ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

ARCHIVIO ISTITUZIONALE
DELLA RICERCA

Alma Mater Studiorum Università di Bologna
Archivio istituzionale della ricerca

MADRL-Based UAVs Trajectory Design with Anti-Collision Mechanism in Vehicular Networks

This is the final peer-reviewed author's accepted manuscript (postprint) of the following publication:

Published Version:
Spampinato, L., Testi, E., Buratti, C., Marini, R. (2024). MADRL-Based UAVs Trajectory Design with Anti-Collision Mechanism in Vehicular Networks. Piscataway : Institute of Electrical and Electronics Engineers Inc. [10.1109/ICASSP48485.2024.10446347].

Availability:
This version is available at: https://hdl.handle.net/11585/982399 since: 2024-09-10

Published:
DOI: http://doi.org/10.1109/ICASSP48485.2024.10446347

Terms of use:

Some rights reserved. The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

This item was downloaded from IRIS Università di Bologna (https://cris.unibo.it/).
When citing, please refer to the published version.

(Article begins on next page)

09 December 2025

MADRL-BASED UAVS TRAJECTORY DESIGN WITH ANTI-COLLISION MECHANISM IN VEHICULAR NETWORKS

Leonardo Spampinato[1,2], *Enrico Testi*[1,2], *Chiara Buratti*[1,2], *Riccardo Marini*[1]

[1]Wilab, CNIT, Bologna, Italy;
[2]DEI Department, University of Bologna, Bologna, Italy;
{leonardo.spampinato, enrico.testi, c.buratti}@unibo.it, riccardo.marini@cnit.it

ABSTRACT

In upcoming 6G networks, unmanned aerial vehicles (UAVs) are expected to play a fundamental role by acting as mobile base stations, particularly for demanding vehicle-to-everything (V2X) applications. In this scenario, one of the most challenging problems is the design of trajectories for multiple UAVs, cooperatively serving the same area. Such joint trajectory design can be performed using multi-agent deep reinforcement learning (MADRL) algorithms, but ensuring collision-free paths among UAVs becomes a critical challenge. Traditional methods involve imposing high penalties during training to discourage unsafe conditions, but these can be proven to be ineffective, whereas binary masks can be used to restrict unsafe actions, but naively applying them to all agents can lead to suboptimal solutions and inefficiencies. To address these issues, we propose a rank-based binary masking approach. Higher-ranked UAVs move optimally, while lower-ranked UAVs use this information to define improved binary masks, reducing the number of unsafe actions. This approach allows to obtain a good trade-off between exploration and exploitation, resulting in enhanced training performance, while maintaining safety constraints.

***Index Terms*—** UAVs, V2X, Reinforcement Learning, Collision-avoidance.

1. INTRODUCTION

Nowadays, the use of Unmanned Aerial Vehicles (UAVs) as flying Base Stations (BSs), namely Unmanned Aerial Base Stations (UABSs), has gained considerable attention in the literature as a promising technology for future 6G networks [1, 2]. The ability of UABS to operate at high altitudes opens up the possibility of providing Line-of-Sight (LoS) links with ground users, leading to enhanced network coverage and capacity for a wide variety of applications. An important use case is foreseen in vehicular communications [3, 4], where vehicles have to exchange large amounts of information with the network and among themselves. In such scenarios, UABS can significantly enhance network performance.

Moreover, addressing complex urban environments poses a crucial challenge: the design of optimized trajectories for multiple UABS to ensure a desired network performance. Additionally, introducing a coordination mechanism among UABSs becomes crucial to ensure optimal coverage of the environment while avoiding overlapping trajectories that may lead to interference or, even worse, flight collisions.

