Multi-Agent Q-Learning in UAV Networks for Target Detection and Indoor Mapping

Multi-Agent Q-Learning in UAV Networks for Target Detection and Indoor Mapping

Anna Guerra
DEI
University of Bologna
Cesena, Italy
anna.guerra3@unibo.it

Francesco Guidi
IEIIT
National Research Council of Italy
Bologna, Italy
francesco.guidi@ieiit.cnr.it

Davide Dardari
DEI
University of Bologna
Cesena, Italy
davide.dardari@unibo.it

Petar M. Djurić
ECE
Stony Brook University
New York, USA
petar.djuric@stonybrook.edu

***Abstract*—We consider a network of unmanned aerial vehicles (UAVs) for a search-and-rescue operations involving both detection of multiple targets and mapping of environment, where the learning time is limited. One possibility for accomplishing the goal while guaranteeing short learning time is to employ cooperation among UAVs. With this objective, we adopt a multi-agent Q-learning algorithm that allows the UAVs to learn a suitable navigation policy in real-time in order to complete a mission within a fixed time frame. The obtained results demonstrate that proper combination of the information gathered by the UAVs allows for an accelerated learning process.**

I. Introduction

UAVs are autonomous flying agents capable of performing multiple tasks, and they are usually deployed to carry out missions that are too risky for human operators. For example, UAVs have played a central role in emergency situations in hazardous environments, for post natural disasters, or for search-and-rescue operations. In such events, UAVs have been used as a temporary network infrastructure for localization, communications, and for delivering items [1]–[3].

Commonly, a major challenge of UAV networks is the limited time the UAVs have to complete tasks because of the need that their batteries be frequently recharged. Unlike terrestrial sensors, all tasks and navigation must be optimized as not to waste time flying over areas of little interest from the mission perspective [4]. A possible solution to shorten the mission time is to leverage over UAV cooperation that can also be orchestrated by the higher layers of the communication infrastructure, e.g., the edges. In this direction, preliminary studies foresee that the sixth generation (6G) wireless networks will integrate UAV-based flying networks within the cellular infrastructure, facilitating the connection with edges that are characterized by a higher reliability [5].

In this paper, we aim to study a navigation approach for multi-target detection (primary task) and for improving the ambient awareness by reproducing an occupancy map of the environment [4]. Differently from classical offline optimization, the UAVs discover the environment in real-time and in absence of any pre-defined fixed waypoint. Moreover, we assume lack of a system dynamic model so that we cannot rely on classical control optimization, i.e., model predictive control. In this sense, machine learning (ML) can help in acquiring a knowledge of the model through experience. To that end, we adopt reinforcement learning (RL), which is based on the "*trial-and-error*" philosophy that allows to choose actions in order to maximize the sum of the discounted rewards over the future [3], [6]–[8]. In such settings, UAV navigation is driven by the balance between "exploration" and "exploitation". Specifically, the UAVs can *exploit* the acquired knowledge, but they can also decide to perform random actions to *explore* new areas. As an example, Q-learning is often used in practical problems where a grid-world representation of the environment is possible. States and actions constitute a Q-table that is updated at each time instant according to the received rewards [9].

In contrast with our previous work in [8], we consider a network of cooperative UAVs where each agent[1] independently infers a state of the environment based on its own measurements and receives a local reward through interactions (i.e., agent–independent control). Then, the UAVs share information by updating the same Q-table in a common digital space (e.g., the edge). In this way, the UAVs take actions based on a "global" shared knowledge, and the information exchange overhead is reduced. Operating like this, it is possible to reduce the overall learning time, thus improving the network's performance.

In the sequel, we describe the considered multi-agent Q-learning approach and how rewards are defined. Through numerical results we demonstrate that the cooperation among agents allows for reduction of the required learning time.

