,j} = c\right)}{N_c}, \tag{28}$$

with N_c being the number of times a cell in map $\mathbf{m}_1$ has the occupancy value c, and $d_{\mathsf{M}}\left(m_{1,i}, m_{2,j}\right)$ is the Manhattan distance between the ith cell of the map $\mathbf{m}_1$ and the jth cell of the map $\mathbf{m}_2$, both having the same occupancy value c.

The mapping metric in (27) can be computed for each UAV and at each Monte Carlo cycle.

c) Simulation Environment: The reference scenario is displayed in Fig. 3 where the color of each cell represents its occupancy value: empty cells are displayed in white, whereas occupied cells are in black. To describe complete uncertainty about the map status, we initially set $b_0\left(m_j\right) = 0.5$, $\forall j \in \mathcal{M}$. The navigation task was solved by running a multi-agent tabular Q-learning where the learning parameters were set to

[10] Note that the related maximum range is of $7.5\,\mathrm{m}$ considering the radar two-way link.

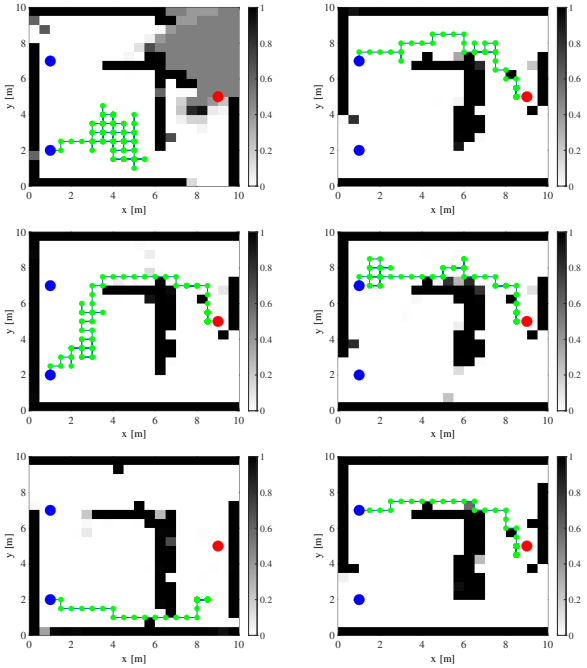


Figure 4: Example of trajectories for the last episode, i.e., $e = 100$, and two agents, and for a single Monte Carlo iteration. Reward weights were $\eta = 1, \xi = 0$. Top: Independent learning, Middle: Centralized cooperative learning, Bottom: Distributed cooperation with BestAbs-Q.

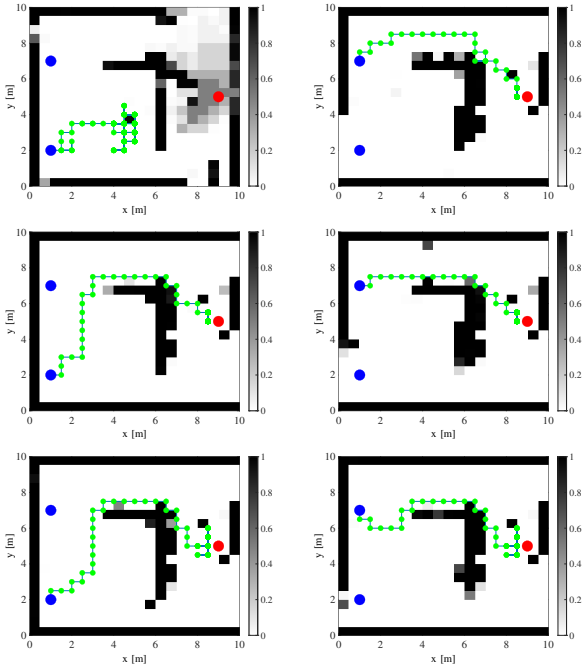


Figure 5: Example of trajectories for the last episode, i.e., $e = 100$, and two agents, and for a single Monte Carlo iteration. Reward weights were $\eta = 1$ and $\xi = 0.3$. Top: Independent learning, Middle: Centralized cooperative learning, Bottom: Distributed cooperation with BestAbs-Q.

$\alpha = 0.99$, $\gamma = 0.9$, and the probability of taking a random action, i.e., ϵ, was considered as a time-decaying function to favor exploitation phases over exploration behaviors (similar to *decayed epsilon-greedy* [75]). For this reason, we have considered the empirical strategy reported in Table II.

We fixed the mission time K for each episode to 150 and the number of episodes N_{ep} to 100. The training episodes allowed UAVs to leverage over prior knowledge acquired through time and experience.

We denote with $\xi = \xi_1 = \xi_2$ the mapping weight.[11]

[11]We omit a discussion about the convergence to an optimal policy because it would depend on the chosen parameters, and it would require lengthy training procedures that are not compliant with time-critical missions [76]. Moreover, for a standard Q-learning approach, it has already been demonstrated that, for a large number of episodes, this approach converges to its optimum [76]–[79]. The conditions to achieve such convergence are related to the exploration policy and learning rate [80]–[82].

Table II: Adopted *time decayed epsilon-greedy* rule.

Step \ Episode	$e < N_{\text{ep}}/2$	$N_{\text{ep}}/2 < e < N_{\text{ep}} - 1$	$e = N_{\text{ep}}$
$k \leqslant K/4$	0.8	0.5	0
$K/4 < k \leqslant K/2$	0.6	0.3	0
$K/2 < k \leqslant 3/4K$	0.4	0.2	0
$k > 3/4K$	0.3	0.1	0
Exploration	High	Medium-Low	None

A. *Performance in Single-Target Scenario*

We first consider a scenario with only one target, and the UAVs of the DRN cooperate to reduce the learning time. In such settings, being the mission goal common for all the network, UAVs do not perform a task allocation phase.

