Reinforcement Learning for Optimization of Collision Avoidance in Multiple Autonomous Aerial Robots Systems

Marcos Rodrigues Vizzotto [1,*], Guilherme Kohl [2], Carlos Eduardo Pereira [1,3], Edison Pignaton de Freitas [1,3,4]

***Abstract*—The increasing use of Unmanned Aerial Vehicles (UAVs) in applications such as logistics, environmental monitoring, infrastructure inspections, and precision agriculture has brought new safety and operational efficiency challenges. In this context, ensuring collision avoidance in operations with multiple UAVs in shared spaces has become a priority. This work proposes a framework based on Deep Reinforcement Learning (DRL), designed to optimize collaborative trajectory planning and collision avoidance in dynamic environments, which include static and moving obstacles. The approach combines LiDAR sensors with Detection And Avoidance (DAA) algorithms. These systems allow UAVs to identify and avoid obstacles accurately, even in dense urban scenarios and with inaccurate sensors. In addition, the framework promotes collaboration between UAVs through decentralized architectures, allowing greater efficiency in the use of airspace and reducing overlap in shared missions. Performed simulations have shown that the proposed approach provides improvement in movement safety. The results show that the system is adaptable in dynamic scenarios, making it promising for applications in complex airspaces.**

***Index Terms*—Deep Reinforcement Learning, Path Planning, Collision Avoidance, Multiples Unmanned Aerial Vehicles.**

I. Introduction

The increasing use of unmanned aerial vehicles (UAVs) in applications such as logistics, infrastructure inspections, precision agriculture, and environmental monitoring has transformed how people perform many activities. However, this increase in the number of UAVs operating in shared airspaces poses critical challenges related to the safety and efficiency of operations. In particular, the need to avoid collisions between drones and other objects has become a central concern to ensure safe and sustainable operations [1] [2] [3].

Introducing algorithms based on deep reinforcement learning (DRL) has shown promise in solving complex robotics navigation and trajectory planning problems [4]. Applied to UAVs, DRL allows them to learn optimal navigation policies directly from interactions with the environment, even under uncertain conditions such as imperfect sensing and complex environmental dynamics [5] [6] [7]. Learning-based systems, such as those proposed in recent studies, have demonstrated significant advances in the efficiency of collision avoidance maneuvers, reducing energy consumption and increasing operational safety [8] [9].

Integrating detection and avoidance (DAA) systems with advanced technologies like LiDAR sensors and data fusion algorithms allows UAVs to detect and avoid obstacles in runtime. These systems are essential for urban operations, where the proximity of buildings, vegetation, and other UAVs represents a significant challenge. In addition, studies highlight the importance of decentralized and collaborative architectures that allow efficient coordination between multiple UAVs in the same airspace [3] [7] [10].

Another area of focus is collaborative trajectory planning for multiple UAVs, which aims to improve efficiency in area coverage while avoiding collisions. Methods such as map sharing and data fusion allow for reducing overlaps, increasing mission efficiency, and optimizing airspace use. These advances also involve planning strategies based on three-dimensional semantic maps and uncertainty prediction using probabilistic models and deep learning [11] [12] [13].

In addition, the use of hybrid methods, such as reinforcement learning combined with trajectory-based optimization, has expanded the field of application of UAVs. These methods address challenges such as highly dynamic environments, high air traffic, and integration with sensor networks and IoT (Internet of Things) devices, allowing greater resilience and adaptability to operations [14] [15] [16].

Although research has achieved significant advances, challenges remain related to the efficient coordination of multiple UAVs, runtime collision avoidance, and performance evaluation in practical applications in dynamic and complex environments.

This work aims to overcome these limitations by proposing a DRL-based approach that integrates trajectory planning and collision avoidance for multiple UAVs, presenting:

- Runtime collision avoidance between UAVs and static objects;
- Improved trajectory planning in cooperative operations; and
- Adaptation to dynamic scenarios, including dense environments, with imprecise sensors.

The proposed methodology is evaluated through simulations in complex environments, highlighting its effectiveness in enhancing UAVs' safety operational.

This paper is organized as follows: Section II discusses related works. Section III describes the addressed problem in the work. Section IV details the proposed approach. Section V presents and discusses the acquired results. Finally, Section VI concludes the paper by presenting directions for future work.