II. Multi-Agent Q-learning

A. Problem Formulation

The scenario of this paper considers a network of M UAVs that explores an indoor environment. More specifically, such agents cooperate in order to achieve two goals: *(i)* detection of targets that can be, for example, cooperative users that need to be rescued or hidden malicious targets whose unwanted communication is sniffed within a certain radiofrequency

P.M.D. thanks the support of NSF under Award 2021002. This work has also received funding from the EU's H2020 research and innovation programme under the Marie Skłodowska-Curie project AirSens (no. 793581).

[1]In the sequel, the terms "agent" and "UAV" are interchangeably used.

band; *(ii)* estimation of an occupancy map of the explored area. In this context, each UAV has to select the best trajectory (i.e., sequence of actions) in order to maximize the overall cumulative expected reward gained by accomplishing the two aforementioned tasks in a fixed mission time T_{M}. In our case, T_{M} is given by the number of discrete time instants of each episode K multiplied by the number of training episodes E, i.e., $T_{\mathsf{M}} = K\,E$.

A model of the environment is not available and, thus, it should be learnt through experience and interactions with the environment itself. Because of the time-critical nature of the mission, a hidden goal of the network is to speed up the learning process while guaranteeing a good mapping accuracy and a high detection probability. This is of particular importance in emergency or post-disaster situations.

Each UAV is a system comprising two estimation processes. A first *state estimation phase* allows the unknowns of the state to be inferred based on upcoming measurements. We assume that each UAV is equipped with RF sensors to collect measurements for detection and mapping and with receivers to process the information in an adequate manner. For example, in our case, we implement an occupancy grid (OG) for mapping and use a detection module to determine if a target is present. The second step is a *navigation phase* where the UAVs decide where to go next based on the global acquired knowledge, learnt through experience. Both the *state estimation* and *navigation phases* run on-board UAVs so that the UAVs are capable to make their own independent decisions. On the other side, instantaneous rewards, actions and states are shared by the UAVs via an edge to form a global repository of knowledge that can be accessed by all the UAVs to take a proper navigation decision.

In the rest of the paper, we focus on the capability to take informative navigation decisions according to multi-agent Q-learning. More details on state estimation can be found in [8].

B. A Q-learning Algorithm for UAV Navigation

As previously stated, we consider a trajectory that is chosen by the UAVs to maximize the target detection and mapping accuracy subject to the mission time T_{M} and collision avoidance. This optimization problem can be properly formulated by a Markov decision process (MDP), which is defined by a tuple containing 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 state $\mathbf{s}_{k+1}$ at time instant $k+1$. More specifically, we define $\mathbf{s}_{i,k} \in \mathcal{S}$ as the vector containing the states of the ith UAV at time instant k, that is, the ith UAV position, the map of the environment and a detection vector, i.e.,

$$\mathbf{s}_{i,k} = [\mathbf{p}_{i,k},\, \mathbf{m}_k,\, \mathbf{t}_k]^{\mathsf{T}} \tag{1}$$

with $i \in \mathcal{M} = \{1, 2, \ldots, M\}$, where $\mathbf{p}_{i,k} = [x_{i,k},\, y_{i,k}]^{\mathsf{T}} \in \mathbb{R}^2$ 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 that represent the map, and $\mathbf{t}_k \in \mathbb{B}^N$ is the target vector (equal to one if the target is present and zero otherwise) with N, being the number of targets. As the environment is considered stationary, it is $\mathbf{t}_k = \mathbf{t}$ and $\mathbf{m}_k = \mathbf{m}, \forall k$, with $\mathbf{m} = [m_1, \ldots, m_j, \ldots, m_{N_{\mathsf{cell}}}]^{\mathsf{T}}$, containing the occupancy value of each cell, i.e., $m_j \in \mathbb{B}$, and N_{cell} being the total number of cells. According to the above definitions, the state space dimension for each UAV is $\mathcal{S} = \mathbb{R}^2 \times \mathbb{B}^{N_{\mathsf{cell}}} \times \mathbb{B}^N$.