In Figs. 4-5, we report the estimated occupancy maps and trajectories during the last episode by considering a single Monte Carlo realization. Rewards were set to optimize the detection of the source, i.e., $\eta = 1$ and $\xi = 0$ (Fig. 4), or for joint detection and mapping, i.e., $\eta = 1$ and $\xi = 0.3$ (Fig. 5). We tested the following configurations with two UAVs: (i) Independent learning (top) where each UAV has and updates its own Q-table; (ii) Centralized learning (middle) where each UAV updates a common Q-table; and (iii) Distributed learning (bottom) where each UAV cooperates by exchanging the Q-values (BestAbs-Q). It is possible to notice that when a centralized or distributed cooperation is performed, all the UAVs successfully complete the mission by following a trajectory reaching the source as a destination point. Contrarily, independent learning can fail when there are NLOS conditions as it happens in Figs. 4 and 5-(top, left).

Figures 6-7 depict the Q-table of Figs. 4-5-bottom. Each map corresponds to a possible action (left, right, up, and down), and each Cartesian coordinate inside each map is a possible UAV state. The actual map is reported in white,

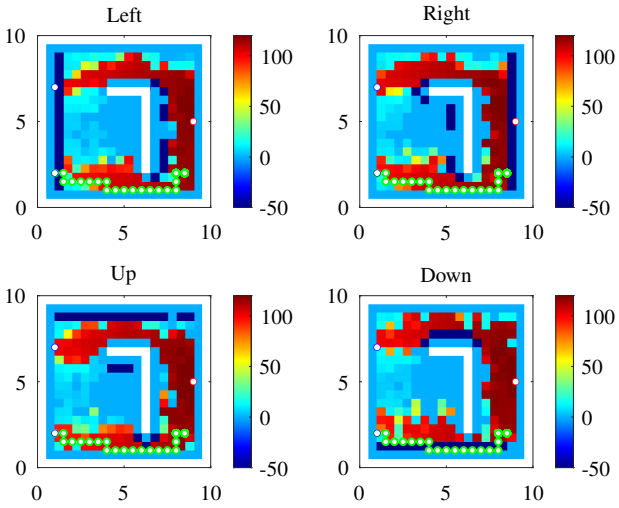


Figure 6: Q-Table for the distributed cooperation with BestAbs-Q with $\eta = 1$, $\xi = 0$. The true map is juxtaposed in white. The initial UAV position are white markers with blue edges; the source is indicated with a white marker with a red contour. The UAV trajectory is displayed with a white marker and green edges.

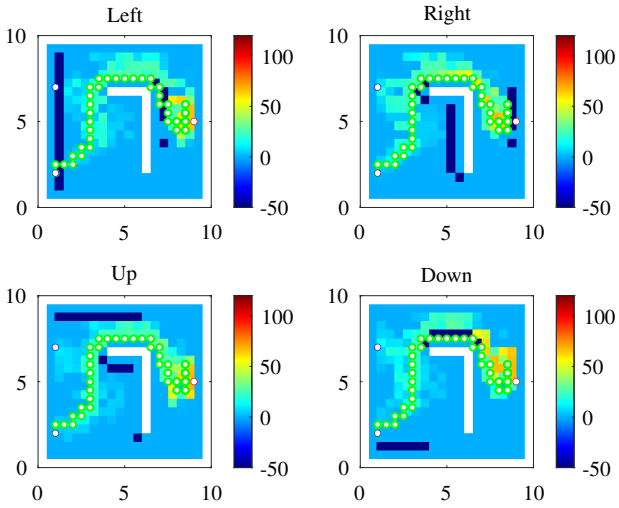


Figure 7: Q-Table for the distributed cooperation with BestAbs-Q with $\eta = 1$, $\xi = 0.3$. The meaning of colors is the same as Fig. 6.

whereas colors represent the intensity of the Q-values, that is, a color tending towards red indicates that a UAV located at the considered cell and choosing the action indicated in the title will receive a good reward based on past experiences. By contrast, colors tending to blue indicate state-action pair that did not lead to high rewards in the previous episodes. As expected, in Fig. 6, UAV trajectories are mainly driven by the target detection, and hence the cells with higher values minimize the distance from the target. Instead, in Fig. 7, the path towards the target leads to lower rewards because the mapping penalizes navigation over the same trajectories in favor of exploring new areas.

In Figs. 8-9 the Q-tables are displayed as functions of the cooperation scheme for $\xi = 0$ and $\xi = 0.3$, respectively. As it can be noticed, independent learners update only a local part of the Q-table, whereas, when cooperation is performed, UAVs can opt to follow paths explored by others and lead to higher rewards.

Now, we investigate the performance averaged over Monte Carlo iterations. To this end, we set the number of simulations to 50.

Figure 10 reports the IS score for two agents and for different cooperation strategies and values of ξ. Continuous and dashed lines refer to a radar (maximum) range of $7.5\,\mathrm{m}$, and $3.75\,\mathrm{m}$, respectively. The image similarity index diminishes over time as the map reconstruction accuracy improves. Instead, the choice of the cooperation strategy does not significantly impact this metric. Finally, a variation of ξ does not impact the mapping performance when the radar range is similar to the environment size, whereas some changes can be perceived when the radar has a scarce illumination capability.

Figure 11 reports the SR as a function of time and for different rewards, that is, for joint detection and mapping rewards, when $\xi = 0.3$. As expected and confirmed by the estimated trajectories, cooperation among the UAVs is beneficial in reducing the time needed to find a single source. Indeed, with cooperation, 80 episodes are sufficient to accomplish the mission in more than 85% of cases.

Figure 12 depicts the SR by varying the mapping weight and changing the cooperation scheme. Through Fig. 12-top, it is confirmed that centralized or distributed cooperation is beneficial for speeding up target detection. Figure 12-bottom instead puts in evidence that when the radar has a limited reading range (e.g., $3.75\,\mathrm{m}$ instead of $7.5\,\mathrm{m}$), the mission cannot be successfully accomplished even in the presence of cooperation, which has almost no impact (the SR is always below the 40% at the end of the mission in all cases). Note also that for $\xi = 1$, Fig. 12-top shows that the detection performance is worsening because the UAVs do not focus only on the primary task but also on mapping. On the other way round, when the reading range (RR) is reduced (Fig. 12-bottom), having $\xi = 1$ helps the UAVs privilege the exploration phase with an increased likelihood to find the target.