Extensive research has been conducted on collision avoidance for UAV swarms. Conventional methods include velocity obstacles (VO), which involves frequent changes in UAVs velocities [5], artificial potential field (APF), which does not fully leverage drone cooperation [6], and particle swarm optimization (PSO), which generates energy inefficient zigzag trajectories [7]. Addressing these challenges often requires complex mathematical models and significant computational time. In response, machine learning-based solutions [8–10], particularly Multi-Agent Deep Reinforcement Learning (MADRL)-based algorithms [11, 12], offer promising alternative approaches to mitigate such issues. Many existing works in the literature address collision avoidance by introducing constraints on the agents' mobility models and tweaking the reward function. These constraints enforce a minimum safety distance between any two agents at any time instant [13–16]. In these works, the reward function is modified to include a simple penalty term based on mutual distances between agents. A slightly different approach is proposed in [15], where each agent is surrounded by a 'repulsive field' that induces negative and positive rewards proportional to encroachment and target distances, respectively. Nevertheless, such methods restrict the movement space of agents, that are forced to learn suboptimal policies. Furthermore, solutions like the ones proposed in the aforementioned works limit the collision probability but do not entirely prevent collisions, posing challenges when applied to real-world systems.

This paper presents a MADRL model that can optimize the trajectories of a swarm of UABS deployed over an urban scenario. The model is characterized by a mask-based mechanism that allows UABS to avoid any flight collisions, as well as possible interference with other agents that are serving vehicular users under their fields of view. More specifically, to improve the agents' performance, a ranking system is introduced: high-ranked agents will have the opportunity to follow the best trajectories, whereas low ones will have to respect the safety constraint using binary masking. Results are presented in terms of the cumulative reward collected by agents, showing the convergence property of the training algorithm designed, and in terms of the percentage of satisfied users, indicating how well the proposed system can meet the stringent requirements of Vehicle-To-Everything (V2X) applications while respecting the safety constraint introduced.

The rest of the paper is organized as follows: Section 2 describes the system model, Section 3 offers an overview of the MADRL model and a detailed analysis of the proposed anti-collision mechanism, Section 4 shows the results obtained and Section 5 depicts the main conclusions.

©2024 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

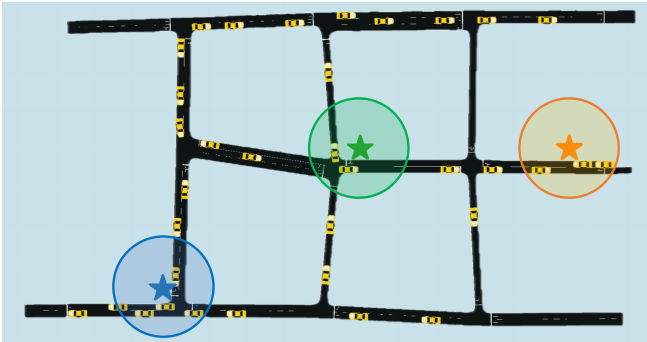

Fig. 1. Manhattan-style street layout showcasing UABSs as star points, with colored circles indicating their coverage areas.

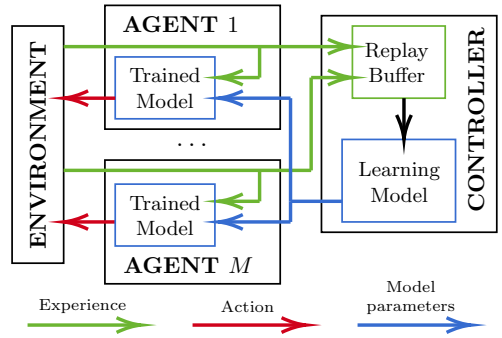

Fig. 2. Multi-agent DRL architecture. Agents continuously interact with a shared environment. Experiences are collected in a shared replay buffer at the central controller, which periodically updates the learning model and sends it back to each agent.

2. SYSTEM MODEL

2.1. Scenario

Let us consider a set $\mathcal{G}$ of vehicles, namely Ground User Equipments (GUEs), moving within an urban area while running an extended sensing application. This application necessitates each GUE to gather data from its onboard sensors and exchange information with other GUEs and the network. Let us also consider a set $\mathcal{U}$ of M UABSs, equipped with mmWave radio antenna systems working at carrier frequency f_c, flying in the same region and acting as relays to provide reliable and continuous service, capable of meeting stringent requirements.

We assume that each UABS $u \in \mathcal{U}$ flies at constant speed v, with fixed altitude h, with positions on xy-plane given by $(x_t^{(u)}, y_t^{(u)})$ at each discrete time instant t. Time is discretized as $t = 0, 1, \ldots, T$, where T is the duration of the flight mission and each timestep has duration δt. A flight comprises a discrete sequence of positions assumed at each timestep.gene

It is assumed that each UABS always has an active and reliable Command and Control (C2) link towards a central unit, hereafter denoted as *controller*. We assume that the C2 links operate on distinct frequency bands w.r.t. those used for communication with GUEs, ensuring no interference between the two systems.