The UAV navigation actions are $\mathbf{a}_{i,k} = [\Delta x_{i,k}, \Delta y_{i,k}]^{\mathsf{T}}$ according to four possible actions belonging to the action space $\mathcal{A} = \Big\{[\Delta,\, 0],\, [-\Delta,\, 0],\, [0,\, \Delta],\, [0,\, -\Delta]\Big\}$ with Δ being the spatial step. The actions are chosen according to a specific policy $\pi_i\left(\mathbf{a}_{i,k}|\mathbf{s}_{i,k}\right)$ for the ith UAV. The optimal policy selects actions that maximize a value function by

$$\pi_i^*\left(\mathbf{a}_{i,k}|\mathbf{s}_{i,k}\right) = \underset{\mathbf{a}_{i,k}}{\arg\max}\, Q_{\pi_i}\left(\mathbf{s}_{i,k}, \mathbf{a}_{i,k}\right) \tag{2}$$

with $i \in \mathcal{M}$, and where $Q_{\pi_i}(\cdot)$ is the expected sum of discounted rewards over all possible policies (namely, Q-function). For multi–agent set–ups, π_i indicates that each UAV policy accounts for the shared information and comes up to a joint strategy by considering the joint policies at each UAV, that is, $\pi = \{\pi_1, \pi_2, \ldots, \pi_M\}$.

As previously mentioned, we focus on Q-learning, where the policy is learnt during run-time, i.e., simultaneously as the UAV navigates. It is a model-free tabular algorithm whose main steps are reported in [10], where the possibility of choosing a random action with probability ϵ (ϵ-greedy approach) is also included [8], [10]. Even if an update of the return is made at each time step, a number of trials (episodes) is required to

Algorithm 1: Q-Learning - Single Episode.

Parameters: Set the learning parameters $(\gamma, \alpha, \epsilon)$, the mission time $T_{\mathsf{M}} = K\,E$, the number of UAVs M and the number of episodes E;
Init.: Initialize $\mathbf{s}_{i,0}$, $\forall\, i \in \mathcal{M} = \{1, \ldots, M\}$ and $k = 0$;
if $e = 0$ **then**
  Set all the elements of the Q-table to 0;
else
  Inherit the Q-table from the previous episode, i.e., $Q_e = Q_{e-1}$.
end
while $k < K$ **do**
  while $i < M$ **do**
    Generate a random value $\epsilon^\star$;
    if $\epsilon^\star < \epsilon$ **then**
      Choose a random action $\mathbf{a}_{i,k} \in \mathcal{A}$ *(exploration)*.
    else
      Choose a greedy action $\mathbf{a}_{i,k} \in \mathcal{A}$ that corresponds to the maximum Q-value in $Q(\mathbf{s}_{i,k}, :)$; *(exploitation)*.
    end
    UAV moves to the new state, collects rewards, and updates the Q-table using (3).
  end
end

find the proper trajectory. Setback are possible from situations that have not been learned before. This problem can be in part mitigated by the presence of more than one agent that explore different areas of the environment, and accelerate the learning.

When the Q-function is represented by a Q-table, the update rule for the ith agent is [10]

$$Q(\mathbf{s}_{i,k}, \mathbf{a}_{i,k}) \leftarrow Q(\mathbf{s}_{i,k}, \mathbf{a}_{i,k}) + \alpha \left[r_{i,k+1} + \gamma \max_{\mathbf{a}} Q(\mathbf{s}_{i,k}, \mathbf{a}) - Q(\mathbf{s}_{i,k}, \mathbf{a}_{i,k}) \right], \quad (3)$$

where $r_{i,k+1}$ is the received instantaneous reward, α is the learning rate, and γ is the discount factor. Note that while actions are selected on-board by each UAV, the Q-table is shared among all of them (e.g., through an edge), and this allows the UAVs to take more informative decisions.