In Fig. 13, performance is compared by accounting for the different Q-learning cooperation schemes described in Sec. IV. Notably, cooperation allows for boosting performance, regardless of the choice of a specific algorithm.

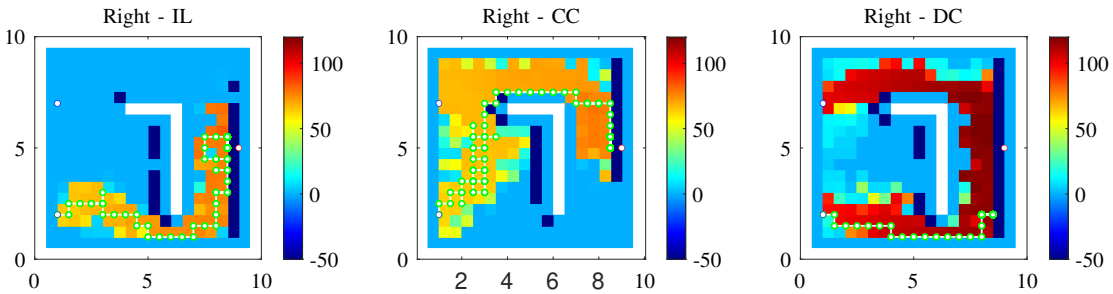


Figure 8: Q-Tables for the right action as a function of the learning strategy with $\eta = 1$, $\xi = 0$. Left: Independent learning (IL); Middle: Centralized cooperative learning (CC); Right: Distributed cooperative learning (DC). The meaning of colors is the same as for Fig. 6.

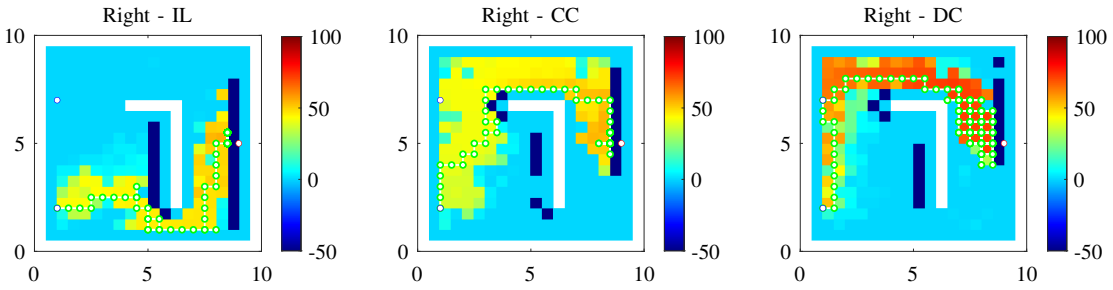


Figure 9: Q-Tables for the right action as a function of the learning strategy with $\eta = 1$, $\xi = 0.3$. Left: Independent learning (IL); Middle: Centralized cooperative learning (CC); Right: Distributed cooperative learning (DC). The meaning of colors is the same as for Fig. 6.

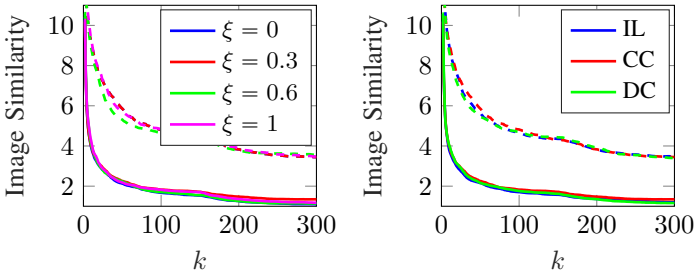


Figure 10: Example of image similarity index computed using (27) for joint detection and mapping for the first two episodes. Continuous and dashed lines refer to a maximum radar reading range of 7.5 m, and 3.75 m, respectively. Left: centralized cooperative learning and different values of ξ; Right: $\xi = 0.3$ and different forms of cooperation.

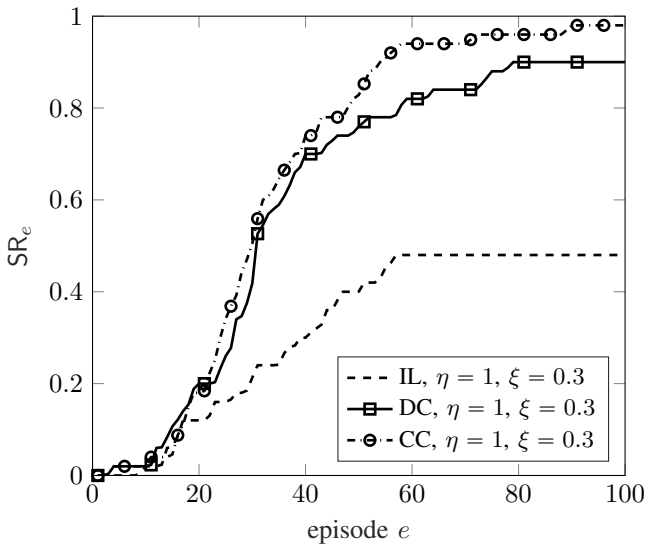


Figure 11: SR for source detection as a function of the number of episodes averaged over Monte Carlo iterations and the number of agents ($N = 2$). Dashed lines: Independent learning (IL), Dot-Dashed lines: Centralized cooperative learning (CC), Continuous lines: Distributed cooperation with BestAbs-Q (DC).