The service area, depicted in Fig. 1, follows a Manhattan-style street layout. The routes taken by GUEs and their movement are simulated using Simulation of Urban MObility (SUMO) [17]. Since in this work we focus on the anti-collision mechanism, in our scenario, we simulate a small portion of a city where GUEs are concentrated.

2.2. Application Requirements

For this work, we focus on the collection of uplink packets, from GUEs to UABSs. It is assumed that GUEs transmit one packet per timestep, i.e., they generate periodic traffic with period δt. A GUE is *served* at timestep t if its packet is correctly received by at least one UABS, i.e., the corresponding uplink Signal-to-Noise Ratio (SNR) $\mathrm{SNR}^{(\mathrm{g,u})}$ evaluated at one UABS u exceeds a threshold, $\mathrm{SNR}_{\mathrm{th}}$. Let us define a service window as a series of N consecutive timesteps starting from a generic timestep $\bar{t}$. We now assume that each GUE needs to be served for at least $\hat{N}_s$ time slots inside the same service window, to provide continuous service. Then, a GUE is said to be *satisfied* in a service window if $N_s \geq \hat{N}_s$, where N_s is the number of slots for which it is served. With these definitions, we model the reliability and continuity of service. Furthermore, to keep track of the service history, a priority term $p_t^{(g)}$ is assigned to each GUE g, and for each service window varies as follows:

$$p_t^{(g)} = \begin{cases} 1, & \text{if } t = \bar{t} \quad (1) \\ p_{t-1}^{(g)} + 1, & \text{if } \exists u \in \mathcal{U} : \mathrm{SNR}_{\mathrm{t}}^{(\mathrm{g,u})} \geq \mathrm{SNR}_{\mathrm{th}} \quad (2) \\ p_{t-1}^{(g)}, & \text{otherwise.} \quad (3) \end{cases}$$

2.3. Channel Model

The channel model considered is the 3GPP Urban Macro (UMa) [18]. For a generic link between two terminals, such a model defines a probability to be in LoS condition, ρ_L. The SNR in dB can be expressed as $\mathrm{SNR} = P_{\mathrm{tx}} + G_{\mathrm{tx}} + G_{\mathrm{rx}} - PL - P_{\mathrm{n}}$, where P_{tx} is the transmit power in dBm, G_{tx} and G_{rx} are, respectively, the gains of the transmitter and receiver antenna in dB, PL is the path loss in dB (from Table 7.4.1-1 and 7.4.2-1 in [18]), which depends on ρ_L, and P_{n} is the noise power at the receiver side in dBm. Furthermore, we assume UABSs use beamforming generating $B = 9$ circular beam footprint on the ground in a 3x3 grid. By defining the field of view of the UABS on the vertical plane as ϕ, the solid angle of a single beam can be derived as $\Phi_{\mathrm{beam}} \approx \frac{2\pi(1-cos(\phi/2))}{B}$ and the maximum gain G can be expressed as $G = \frac{41000}{\left(\Phi_{beam}\frac{360}{2\pi}\right)^2}$, assuming an ideal radiation pattern, with gain G inside Φ_{beam} and 0 dB outside [19].

3. MULTI AGENT REINFORCEMENT LEARNING MODEL

The architecture of the proposed MADRL algorithm is depicted in Fig. 2. Multiple agents (i.e., the UABSs) cooperate to accomplish a common task in a shared, dynamic, and unknown environment, that is the urban area with GUEs moving along the streets. For these reasons, UABSs must interact with each other to learn how to effectively perform the task and find a behavioral policy π, represented by a Neural Network (NN) whose parameters must be tuned by means of a learning algorithm. The controller is a central unit that collects all the experiences of the agents in a replay buffer, trains the learning model, and periodically shares the updated policy with the agents.