III. Reward Definition

Rewards are usually categorized in two groups, that is, extrinsic and intrinsic rewards [11], [12]. The first ones are usually task-specific and they associate a state-action pair into a real-valued reward. On the other side, intrinsic rewards only indirectly depend on the world state through the beliefs of the UAV about such a state [11]. In our scenario, target detection represents the UAVs extrinsic reward, whereas mapping can be well described by intrinsic rewards. Obviously, their proper joint combination allows to accelerate the overall learning procedure. According to the considerations in [8], we can write

$$r_{i,k+1} = \eta^{(\mathsf{int})}\, r^{(\mathsf{int})}_{i,k+1} + \eta^{(\mathsf{ext})}\, r^{(\mathsf{ext})}_{i,k+1}, \quad (4)$$

where $\eta^{(\mathsf{int})}$ and $\eta^{(\mathsf{ext})}$ are weight coefficients, and

$$r^{(\mathsf{int})}_{i,k+1} = r^{(\mathsf{c})}_{i,k+1} + r^{(\mathsf{m})}_{i,k+1}, \qquad r^{(\mathsf{ext})}_{i,k+1} = r^{(\mathsf{d})}_{i,k+1}, \quad (5)$$

are the intrinsic reward, $r^{(\mathsf{int})}_{i,k+1}$, used for obtaining a sufficient knowledge of the surrounding environment, and an extrinsic reward, $r^{(\mathsf{ext})}_{i,k+1}$, for the considered UAV task. The rewards $r^{(\mathsf{c})}_{i,k+1}$ and $r^{(\mathsf{d})}_{i,k+1}$ are defined in [8]. More specifically, $r^{(\mathsf{d})}_{i,k+1} = \lambda_{i,k}/\lambda_{\max}$ accounts for the detection performance, where $\lambda_{i,k}$ depends on the total signal-to-noise ratio (SNR) considering all the targets present in the environment measured at the ith UAV at time instant k. The normalization factor, $\lambda_{\max}$, is the maximum SNR computed as if the UAV was in an adjacent cell of the target. Note that $\lambda_{i,k}$ is implicitly mapped into the detection performance according to the analysis in [13], where the detection probability is expressed as $\mathcal{Q}_h(\sqrt{\lambda_{i,k}}, \sqrt{\xi})$, with $\mathcal{Q}_h$ being the Marcum's $\mathcal{Q}$-function of order h, and ξ is a threshold.

The mapping reward is defined to account for the coverage (i.e, $r^{(\mathsf{c})}_{i,k+1}$) and the accuracy (i.e., $r^{(\mathsf{m})}_{i,k+1}$) as [8]

$$r^{(\mathsf{c})}_{i,k+1} \triangleq \frac{\sum_{j \in \mathcal{I}_{i,k}} \mathbb{I}_{(j \in \mathcal{D}_{i,k})}}{N_{\mathsf{cell}}}, \quad r^{(\mathsf{m})}_{i,k+1} \triangleq \frac{H_{i,k+1|k}(\mathbf{m})}{|\mathcal{I}_{i,k}|}, \quad (6)$$

where $\mathcal{D}_{i,k} \subseteq \mathcal{I}_{i,k}$ indicates the cells visited for the first time by the ith agent, at time k, whereas $\mathcal{I}_{i,k}$ represents the set of indices of all the cells illuminated by the ith UAV at the same kth instant, with $\mathbb{I}_{(x)} = 1$ if the logical condition x is verified, 0 otherwise. In other words, the higher is the number of cells visited for the first time, the higher is the reward. The entropy of $\mathbf{m}$ estimated by the ith agent is

$$H_{i,k+1|k}(\mathbf{m}) = -\sum_{j \in \mathcal{I}_{i,k}} b_{i,k+1|k}(m_j)\, \log_2 \left(b_{i,k+1|k}(m_j) \right),$$

where $b_{i,k+1|k}(m_j)$ is the belief of the occupancy state of the jth cell predicted by the ith agent at time slot k. In our case, we have shaped $r^{(\mathsf{m})}_{i,k+1}$ so that actions should minimize the uncertainty of the map in the shortest possible time.