[1] Graduate Program in Electrical Engineering, Federal University of Rio Grande do Sul, Porto Alegre, Brazil.

[2] Bachelor in Computer Science, Federal University of Rio Grande do Sul, Porto Alegre, Brazil, 91.501-970.

[3] Competence Center on Digital Agriculture (EMBRAPII / SENAI-RS), São Leopoldo-RS, Brazil.

[4] School of Information Technology, Halmstad University, Halmstad, Sweden.

[*] correspondent author: marcos.vizzotto@ufrgs.br

II. Related Works

A. Autonomous Exploration and Mapping

Liu et al. [11] introduce the concept of active metric-semantic mapping, which combines geometric and semantic information to construct informative and optimized 3D maps for long-range missions. Their approach minimizes map uncertainty and allows multiple UAVs to collaborate efficiently in tasks such as precision agriculture and infrastructure inspection. This strategy is particularly relevant for urban missions where the semantics of the environment (such as building and vehicle identification) are critical for safe navigation.

Complementarily, Tao et al. [15] propose a framework that integrates deep learning-based prediction to estimate the occupancy of unexplored areas. They use RGB-D cameras and planning algorithms that adjust navigation objectives based on predictive data, ensuring safe trajectories. Their work stands out for validating the framework in real environments with high obstacle density, demonstrating a 50–60

B. Collision Avoidance with DRL

The use of DRL for collision avoidance has proven to be a robust solution, especially in dynamic scenarios. Wang et al. present an innovative two-step approach. Supervised training focuses on known collision avoidance strategies, while the subsequent step uses gradient-based refinement of policies to improve the system's adaptability [17]. This methodology is ideal for UAVs operating under imperfect sensing conditions, such as LiDAR sensors or camera interference.

Xue and Gonsalves [18] investigate the use of deep reinforcement networks for obstacle avoidance in three-dimensional spaces using only visual data. They integrate a SAC (Soft Actor-Critic) algorithm with a flight simulator (Airsim). This shows that the trained model can generalize to previously unseen environments, maintaining high success rates in obstacle avoidance in environments such as forests and urban spaces.

C. Cooperative Trajectory Planning and Map Fusion

The challenges of trajectory planning for multiple UAVs have led to the development of innovative methods, such as the one proposed by Ivić et al. [13]. They use an algorithm based on potential fields combined with camera orientation control for detailed visual inspections of complex structures, such as wind turbines and bridges. The work highlights the use of coverage metrics to ensure that all relevant surfaces are inspected while avoiding collisions between UAVs.

D. Detection and Evasion (DAA) Systems

Integrating Detection And Evasion (DAA) systems has been critical to the safety of UAVs in urban environments. Ince et al. [7] present a framework that uses LiDAR sensors and RGB-D cameras integrated with fusion algorithms to perform real-time obstacle detection and avoidance. This approach is aligned with beyond-line-of-sight (BVLOS) regulations and has been validated in simulated high-density urban scenarios where accuracy and reaction speed are crucial.

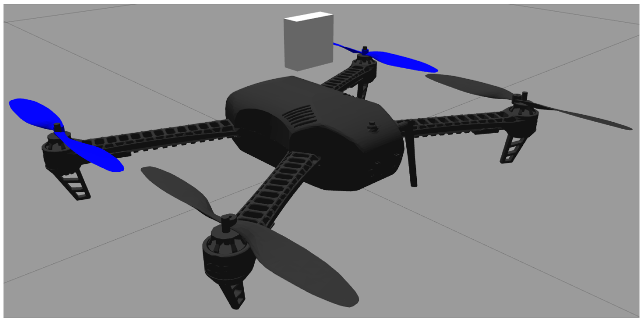

Fig. 1. Iris model with rplidar model.

Computer vision-based systems, such as those explored by Jaiswal et al. [10], also play an important role by leveraging advanced algorithms for image analysis and collision detection. Their application in small UAVs is highlighted for reducing operational costs and facilitating integration into power-constrained platforms.

E. Current Gaps and Challenges

Despite the progress so far, significant challenges still exist in the field. For example, real-time 3D map fusion methods, although promising, lack scalable solutions for scenarios with dozens of UAVs operating simultaneously. Furthermore, Jaiswal et al. [10] point out the lack of consensus on strategies to minimize energy consumption in collaborative systems.