B. Performance in Multi-Task Scenario

We now analyze the task assignment performance in the presence of multiple targets. To this end, we considered the scenario of Fig. 14-top, where $\mathsf{M} = 2$ and $\mathsf{T} = 2$ are placed in three different geometric settings, namely Config. #1, Config. #2 Config. #3. In the first configuration, the UAV initial positions are close to each other, whereas, in the second, they are more spread out in the environment. The third

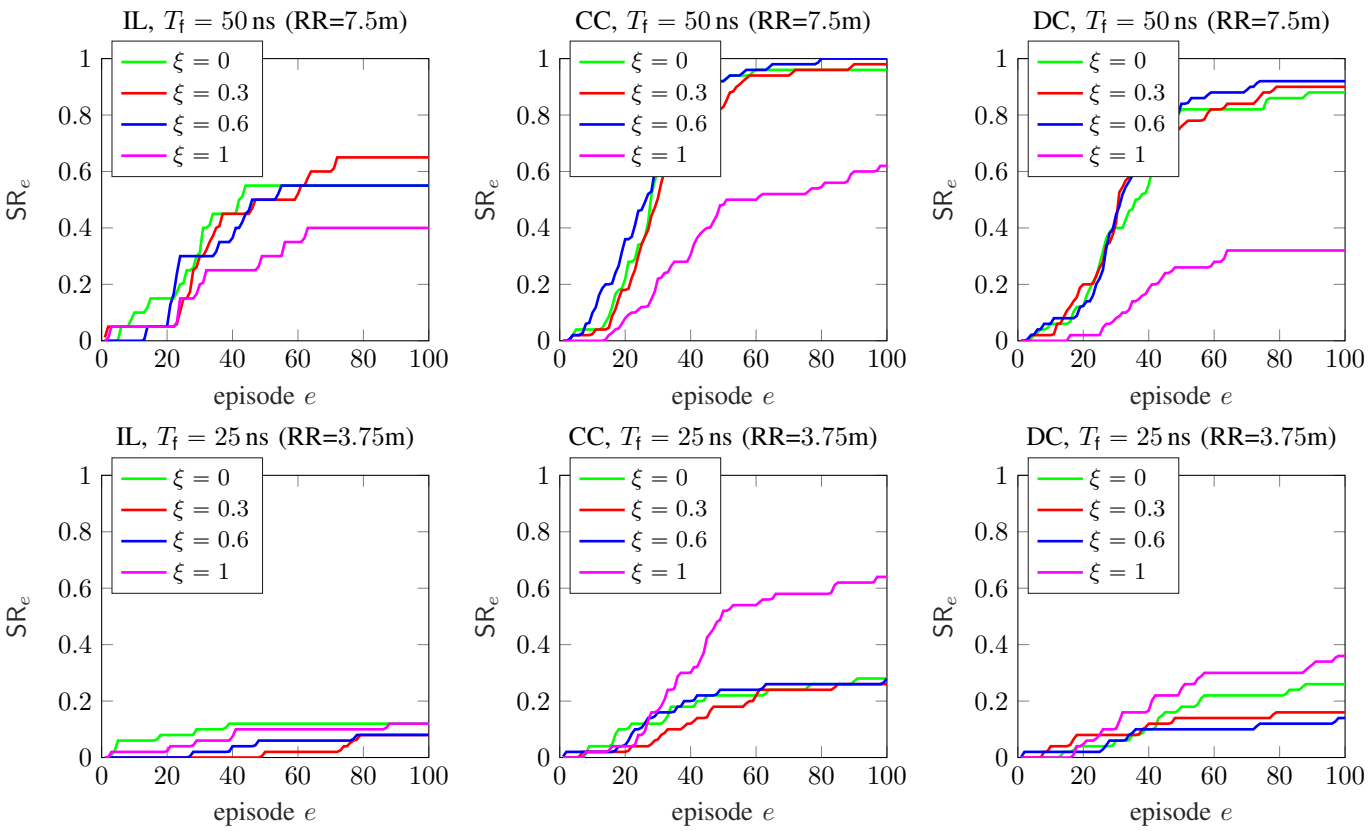


Figure 12: SR for source detection as a function of the number of episodes averaged over Monte Carlo iterations and the number of agents ($N = 2$). The radar reading range was $7.5\,\mathrm{m}$ (top), and $3.75\,\mathrm{m}$ (bottom). The mapping weights are varied as reported in the legend, whereas the weight associated with detection was set to $\eta = 1$.

configuration has been chosen to investigate the performance when one target is close to an obstacle. In our simulations, the task assignment was performed from $k = 1$ till $k = K/2$ with a step of 20. The training duration of the MAB was set to $\tau = 20$ steps.

In Fig. 14-bottom, we plotted the obtained results for different multi-task assignment techniques, that is, (i) cooperation through MAB; (ii) independent by considering the maximum measured RSSI (namely, max − RSSI) and; (iii) independent and random task assignment (namely Random).

The performance shows that the case where the maximum RSSI and the MAB are employed allows for drastically reducing the mission time with respect to the random approach. Moreover, the MAB-based approach avoids having different UAVs sharing the same task in the environment and, consequently, the UAVs can focus on other operations. Nevertheless, the MAB-based approach requires a certain number of training episodes and, thus, a higher complexity.

VI. Conclusions

In this paper, we have investigated the possibility of employing a DRN for indoor scenarios, where targets must be revealed in the shortest possible time, and the environment has to be reconstructed. More specifically, we have investigated both scenarios where cooperation is exploited alternatively for navigation or task assignment. First, we proposed an ad-hoc model for both situations and assessed the performance through extensive simulation analysis. Our results showed that the proposed framework allows for attaining robust performance (in terms of SR and IS) under different settings, which makes DRN a promising solution for solving joint detection and mapping problems.

Appendix A
Mapping

We provide insights about how the mapping reward is evaluated when a multi-antenna radar is exploited by the ith UAV. In this case, for each steering direction, the radar transmits a train of N_{p} pulses, and for each pulse, it collects the backscattering response (e.g., an echo). The time frame is subdivided into N_{bin} bins, and for each time bin, the radar computes the corresponding energy profile. Notably, according to [52], each energy element e_{bs}, referring to a