Obstacles are avoided assuming that the radar is equipped with proximity sensors. To this purpose, numerical penalties are included in the Q-table in order to prevent crashes each time a target or an obstacle is detected.

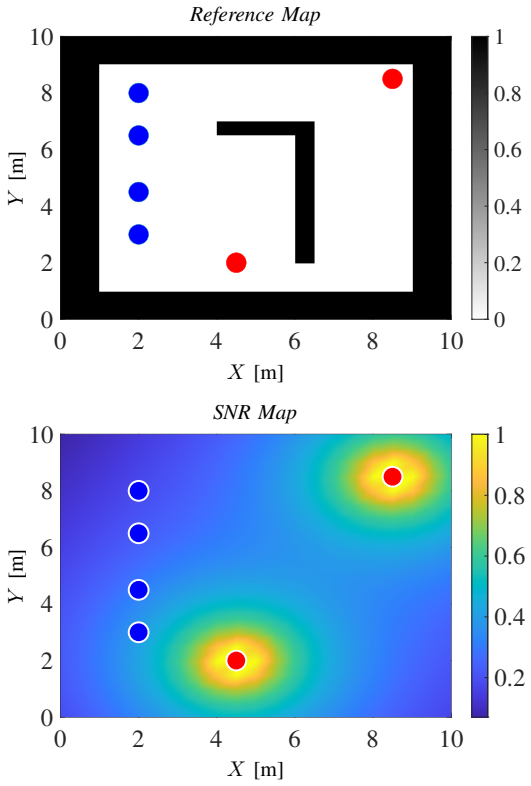

Fig. 1. Top: Reference map with UAVs and targets denoted with blue and red markers, respectively. Bottom: Map of the total SNR at each positions.

IV. Numerical Results

We now provide numerical results for networks of UAVs navigating an unknown indoor environment while detecting the presence of targets.

a) Onboard Sensors: The UAVs are equipped with sub-Terahertz (THz) radar for environmental mapping purposes and with a radio receiver for target detection, both operating at 140 GHz. Concerning the mapping, we assumed that each UAV uses a radar with a 10×10 planar antenna array, with an effective isotropic radiated power (EIRP) of 5 dBm. The scanning time was set to 80μs, and the number of total steering directions was fixed to $N_{\mathsf{steer}} = 10$ in accordance to the

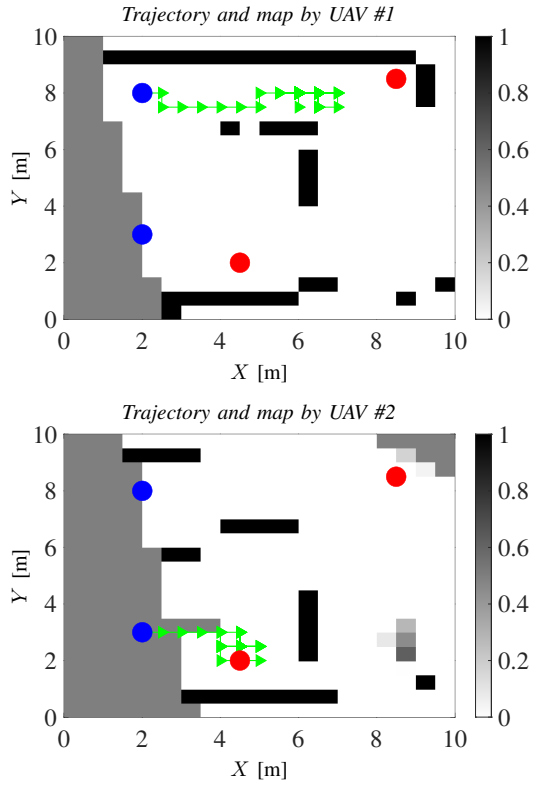

Fig. 2. Estimated trajectories and maps for the first (top) and the second (bottom) UAV at the end of the mission.