Another important challenge is the adaptation of exploration algorithms to highly dynamic environments. Shin et al. [9] highlight the difficulty of adjusting trajectories in runtime when UAVs face frequent changes in the environment, such as new obstacles or changing weather conditions.

This work aims to fill these gaps by proposing a DRL-based algorithm that integrates trajectory planning and collision avoidance for multiple UAVs. The proposed methodology focuses on the collaborative fusion of maps and adaptation to dynamic scenarios, seeking to maximize safety and operational efficiency in challenging missions.

III. Problem Statement

This paper addresses the problem of UAVs' movement control in order to avoid collisions with boundary walls, static objects, and other moving UAVs. The UAVs need to move within a confined area of dimension X×Y×Z while avoiding collisions. The movement of the UAVs starts from a predefined initial position at $StartPoint$ and must reach a destination $TargetPoint$. The UAVs must initially fly at the same altitude with initial intersecting trajectories. The possible movements of each UAV are limited according to the environment.

For simplicity, this work assumes a constant speed for the UAVs. Furthermore, it is assumed that the UAVs operate autonomously, without any ground remote control or predefined waypoint plan; only the starting and target positions are defined. On the other hand, it is assumed that the 3D flight area may include both static and moving obstacles. Therefore,

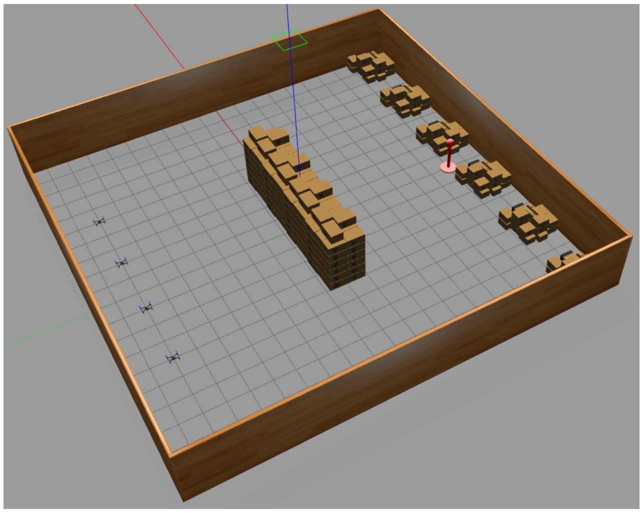

Fig. 2. Simulated environment for collision avoidance study.

the UAVs must be equipped with an accurate LiDAR sensor to accurately detect the positions of the surrounding objects as illustrated in Figure 2.

UAVs may collide with one of the moving and static obstacles. Each UAV assumes that the trajectory of the other UAVs is unknown and that they may become the moving obstacles in its trajectory. A collision occurs whenever the Euclidean distance is smaller than a predefined threshold distance between the UAV pose and the obstacle pose. Then, a collision instance is formally defined as follows:

$$\text{Collision} = \begin{cases} \text{True}, & \text{if } d < d_{\text{limit}}, \\ \text{False}, & \text{otherwise.} \end{cases}$$

Where,

- d is the Euclidean distance between the UAV and the obstacle, defined as follows:

$$d = \sqrt{(x_{\text{UAV}} - x_{\text{obs}})^2 + (y_{\text{UAV}} - y_{\text{obs}})^2 + (z_{\text{UAV}} - z_{\text{obs}})^2}$$

- x_{UAV}, y_{UAV}, z_{UAV} are UAV coordinates.
- x_{obs}, y_{obs}, z_{obs} are coordinates of the obstacles.
- d_{limit} is the predefined limit distance to avoid collision.

Since the main objective is to reach the $TargetPoint$ following the shortest possible path without collision, this work also takes into account the battery capacity of the UAV since the shortest path contributes to energy savings.

IV. Proposed Approach

A. Approach Overview

The proposed strategy employs DLR to enable UAVs to navigate complex environments while avoiding collisions. The learning process is based on the Proximal Policy Optimization (PPO) algorithm, a well-established policy gradient method known for its stability and sample efficiency. PPO is particularly suitable for continuous control tasks, making it a strong candidate for UAV navigation in dynamic and unpredictable environments.