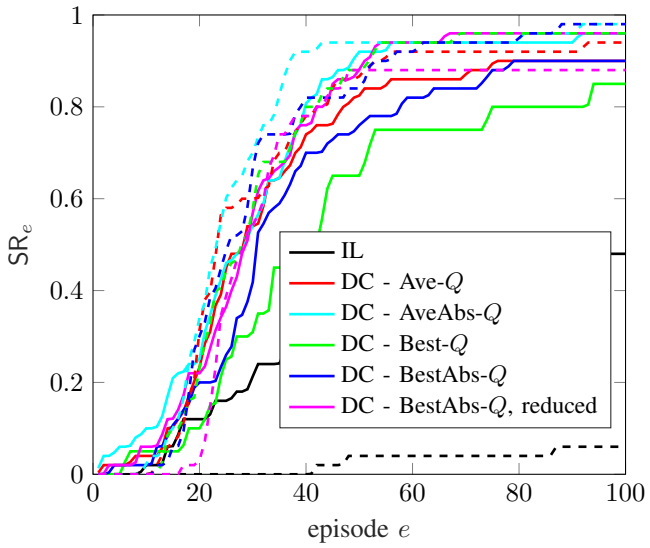


Figure 13: SR for source detection as a function of the number of episodes and the number of agents. We consider a distributed cooperation with different schemes for updating the Q-tables. In all cases, rewards were set considering $\eta = 1, \xi = 0.3$. Solid and dashed lines refer to $\mathsf{M} = 2$ and $\mathsf{M} = 4$, respectively.

specific steering direction b and a specific time bin s, is expressed with

$$e_{i,bs} = \sum_{n=0}^{N_\mathsf{p}-1} \int_{(s-1)\,T_\mathsf{ED}}^{s\,T_\mathsf{ED}} y_i^2(t + nT_\mathsf{f}, \theta_b)\, dt\,, \tag{29}$$

where N_p is the number of pulses transmitted for each steering direction, θ_b is the bth steering direction, $T_\mathsf{ED} \approx 1/W$ is the duration of the energy bin, $y_i(t)$ is the band-pass filtered version of the signal, and T_f is the time frame. Starting from (29), it is possible to write a *Range-Angle* matrix given by

$$\mathbf{e}_i = \begin{array}{c} \begin{array}{cccc} s=1 & s=2 & \dots & s=N_\mathsf{bin} \end{array} \\ \begin{pmatrix} e_{i,11} & e_{i,12} & \dots & e_{i,1N_\mathsf{bin}} \\ e_{i,21} & e_{i,22} & \dots & e_{i,2N_\mathsf{bin}} \\ \vdots & \vdots & \vdots & \vdots \\ e_{i,N_\mathsf{steer}\,1} & e_{i,N_\mathsf{steer}\,2} & \dots & e_{i,N_\mathsf{steer}\,N_\mathsf{bin}} \end{pmatrix} \end{array} \begin{array}{l} b=1 \\ b=2 \\ \vdots \\ b=N_\mathsf{steer} \end{array} . \tag{30}$$

A possible approach to estimate a map of the environment is to use an OG algorithm that, given these observations and by operating cell-by-cell, estimates the log-odd of occupancy for the jth cell as

$$\ell_{i,k}(m_j) = \log\left(\frac{b_{i,k}(m_j)}{1 - b_{i,k}(m_j)}\right), \tag{31}$$

where $b_{i,k}(m_j)$ is the belief of the occupancy state of the j th cell computed by the i th UAV at time instant k. To this end, the algorithm proceeds in two main steps [52]:

- *Initialization:* The map is initialized as $\ell_{i,0}(m_j) = \log\left(\frac{b_{i,0}(m_j)}{1-b_{i,0}(m_j)}\right)$ with $b_{i,0}(m_j) = 0.5$, corresponding to a complete uncertainty, $\forall j = 1, 2, \dots, N_\mathsf{cell}$.
- *Measurement and Scan Vector Generation:* A new *Range-Angle* matrix. as in (30), is acquired and each row $\mathbf{e}_{i;b} = [e_{i,b1}, e_{i,b2}, \dots, e_{i,bs}, \dots, e_{i,bN_\mathsf{bin}}]$ is compared with a threshold $0 < \gamma \leqslant 1$. The distance corresponding to the first element exceeding $\gamma \max(\mathbf{e}_{i;b})$ is saved in a vector $\mathbf{r}_{i,k}$. At the same time, all the steering angles are collected into the angle vector $\boldsymbol{\phi} = [\phi_1, \phi_2, \dots, \phi_{N_\mathsf{steer}}]$. The final scan vector at time instant k is given by $\mathbf{s}_{i,k} = \left[\mathbf{r}_{i,k}^T,\, \boldsymbol{\phi}^T\right]$.
- *Log-Odd Update:* Starting from $\mathbf{s}_k$, the beliefs are updated following a classic occupancy grid algorithm [83], i.e.,

$$\ell_{i,k}(m_j) = \log\left(\frac{p(\mathbf{s}_k|m_j = 1)}{p(\mathbf{s}_k|m_j = 0)}\right) + \ell_{i,k-1}(m_i) \quad \forall\, i = 1, \dots, N_\mathsf{cell} \tag{32}$$

where $p(\mathbf{s}_k|m_j = 1)$ $(p(\mathbf{s}_k|m_j = 0))$ is the likelihood function considering the current scan $\mathbf{s}_k$ given the presence of an occupied (free) cell in m_j.

Note that (32) assumes that each cell is independent of all the others (including the adjacent cells).

REFERENCES

[1] V. Chernyak, "Multisite radar systems composed of MIMO radars," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 29, no. 12, pp. 28–37, 2014.
[2] V. Kumar, D. Rus, and S. Singh, "Robot and sensor networks for first responders," *IEEE Pervasive Comput.*, vol. 3, no. 4, pp. 24–33, 2004.
[3] E. Paolini *et al.*, "Localization capability of cooperative anti-intruder radar systems," *EURASIP J. Adv. Signal Process.*, vol. 2008, pp. 1–14, 2008.
[4] A. Guerra *et al.*, "Networks of UAVs of low-complexity for time-critical localization," *arXiv preprint arXiv:2108.13181*, 2021.
[5] Y. Zeng, R. Zhang, and T. J. Lim, "Wireless communications with unmanned aerial vehicles: Opportunities and challenges," *IEEE Commun. Mag.*, vol. 54, no. 5, pp. 36–42, 2016.
[6] H. Yao, R. Qin, and X. Chen, "Unmanned aerial vehicle for remote sensing applications—A review," *Remote Sensing*, vol. 11, no. 12, p. 1443, 2019.
[7] A. Guerra, D. Dardari, and P. M. Djuric, "Dynamic radar networks of UAVs: A tutorial overview and tracking performance comparison with terrestrial radar networks," *IEEE Veh. Technol. Mag.*, vol. 15, no. 2, pp. 113–120, 2020.
[8] A. Guerra, D. Dardari, and P. M. Djuric, "Dynamic radar network of UAVs: A joint navigation and tracking approach," *IEEE Access*, vol. 8, pp. 1–1, 2020.
[9] A. Rahmati *et al.*, "Dynamic interference management for UAV-assisted wireless networks," *IEEE Trans. Wireless Commun.*, pp. 1–1, 2021.
[10] Y. Liu *et al.*, "Distributed 3D relative localization of UAVs," *IEEE Trans. Veh. Technol.*, vol. 69, no. 10, pp. 11 756–11 770, 2020.
[11] E. Testi, E. Favarelli, and A. Giorgetti, "Reinforcement learning for connected autonomous vehicle localization via UAVs," in *Proc. IEEE Int. Workshop Metro. Agri. For. (MetroAgriFor)*, 2020, pp. 13–17.
[12] E. Staudinger *et al.*, "The role of time in a robotic swarm: A joint view on communications, localization, and sensing," *IEEE Commun. Mag.*, vol. 59, no. 2, pp. 98–104, 2021.
[13] W. Khawaja *et al.*, "A survey of air-to-ground propagation channel modeling for unmanned aerial vehicles," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2361–2391, 2019.