array half-power-beamwidth. Concerning the detection, the transmitted power of each target was $P_{\text{TX}} = 1\,\text{W}$, whereas each agent had an ad-hoc RF receiver with observation time of 50 ns, a noise figure of $F = 4\,\text{dB}$, and with an antenna with gain of $G = 10\,\text{dBi}$.

b) Simulation Scenario: The reference scenario is displayed in Fig. 1-top and it is represented by a grid of cells. Empty cells are displayed in white, whereas occupied cells in black. To estimate if a cell is occupied or not, the occupancy grid algorithm in [14] was run on-board each UAV. The map was initialized to $b_0\left(m_j\right) = 0.5,\ \forall j = 1, \ldots, N_{\text{cell}}$ (i.e., maximum uncertainty). The navigation task was solved by running the multi-agent Q-learning described in Alg. 1, where the learning parameters were set to $\alpha = 0.99$, $\gamma = 0.9$, and the probability of taking a random action, i.e., ϵ, was considered as a time-decaying function according to

$$\epsilon = \begin{cases} 0.8, & e < \frac{E}{2},\ k < \frac{K}{4} \\ 0.6, & e < \frac{E}{2},\ \frac{K}{4} \leq k < \frac{K}{2} \\ 0.5, & e < \frac{E}{2},\ \frac{K}{2} \leq k < \frac{3\,K}{4} \\ 0.2, & \text{otherwise.} \end{cases} \tag{7}$$

We fixed the mission time K for each episode to 100, and the number of episodes N_{ep} to 20. The total mission time is given by $T_{\text{M}} = K\,N_{\text{ep}}$.

The UAVs were initially assumed to be in positions $[2,\ y_{i,0}]$ (m) (blue markers of Fig. 1-top) where $y_{i,0}$ is linearly

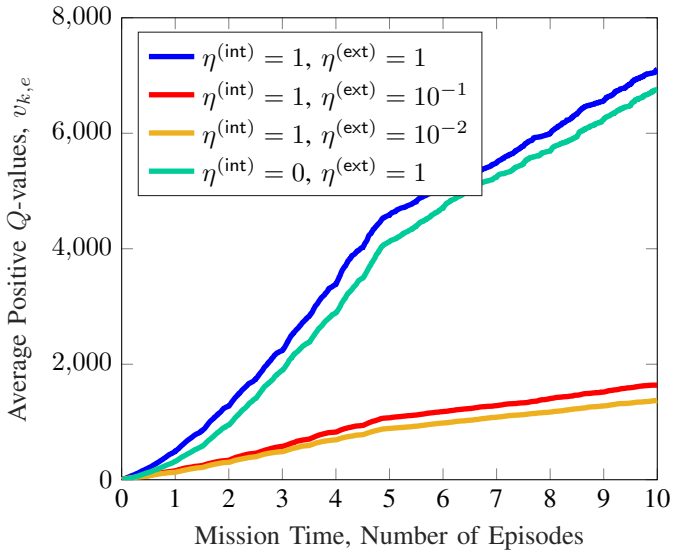

Fig. 3. Positive Q-values as a function of the weighting factors.

spaced between 3 m and 8 m according to the number of agents. Each UAV moved with steps $\Delta = 0.5\,\text{m}$. The number of targets was set to two, and their positions (depicted with red markers in Fig. 1-top) were $[8.5,\ 8.5]$ (m) and $[4.5,\ 2]$ (m). The number of UAVs was varied during the simulations, as we considered $M \in \{1,\ 2,\ 4\}$.