The UAV operates within an environment where it must explore the space while avoiding static and dynamic obstacles. The environment is modeled as a Markov Decision Process (MDP), where the state space consists of sensor readings, positional information, and historical trajectory data. The action space allows the UAV to move in multiple directions, adjusting its velocity and orientation accordingly.

The reward function encourages movement towards the target and penalizes collisions with three main components:

- A negative reward for collisions; and
- A path efficiency reward that encourages shorter and more direct trajectories.

This formulation balances safety and efficiency, guiding the UAV toward optimal navigation strategies.

The training pipeline follows an iterative learning process where the UAV interacts with the environment, collects experiences, and updates the policy network accordingly. Training begins with a randomly initialized policy, which is progressively refined using PPO updates. Each training episode consists of multiple steps where the agent selects actions based on the current policy and receives reward feedbacks.

A key component of the training strategy is the rollout buffer, which stores state-action-reward sequences before performing policy updates. This buffer helps stabilize training by providing mini-batches for gradient optimization. To monitor the learning progress, a custom TensorBoard callback is utilized to log key training metrics, including average rewards and policy loss. This allows for real-time evaluation and fine-tuning of hyperparameters as needed. The trained model is then tested in multiple scenarios to assess its generalization capability and robustness to environmental variations.

Once training is completed, the learned policy is deployed in the simulated environment to evaluate its performance. The evaluation process involves multiple test runs, during which the UAV's trajectory, and collision rate are recorded. Statistical analysis of these metrics provides insights into the agent's ability to generalize beyond the training scenarios and operate effectively in unseen environments.

B. Algorithm

Each UAV operates in a partially observable environment and must learn an optimal policy $\pi(a|s)$ that maps states to actions, maximizing the expected cumulative reward over an episode. The observations available to each UAV include positional data and LiDAR sensor readings, providing sufficient information to navigate. The actions passed to the UAVs are movements in two dimensions $\{dx, dy\}$ while the reward function is designed to encourage goal-directed behavior while penalizing undesired actions such as moving away from the target, abrupt trajectory changes, and collisions with obstacles or other UAVs. The reward and penalty values were empirically defined, and the structure is summarized as follows:

- Approaching Target: +10 per step if moving toward the target;
- Moving Away Penalty: -5 per step if moving away from the target;
- Smooth Trajectory Penalty: -0.1 per step to encourage stable motion;
- Collision Penalty: -5 per step if within 2.0 units of another UAV or obstacle; and
- Target Reached Reward: +10 when successfully reaching the goal.

The training is conducted in episodes with a maximum length of 300 steps per UAV, ensuring that the learning process converges to stable policies. The UAVs start at a fixed initial position with spacing constraints and must reach a predefined target position while avoiding collisions.

Each UAV follows an independent policy rather than a centralized one, allowing them to operate in a fully decentralized manner. This multi-policy approach ensures better scalability and adaptability to dynamic environments. During training, PPO maintains a rollout buffer with a batch size of 32 and collects trajectories for 64 steps before updating the policy over 10 epochs, ensuring efficient learning.

The training process is performed using a total of 512 time steps (for initial testing, with planned expansion to 40 million steps). The training runs on GPU-accelerated computations to leverage faster policy updates and improve efficiency. Each policy is saved periodically, allowing for inference and evaluation of learned behaviors in different test environments.

By employing a DRL-based decentralized control approach, the UAV fleet can navigate complex environments autonomously, demonstrating effective collision avoidance and target-reaching behaviors. The proposed PPO-based multi-policy strategy ensures robustness and scalability. Algorithm 1 depicts the proposed RL-based approach. This decentralized approach improves system scalability by avoiding a central processing unit that could create a bottleneck in the system, also avoiding the single-point-of-failure risk.