At timestep t each UABS u observes a limited and local representation of the environment, namely $o_t^{(u)}$. We define a local observation as a tuple $o_t^{(u)} = (x_t^{(u)}, y_t^{(u)}, t, \mathbf{L}_t, \mathbf{b}_t^{(u)})$, where $\mathbf{L}_t$ is a

matrix containing all the current locations of agents in the fleet, and $\mathbf{b}_t^{(u)}$ is the *per beam information*, i.e., a vector of length B whose elements $b_{i,t}^{(u)}$ corresponds to the sum of priority $p_t^{(g)}$ of GUEs under the i-th beam of u-th agent at timestep t. Based on the policy π, each UABS selects its own action $a_t^{(u)}$ belonging to the set $\mathcal{A}$ of possible directions of movement, with $\mathcal{A} = \{\uparrow, \rightarrow, \downarrow, \leftarrow\}$. As a consequence of the collective actions taken by all agents, the environment will change leading to new local observations $o_{t+1}^{(u)}$ for each agent. Local observations and chosen actions are sent back to the controller so that a per agent reward $r_t^{(u)}$ can be computed. The reward indicates how favorable or unfavorable has been choosing action $a_t^{(u)}$ within the context of local observations $o_t^{(u)}$ and $o_{t+1}^{(u)}$. At this point, the experience tuple $(o_t^{(u)}, a_t^{(u)}, r_t^{(u)}, o_{t+1}^{(u)})$ is stored in the shared replay buffer. By defining $d_t^{(u,w)}$ as the euclidean distance between agents u and w at a timestep t and $\|\mathbf{x}\|_1 = \sum_{x \in \mathbf{x}} |x|$ the 1-norm of a vector $\mathbf{x}$, the reward function can be written as:

$$r_t^{(u)} = \begin{cases} \|\mathbf{b}_{t+1}^{(u)}\|_1, & \text{if } \forall w \in \mathcal{U} \setminus \{u\} : d_{t+1}^{(u,w)} \geq d_{\text{th}} \quad \text{(4a)} \\ -\lambda_s, & \text{if } \exists w \in \mathcal{U} \setminus \{u\} : d_{t+1}^{(u,w)} < d_{\text{th}} \quad \text{(4b)} \\ -\lambda_c, & \text{if } \exists w \in \mathcal{U} \setminus \{u\} : d_{t+1}^{(u,w)} < 1. \quad \text{(4c)} \end{cases}$$

(4a) corresponds to the sum of the priority of GUEs covered by the beams of UABS u, after choosing action $a_t^{(u)}$ at time t, reaching a new position and getting the per beam information at timestep $t+1$. On the other hand, when distance requirements are not met in (4b) and (4c), agents get penalties equal to λ_s and λ_c, respectively. The former condition happens when two agents start to interfere with each other, with $d_{\text{th}} = 2h \arctan \phi$ being twice their coverage radius, whereas the latter corresponds to a collision event between them.

After T_u timesteps, the controller will update the NN parameters following a Double Dueling Deep Q Network (3DQN) learning algorithm and share the new model with each agent. To reduce complexity, improve performance, and leverage a *centralized learning distributed execution* paradigm, the trained policy π is shared among all agents so that each one exploits the same one. Additionally, centralized training avoids high consumption of energy at the UABS, whereas the distributed execution, guaranteed by the local model replica, allows each UABS to act quickly once new observations arrive, reducing possible delays.

Fundamental concepts of the 3DQN implementation are reported hereafter. More details can be found in [8]. In short, the aim of 3DQN algorithm is to estimate the Q-values, representing the expected future rewards when choosing an action a in a given observation o, for all possible actions and observations. Periodical updates of the NN parameters are based on the stochastic gradient descent on mini-batches of local experiences (o_t, a_t, r_t, o_{t+1}) randomly sampled from the shared replay buffer. These experiences can be collected either by *exploiting* the current policy π and choosing the action with the highest Q-value when observation o_t is perceived, or by *exploring* the environment, thus choosing random actions, possibly transitioning to new positions obtaining higher rewards, and learn a better policy. Agents must strike a good balance between exploration and exploitation by following an ϵ-greedy policy. Additionally, a *double* loss calculation, to avoid overestimation, and a *dueling* architecture, useful to increase the sample-efficiency of the algorithm, are exploited to provide better learning outcomes. Indeed, 3DQN provides good performance in this type of scenario, balancing training convergence and real-time execution of new trajectories.