For each episode, the Q-table was inherited from the previous episode so that the acquired global experience could be exploited. Conversely, for each Monte Carlo iteration, the Q-table was reset to zero in order to start a completely new learning process. The episodes were used as training, whereas the last episode was a testing trial.

c) Performance Metric: We also evaluated the learning performance by averaging the total Q-values for all possible states and actions over the Monte Carlo iterations. More specifically, the average of positive Q-values was computed as

$$v_{k,e} = \frac{1}{N_{\text{MC}}} \sum_{q=1}^{N_{\text{MC}}} \left[\sum_{\mathbf{s} \in \mathcal{S}} \sum_{\mathbf{a} \in \mathcal{A}} Q_{k,e}^{q}\left(\mathbf{s}, \mathbf{a}\right) \mathbb{I}_{Q>0} \right], \tag{8}$$

where $Q_{k,e}^{q}$ is the Q-table at the eth training episode, kth time instant, and for the q-th Monte Carlo iteration, and $\mathbb{I}_{Q>0}$ is an indicator function, such that $\mathbb{I}_{Q>0} = 1$ if $Q_{k,e}^{q} > 0$, and zero otherwise.

d) Discussion of Results: Figure 2 shows the estimated trajectories with green markers for the simulation scenario for two UAVs for a single Monte Carlo iteration and for the last episode. As predicted, each UAV got sooner to the closest target with a different level of map reconstruction. The target behind the wall can be hardly revealed at the beginning of the mission because of non-line-of-sight conditions.

In Fig. 3, we evaluate the average positive Q-values by computing (8) for 10 Monte Carlo iterations in a scenario with $M = 2$ UAVs and by varying the weight coefficients in (4) for intrinsic/extrinsic rewards. The detection-based reward

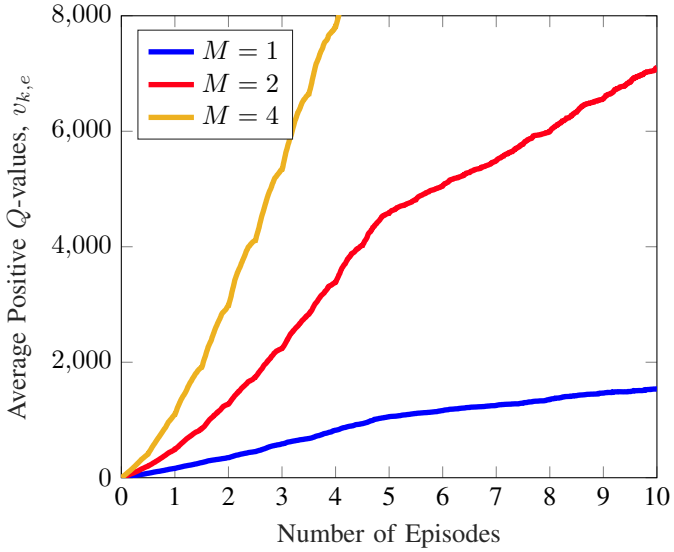

Fig. 4. Positive Q-values as a function of the number of UAVs.

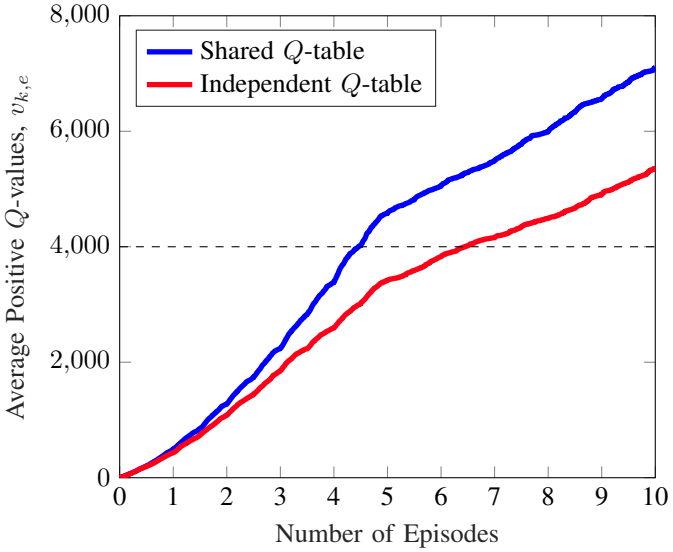

Fig. 5. Positive Q-values with two UAVs sharing or not the Q-table.