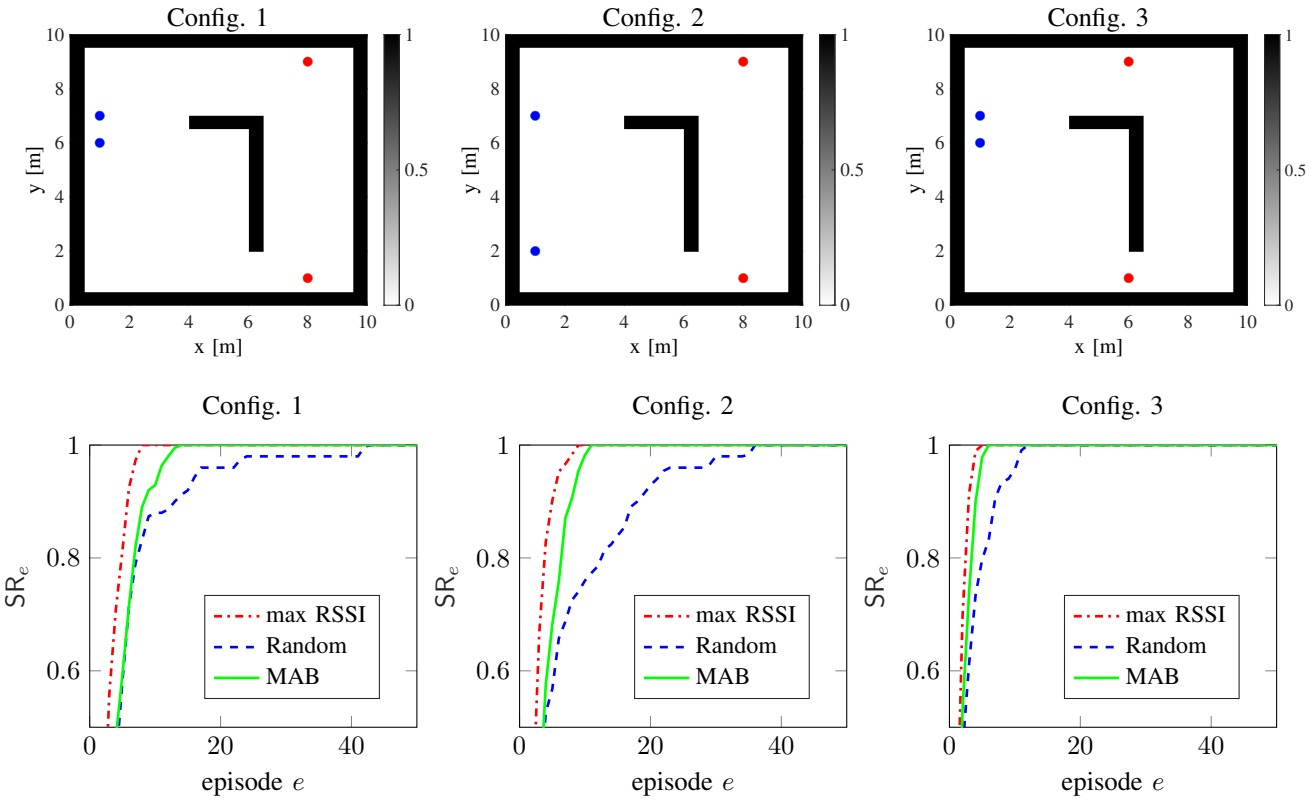


Figure 14: SR as a function of the task assignment strategy and geometric configuration.

[14] M. M. U. Chowdhury, F. Erden, and I. Guvenc, "RSS-based Q-learning for indoor UAV navigation," in *Proc. Military Commun. Conf. (MILCOM)*, 2019, pp. 121–126.

[15] S. Mignardi *et al.*, "Optimizing beam selection and resource allocation in UAV-aided vehicular networks," in *Proc. Joint European Conf. Netw. Commun. & 6G Summit (EuCNC/6G Summit)*. IEEE, 2022, pp. 184–189.

[16] X. Liu *et al.*, "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, 2019.

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

[18] S. Zhang *et al.*, "Self-aware swarm navigation in autonomous exploration missions," *Proc. IEEE*, vol. 108, no. 7, pp. 1168–1195, 2020.

[19] Z. Lou, A. Elzanaty, and M.-S. Alouini, "Green tethered UAVs for EMF-aware cellular networks," *IEEE Trans. Green Commun. Netw.*, vol. 5, no. 4, pp. 1697–1711, 2021.

[20] H. Bayerlein, P. De Kerret, and D. Gesbert, "Trajectory optimization for autonomous flying base station via reinforcement learning," in *Proc. Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, 2018, pp. 1–5.

[21] M. Theile *et al.*, "Uav coverage path planning under varying power constraints using deep reinforcement learning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Sys. (IROS)*, 2020, pp. 1444–1449.