3.1. Anti-Collision Mechanism

In real-world scenarios, the likelihood of UABSs flying in dangerously close proximity is not negligible, making it imperative to effectively diminish the collision risk during their missions. The widely-adopted penalty-based strategy [13–16] requires the definition of a reward function that imposes penalties on agents when they get too close, as outlined in (4b) and (4c). As simulations will show, this approach often results in agents colliding with each other, failing to provide a definitive solution to the problem. Hence, we introduce two novel collision avoidance mechanisms, namely *flat* and *rank masking*, which rely on constraining the agents' action space to entirely avert collisions.

The flat masking algorithm works as follows: i) at timestep t, the controller examines each agent and assesses all their potential future positions; ii) for each couple of agents $u, w \in \mathcal{U}$, actions for which $d_{t+1}^{(u,w)} < d_{\text{th}} + v\,\delta t$ are classified as illegal and removed from the action space, resulting in temporarily restricted sets of safe actions $\tilde{\mathcal{A}}_t^{(u)}$ and $\tilde{\mathcal{A}}_t^{(w)}$. The term $v\,\delta t$ is introduced as an additional safety factor accommodating for the uncertainty caused by the selection of random actions that are not known in advance by the agents. The restricted binary masks are sent to each agent via the C2 link.

It becomes evident that the utilization of flat masking significantly dampens the agents' capacities for both exploration and exploitation. For instance, in a scenario where many agents are directed toward a specific location, the masking effect induces a sense of mutual repulsion, preventing any of them from reaching the destination. This directly affects the quality of service provided to GUEs.

To address this limitation and further improve the system's performance, we introduce the novel concept of rank masking: i) a rank is assigned to each agent by means of a score function $F(u) = \|\mathbf{b}_t^{(u)}\|_1$, that is the sum of the priorities of GUEs currently covered by the u-th agent; ii) the couples $u, w \in \mathcal{U}$ for which $d_t^{(u,w)} < d_{\text{th}} + 2\,v\,\delta t$ are identified as potential colliders; iii) if two agents can potentially collide, the one with the highest rank is free to take the next action without any limitation, according to its policy; iv) the agent u with the lowest rank calculates its constrained set of actions, $\tilde{\mathcal{A}}_t^{(u)}$, for which $d_{t+1}^{(u,w)} < d_{\text{th}}$, based on the action chosen by agent w.

4. RESULTS

In this section, we provide simulation results to compare penalty and masking-based collision avoidance mechanisms, proving that the former is less efficient than the latter in terms of performance. It is also shown that ranked-based masking performs better than flat masking. As performance metrics, we used the cumulative reward, i.e., the sum of agents' reward, $R = \sum_{u \in \mathcal{U}} \sum_{t=0}^{T} r_t^{(u)}$, which provides feedback on the training performance of the designed MADRL algorithm, and the percentage of satisfied users P_g, calculated as

$$P_g = \frac{1}{|\mathcal{G}|} \sum_{g \in \mathcal{G}} \frac{N_g^{(sat)}}{N_g} \tag{5}$$

where N_g is the total number of service windows of the g-th GUE, while $N_g^{(\text{sat})}$ are those for which $N_s \geq \hat{N}_s$. The simulation parameters used are reported in Table 1.

M agents are trained for N_t episodes of duration T. One UABS is randomly placed inside the area, while the others are uniformly distributed on its perimeter. The scenario considered has dimensions $350 \times 170\,\text{m}^2$. Every N_e training episodes we perform an evaluation of the current policy π on vehicle traces that have not been used for

Table 1. Simulation parameters

M	3	h	100 m	v	20 m/s
ϕ	40°	$\lvert\mathcal{G}\rvert$	80	δt	1 s
f_c	30 GHz	P_{tx}	14 dBm	P_{n}	-106.4 dBm
G_{tx}	0 dB	G_{rx},	38 dB	SNR_{th}	-13.7 dB
T	80 s	λ_s	10	λ_c	1000
N_t	1000	N_e	20	T_u	1

Table 2. Average number of collisions occurrences

	Penalty	Flat Masking	Rank Masking
Collisions in Training	170 (17%)	0 (0%)	0 (0%)
Collisions in Evaluation	14 (28%)	0 (0%)	0 (0%)

the training. During evaluation agents are not allowed to perform random actions, so we can evaluate and compare their behavior in a fixed and fair setting.