$(r_{i,k+1}^{(\mathsf{d})})$ was shaped according to the total normalized SNR, and it played the role of driving and dominant reward most of the time. This is evident in Fig. 3 when the weights were equal to one, or when the mapping reward was not considered at all $(\eta^{(\mathsf{int})} = 0)$. To jointly optimize target detection and mapping, a possible reward shaping scheme is to set the detection weight $\eta^{(\mathsf{ext})}$ to lower values, e.g., $\eta^{(\mathsf{ext})} = 10^{-1}$ or 10^{-2}. In this way, the UAVs could decide to navigate areas with a higher level of uncertainty for mapping.

Figure 4 shows the impact of increasing the number of UAVs M on the learning performance for $\eta^{(\mathsf{ext})} = 1$. As expected, increasing M allowed for improvement of the network's overall knowledge of the environment. The UAVs shared their instantaneous rewards to update a common global Q-table allowing the other UAVs to know the effect of their actions given a certain state. This knowledge was exploited for designing their local navigation strategy and improving the mission completion.

Finally, in Fig. 5, we investigated the impact of collaboration for a network of two UAVs. We simulated a situation where each UAV operated independently from the others by updating a local Q-table, and a second situation where the Q-table was shared among the UAVs. The latter case provided better performance, as the agents took more informative decisions about the environment due to the shared global knowledge. As an example, to achieve the same value of knowledge (e.g., $v_{k,e} = 4000$), the collaborative scenario reduced the learning time of two episodes, corresponding to 400 instants.

V. Conclusions

In this paper we studied a multi-agent reinforcement learning navigation control for UAV networks tasked of detecting multiple targets and mapping an unknown environment within a limited mission time. The results confirm that the employment of coordinated and cooperative UAVs allows for accelerated learning procedure while guaranteeing reliable performance.

References

[1] A. Kumbhar *et al.*, "Exploiting LTE-advanced HetNets and feICIC for UAV-assisted public safety communications," *IEEE Access*, vol. 6, pp. 783–796, 2017.

[2] 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.

[3] 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.

[4] A. Guerra *et al.*, "Networks of UAVs of low-complexity for time-critical localization," *arXiv preprint arXiv:2108.13181*, 2021.

[5] P. Popovski *et al.*, "Wireless access for ultra-reliable low-latency communication: Principles and building blocks," *IEEE Netw.*, vol. 32, no. 2, pp. 16–23, 2018.

[6] 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.

[7] A. M. Ahmed *et al.*, "A reinforcement learning based approach for multi-target detection in massive MIMO radar," *IEEE Trans. Aerosp. Elect. Syst.*, pp. 1–1, 2021.

[8] A. Guerra *et al.*, "Real-time learning for THz radar mapping and UAV control," in *Proc. IEEE Conf. on Autonomous Systems (ICAS)*, 2021.

[9] 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, 2019.

[10] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[11] 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.

[12] I. Fasel *et al.*, "Intrinsically motivated information foraging," in *Proc. IEEE 9th Int. Conf. Develop. Learning*, 2010, pp. 101–107.

[13] A. Mariani, A. Giorgetti, and M. Chiani, "Effects of noise power estimation on energy detection for cognitive radio applications," *IEEE Trans. Commun.*, vol. 59, no. 12, pp. 3410–3420, Dec. 2011.

[14] A. Guerra *et al.*, "Occupancy grid mapping for personal radar applications," in *Proc. IEEE Stat. Signal Process. Work. (SSP)*, 2018, pp. 766–770.