[22] Q. Wang *et al.*, "Multi-UAV dynamic wireless networking with deep reinforcement learning," *IEEE Commun. Lett.*, vol. 23, no. 12, pp. 2243–2246, 2019.

[23] T. Wang *et al.*, "A reinforcement learning approach for UAV target searching and tracking," *Multimedia Tools and Applications*, vol. 78, no. 4, pp. 4347–4364, 2019.

[24] A. M. Ahmed *et al.*, "A reinforcement learning based approach for multi-target detection in massive MIMO radar," *IEEE Trans. Aerosp. Elect. Syst.*, vol. 57, no. 5, pp. 2622–2636, 2021.

[25] Y.-J. Chen, D.-K. Chang, and C. Zhang, "Autonomous tracking using a swarm of UAVs: A constrained multi-agent reinforcement learning approach," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 13 702–13 717, 2020.

[26] J. Moon *et al.*, "Deep reinforcement learning multi-UAV trajectory control for target tracking," *IEEE Internet Things J.*, vol. 8, no. 20, pp. 15 441–15 455, 2021.

[27] Z. Xia *et al.*, "Multi-agent reinforcement learning aided intelligent UAV swarm for target tracking," *IEEE Trans. Veh. Technol.*, vol. 71, no. 1, pp. 931–945, 2021.

[28] Y. Zeng *et al.*, "Simultaneous navigation and radio mapping for cellular-connected UAV with deep reinforcement learning," *IEEE Trans. Wireless Commun.*, vol. 20, no. 7, pp. 4205–4220, 2021.

[29] L. P. Osco *et al.*, "A review on deep learning in UAV remote sensing," *Int. J. of Applied Earth Observation and Geoinformation*, vol. 102, p. 102456, 2021.

[30] C. Wang *et al.*, "Autonomous navigation of UAVs in large-scale complex environments: A deep reinforcement learning approach," *IEEE Trans. Veh. Technol.*, vol. 68, no. 3, pp. 2124–2136, 2019.

[31] O. Esrafilian, H. Bayerlein, and D. Gesbert, "Model-aided deep reinforcement learning for sample-efficient UAV trajectory design in IoT networks," *arXiv preprint arXiv:2104.10403*, 2021.

[32] H. Bayerlein *et al.*, "Multi-UAV path planning for wireless data harvesting with deep reinforcement learning," *IEEE Open J. Commun. Soc.*, vol. 2, pp. 1171–1187, 2021.

[33] M. M. U. Chowdhury *et al.*, "3-D trajectory optimization in uav-assisted cellular networks considering antenna radiation pattern and backhaul constraint," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 56, no. 5, pp. 3735–3750, 2020.

[34] J. Cui, Y. Liu, and A. Nallanathan, "Multi-agent reinforcement learning-based resource allocation for UAV networks," *IEEE Trans. Wireless Commun.*, vol. 19, no. 2, pp. 729–743, 2020.