Table 2 lists the average number of collisions in training and evaluation episodes, showing how masking can be used to completely avoid collisions in contrast to penalty-based algorithms.

In Fig. 3 the cumulative reward, R, obtained during evaluation episodes is shown. It is possible to notice that the use of the penalty-based approach turns out to be ineffective in this highly dynamic scenario due to the mix of penalties received by the agents. Additionally, thanks to the improved exploration and exploitation capability, rank-based masking allows agents to collect higher rewards than flat masking. Let us remark that when anti-collision mechanism is active the agents only collect the positive rewards outlined in (4a).

Fig. 4 shows the P_g by varying the service threshold, $\hat{N}_s$. Values are obtained from the best evaluation episode for each setting. As expected, rank-based masking is again proven to be better than flat masking. However, both masking strategies easily outperform the trajectories obtained with penalty-based training. It is worth mentioning that the low values of user satisfaction are mainly attributed to the UABSs' limited coverage footprint and the beam ideality hypothesis, causing a maximum instantaneous coverage area that is only 12% of the overall service area. Indeed, the system performance can be improved by introducing a fixed ground network capable of serving GUEs cooperating with UABSs.

5. CONCLUSIONS

This paper addresses anti-collision methods for a group of autonomous UABS whose aim is to continuously serve groups of GUEs. Indeed, when trajectories are learned using MADRL algorithms, it becomes difficult to respect basic safety constrain on the UABSs' mutual distance without a proper design. To this end, three strategies for collision avoidance are presented and compared. The study finds that incorporating penalties into the reward function is not enough to avoid collisions, necessitating a binary mask to prevent unsafe actions. Moreover, a ranking system can be adopted to ensure service continuity and improve the exploration and exploitation capabilities of the fleet during the training phase.

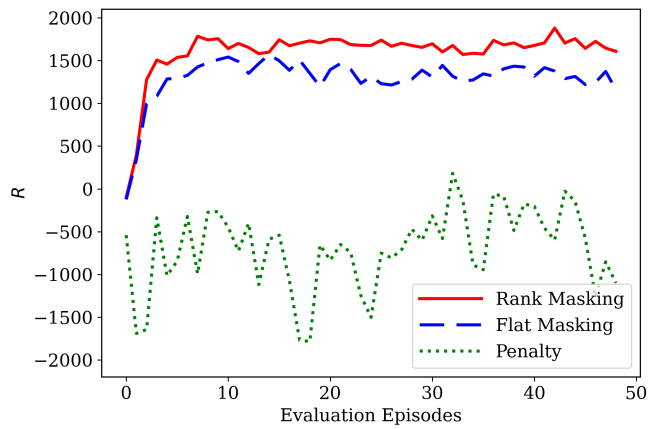

Fig. 3. Cumulative reward in evaluation episodes for different anti-collision mechanisms.

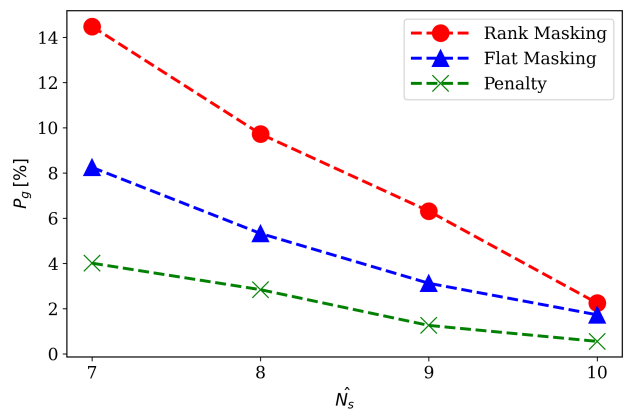

Fig. 4. Percentage of satisfied users for different anti-collision mechanisms.

6. ACKNOWLEDGMENT

This work has been carried out in the framework of the CNIT WiLab-Huawei Joint Innovation Center and also supported by the European Union under the Italian National Recovery and Resilience Plan (NRRP) of NextGenerationEU, partnership on "Telecommunications of the Future" (PE00000001 - program "RESTART", Structural Project 6GWINET). We thank Aman Jassal for the very fruitful discussion on this paper.

7. REFERENCES

[1] A. I. Hentati and L. C. Fourati, "Comprehensive survey of UAVs communication networks," *Computer Standards and Interfaces*, vol. 72, p. 103451, 2020.

[2] S. Mignardi, D. Ferretti, R. Marini, F. Conserva, S. Bartoletti, R. Verdone, and C. Buratti, "Optimizing beam selection and resource allocation in UAV-aided vehicular networks," in *EuCNC/6G Summit 2022*, 2022, pp. 184–189.

[3] B. M. Masini, A. Bazzi, and A. Zanella, "A survey on the roadmap to mandate on board connectivity and enable V2V-based vehicular sensor networks," *Sensors*, vol. 18, no. 7, 2018.

[4] 5GAA, "A visionary roadmap for advanced driving use cases, connectivity technologies, and radio spectrum needs," *White Paper*, Sep. 2020.

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

[6] S. Huang and K. H. Low, "A path planning algorithm for smooth trajectories of unmanned aerial vehicles via potential fields," in *Int. Conf. Control Autom. Robot. Vis. (ICARCV)*, Singapore, Nov. 2018, pp. 1677–1684.

[7] B. Song, Z. Wang, L. Zou, L. Xu, and F. E. Alsaadi, "A new approach to smooth global path planning of mobile robots with kinematic constraints," *Int. J. Mach. Learn. Cybern.*, vol. 10, no. 1, pp. 107–119, Jan. 2019.

[8] L. Spampinato, A. Tarozzi, C. Buratti, and R. Marini, "DRL path planning for UAV-aided V2X networks: comparing discrete to continuous action spaces," in *ICASSP 2023*, 2023, pp. 1–5.

[9] R. Marini, L. Spampinato, S. Mignardi, R. Verdone, and C. Buratti, "Reinforcement learning-based trajectory planning for UAV-aided vehicular communications," in *EUSIPCO 2022*, 2022, pp. 967–971.

[10] R. Marini, S. Park, O. Simeone, and C. Buratti, "Continual meta-reinforcement learning for UAV-aided vehicular wireless networks," in *ICC 2023 - IEEE International Conference on Communications*, 2023, pp. 5664–5669.

[11] E. Testi, E. Favarelli, and A. Giorgetti, "Reinforcement learning for connected autonomous vehicle localization via UAVs," in *IEEE Int. Workshop Metrol. Agric. For. (MetroAgriFor)*, Trento, Italy, Nov. 2020, pp. 13–17.

[12] A. Guerra, F. Guidi, D. Dardari, and P. M. Djuric, "Reinforcement learning for joint detection and mapping using dynamic UAV networks," *IEEE Trans. Aerosp. Electron. Syst.*, pp. 1–16, Aug. 2023.

[13] X. Wang and M. C. Gursoy, "Learning-based uav trajectory optimization with collision avoidance and connectivity constraints," *IEEE Transactions on Wireless Communications*, vol. 21, no. 6, pp. 4350–4363, 2022.

[14] S. Xu, X. Zhang, C. Li, D. Wang, and L. Yang, "Deep reinforcement learning approach for joint trajectory design in multi-UAV IoT networks," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 3, pp. 3389–3394, Mar. 2022.

[15] N. Thumiger and M. Deghat, "A multi-agent deep reinforcement learning approach for practical decentralized UAV collision avoidance," *IEEE Control System Letters*, vol. 6, pp. 2174–2179, Dec. 2022.

[16] H. Qie, D. Shi, T. Shen, X. Xu, Y. Li, and L. Wang, "Joint optimization of multi-UAV target assignment and path planning based on multi-agent reinforcement learning," *IEEE Access*, vol. 7, pp. 146 264–146 272, Sep. 2019.

[17] P. A. L. et al., "Microscopic traffic simulation using SUMO," *2019 IEEE ITSC*, pp. 2575–2582, November 2018.

[18] 3GPP, "Technical specification group radio access network; study on channel model for frequencies from 0.5 to 100 GHz," *TR 38 901 version 16.1.0*, Dec. 2019.

[19] V. Salvia, *Antenna and Wave Propagation*. Laxmi Publications, 2007.