[35] J. Hu, H. Zhang, and L. Song, "Reinforcement learning for decentral-

ized trajectory design in cellular UAV networks with sense-and-send protocol," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 6177–6189, 2018.
[36] X. Liu, Y. Liu, and Y. Chen, "Reinforcement learning in multiple-UAV networks: Deployment and movement design," *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 8036–8049, 2019.
[37] F. Venturini *et al.*, "Distributed reinforcement learning for flexible and efficient uav swarm control," *IEEE Trans. Cogn. Commun. Netw.*, vol. 7, no. 3, pp. 955–969, 2021.
[38] L. Wang *et al.*, "Reinforcement learning-based waveform optimization for MIMO multi-target detection," in *Proc. Asilomar Conf. Signals, Syst., and Comput.*, 2018, pp. 1329–1333.
[39] Z. Xiong *et al.*, "Deep reinforcement learning for mobile 5G and beyond: Fundamentals, applications, and challenges," *IEEE Veh. Technol. Mag.*, vol. 14, no. 2, pp. 44–52, 2019.
[40] C. Wang *et al.*, "Autonomous navigation of uavs in large-scale complex environments: A deep reinforcement learning approach," *IEEE Trans. Veh. Technol.*, vol. 68, no. 3, pp. 2124–2136, 2019.
[41] D. Lee *et al.*, "Optimization for reinforcement learning: From a single agent to cooperative agents," *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 123–135, 2020.
[42] M.-A. Lahmeri, M. A. Kishk, and M.-S. Alouini, "Artificial intelligence for uav-enabled wireless networks: A survey," *IEEE Open J. Commun. Soc.*, vol. 2, pp. 1015–1040, 2021.
[43] Y. Yan *et al.*, "Relative distributed formation and obstacle avoidance with multi-agent reinforcement learning," *arXiv preprint arXiv:2111.07334*, 2021.
[44] P. Valente Klaine *et al.*, "Backhaul aware user-specific cell association using Q-learning," *IEEE Trans. Wireless Commun.*, vol. 18, no. 7, pp. 3528–3541, 2019.
[45] K. Arulkumaran *et al.*, "Deep reinforcement learning: A brief survey," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 26–38, 2017.
[46] R. M. Annasamy and K. Sycara, "Towards better interpretability in deep q-networks," in *Proc. AAAI Conf. Artificial Intell.*, vol. 33, no. 01, 2019, pp. 4561–4569.
[47] M. Dorigo, G. Theraulaz, and V. Trianni, "Swarm robotics: past, present, and future [point of view]," *Proc. IEEE*, vol. 109, no. 7, pp. 1152–1165, 2021.
[48] M. Ahmadabadi and M. Asadpour, "Expertness based cooperative *q*-learning," *IEEE Trans. Syst., Man, Cybern. B. Cybern.*, vol. 32, no. 1, pp. 66–76, 2002.
[49] D. J. Ornia and M. Mazo Jr, "Event-based communication in multi-agent distributed Q-learning," *arXiv preprint arXiv:2109.01417*, 2021.
[50] J. Liu *et al.*, "Distributed Q-learning aided uplink grant-free NOMA for massive machine-type communications," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 7, pp. 2029–2041, 2021.
[51] J. Su *et al.*, "An SCMA-based decoupled distributed Q-learning random access scheme for machine-type communication," *IEEE Wireless Commun. Lett.*, vol. 10, no. 8, pp. 1737–1741, 2021.
[52] A. Guerra *et al.*, "Occupancy grid mapping for personal radar applications," in *Proc. IEEE Stat. Signal Process. Work. (SSP)*, 2018, pp. 766–770.
[53] P. Flak, "Drone detection sensor with continuous 2.4 GHz ISM band coverage based on cost-effective SDR platform," *IEEE Access*, vol. 9, pp. 114 574–114 586, 2021.
[54] F. Guidi *et al.*, "Joint energy detection and massive array design for localization and mapping," *IEEE Trans. Wireless Commun.*, vol. 16, no. 3, pp. 1359–1371, 2016.
[55] F. Liu *et al.*, "Integrated sensing and communications: Towards dual-functional wireless networks for 6g and beyond," *IEEE J. Sel. Areas Commun.*, 2022.
[56] D. Dardari, E. Falletti, and M. Luise, *Satellite and terrestrial radio positioning techniques: a signal processing perspective*. Academic Press, 2012.
[57] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.
[58] N. Patwari *et al.*, "Locating the nodes: cooperative localization in wireless sensor networks," *IEEE Signal Process. Mag.*, vol. 22, no. 4, pp. 54–69, 2005.
[59] H. Obeidat *et al.*, "An indoor path loss prediction model using wall correction factors for wireless local area network and 5g indoor networks," *Radio Science*, vol. 53, no. 4, pp. 544–564, 2018.
[60] M. A. Richards *et al.*, *Principles of Modern Radar*. Citeseer, 2010, vol. 1.
[61] C. J. C. H. Watkins, "Learning from delayed rewards," 1989.
[62] A. Guerra *et al.*, "Real-time learning for THz radar mapping and UAV control," in *Proc. IEEE Conf. on Autonomous Systems (ICAS)*, 2021.
[63] N. Mafi, F. Abtahi, and I. Fasel, "Information theoretic reward shaping for curiosity driven learning in POMDPs," in *Proc. IEEE Int. Conf. Develop. Learning (ICDL)*, vol. 2, 2011, pp. 1–7.
[64] I. Fasel *et al.*, "Intrinsically motivated information foraging," in *Proc. IEEE 9th Int. Conf. Develop. Learning*, 2010, pp. 101–107.
[65] T. Brys *et al.*, "Multi-objectivization of reinforcement learning problems by reward shaping," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, 2014, pp. 2315–2322.
[66] C. F. Hayes *et al.*, "A practical guide to multi-objective reinforcement learning and planning," *Autonomous Agents and Multi-Agent Systems*, vol. 36, no. 1, pp. 1–59, 2022.
[67] G. Fontanesi, A. Zhu, and H. Ahmadi, "Outage analysis for millimeter-wave fronthaul link of uav-aided wireless networks," *IEEE Access*, vol. 8, pp. 111 693–111 706, 2020.
[68] B. H. Abed-Alguni *et al.*, "A comparison study of cooperative q-learning algorithms for independent learners," *Int. J. Artif. Intell.*, vol. 14, no. 1, pp. 71–93, 2016.
[69] W. Xia *et al.*, "Multi-armed bandit-based client scheduling for federated learning," *IEEE Trans. Wireless Commun.*, vol. 19, no. 11, pp. 7108–7123, 2020.
[70] X. Xu, M. Tao, and C. Shen, "Collaborative multi-agent multi-armed bandit learning for small-cell caching," *IEEE Trans. Wireless Commun.*, vol. 19, no. 4, pp. 2570–2585, 2020.
[71] A. Carpentier *et al.*, "Upper-confidence-bound algorithms for active learning in multi-armed bandits," in *Proc. Int. Conf. on Algorithmic Learning Th.* Springer, 2011, pp. 189–203.
[72] V. Kuleshov and D. Precup, "Algorithms for multi-armed bandit problems," *arXiv preprint arXiv:1402.6028*, 2014.
[73] H.-S. Jo *et al.*, "Heterogeneous cellular networks with flexible cell association: A comprehensive downlink SINR analysis," *IEEE Trans. Wireless Commun.*, vol. 11, no. 10, pp. 3484–3495, 2012.
[74] A. Birk and S. Carpin, "Merging occupancy grid maps from multiple robots," *Proc. IEEE*, vol. 94, no. 7, pp. 1384–1397, 2006.
[75] S. Lim, H. Yu, and H. Lee, "Optimal tethered-UAV deployment in A2G communication networks: Multi-agent Q-learning approach," *IEEE Internet Things J.*, pp. 1–1, 2022.
[76] C. J. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, no. 3-4, pp. 279–292, 1992.
[77] M. L. Littman and C. Szepesvári, "A generalized reinforcement-learning model: Convergence and applications," in *Proc. Int. Conf. Machine Learning ICML*, vol. 96. Citeseer, 1996, pp. 310–318.
[78] D. P. Bertsekas and J. N. Tsitsiklis, "Neuro-dynamic programming: an overview," in *Proc. IEEE Conf. Decision Control*, vol. 1. IEEE, 1995, pp. 560–564.
[79] T. Jaakkola, M. I. Jordan, and S. P. Singh, "On the convergence of stochastic iterative dynamic programming algorithms," *Neural Comput.*, vol. 6, no. 6, pp. 1185–1201, 1994.
[80] E. Even-Dar, Y. Mansour, and P. Bartlett, "Learning rates for Q-learning." *J. Machine Learning Research*, vol. 5, no. 1, 2003.
[81] T. Kemmerich and H. K. Büning, "A convergent multiagent reinforcement learning approach for a subclass of cooperative stochastic games," in *Int. Workshop Adapt. Learning Agents*. Springer, 2011, pp. 37–53.
[82] H.-E. Kim and H.-S. Ahn, "Convergence of multiagent Q-learning: Multi action replay process approach," in *IEEE Int. Symp. Intell. Control*, 2010, pp. 789–794.
[83] S. Thrun, "Probabilistic robotics," *Commun. of the ACM*, vol. 45, no. 3, pp. 52–57, 2002.