Algorithm 1 Reinforcement Learning-Based UAV Navigation

```
Initialize environment $env$ with $N$ UAVs
Load PPO model with policy $\pi_\theta$
Initialize TensorBoard logging
while not converged do
    Reset environment: $s_0 \leftarrow env.reset()$
    for each episode do
        for each UAV $i$ do
            Observe state $s_t^i$
            Select action $a_t^i \sim \pi_\theta(s_t^i)$
        end for
        Execute joint action $\mathbf{a}_t$ in $env$
        Receive new state $\mathbf{s}_{t+1}$ and reward $r_t$
        Store $(s_t, a_t, r_t, s_{t+1})$ in replay buffer
    end for
    Perform PPO optimization step
    Log metrics: reward, collisions, exploration rate
end while
Save trained model $\pi_{\theta^*}$
Inference Phase:
Load trained model $\pi_{\theta^*}$
Reset environment
for each test episode do
    for each UAV $i$ do
        Observe state $s_t^i$
        Select action $a_t^i \sim \pi_{\theta^*}(s_t^i)$
    end for
    Execute $\mathbf{a}_t$ in $env$
    Record path length, collisions, exploration rate
end for
```

V. Results and Discussions

This section presents the performance evaluation of the proposed DRL-based solution. First, the simulation setup is described. Then, the convergence of the DRL-based algorithm during training is presented, and finally, the obtained solution is evaluated in comparison with the heuristic one.

A. Simulation Setup

The experiments were conducted in a simulated environment using Gazebo and MAVSDK, which is a collection of libraries that allow communication with MAVLink systems, such as the Iris UAVs that have PX4 flight controllers, offering an API to manage vehicles, obtain information about them, and control their missions and movements [19]. Thus, multiple UAVs navigate within a 20-meter by 20-meter delimited area with obstacles, as can be seen in Figure 2. The objective is to find the path to targets while avoiding collisions with each other and with static obstacles. The implementation used the Stable-Baselines3 framework and the PPO algorithm to train the UAVs in a decentralized manner. The simulated environment includes LiDAR sensors for perception and a control scheme based on velocities in the XY plane, eliminating movements in the Z coordinate.

The algorithm training was performed on a computer with the following hardware specifications: an Intel(R) Xeon(R) CPU E5-2640 with 12GB NVIDIA GeForce RTX 3060 graphics card and 32GB of RAM. A Windows 10 PRO host operating system and Linux WSL2 subsystem running the Ubuntu 22.04 distribution were used. The training was conducted in an environment configured with the following tools and libraries: Stable-Baselines3 reinforcement learning framework, Python 3.10.12 programming language, PyTorch, and CUDA version 12.7. The model training was performed using the GPU to accelerate deep learning calculations. The model architecture was optimized to take advantage of the parallelism capabilities of the NVIDIA GeForce RTX 3060, using the CUDA backend to maximize computational efficiency. Additionally, the low-level API of the PyTorch/TensorFlow library was used to ensure that operations were executed directly on the GPU.

B. Thread and Loop Structure in DRL Training

The training system is proposed in an asynchronous and multi-threaded architecture to ensure the efficient execution of different processes. The execution control structure includes a global asynchronous loop to avoid deadlocks and allow the execution of asynchronous coroutines without interrupting the main flow of the program. A persistent asynchronous loop is created at the start of the execution, and this loop allows multiple asynchronous operations to be executed in parallel without blocking the main training loop. In addition, a main thread is executed and is managed by the *model learn* function, which controls the interaction of the agent with the environment and the optimization of the PPO model. This main loop executes the various training episodes, repeatedly calls *step* on the environment, updates the decision policy parameters using PPO, and monitors the convergence of the training up to total train timesteps. Auxiliary asynchronous threads ensure that UAVs have access to up-to-date data and can perform actions without interfering with training. Separate asynchronous threads are started, such as the sensor thread of updated lidar data, allowing UAVs to have a runtime perception of the environment, which is essential for safe navigation. Another is the connection thread that manages the initial connection and monitoring of UAV communication via MAVSDK. The inner episode loop *step* that is inside the main training loop and in each episode follows the following flow: perform actions in the environment where it moves the UAVs based on the policy decision, collect the position, velocity, and sensor data states, perform the reward calculation, penalizing collisions and unstable trajectories and rewarding the approach to the target. In addition, it checks the need for a reset if it collides or hits the target, updates the PPO policy, and finally checks the thermal condition that is executed when reaching the defined number of steps. In this condition, the final model is saved for future inference use, and the performance metrics are recorded. This structure ensures efficiency and scalability, allowing the coordinated operation of multiple UAVs in a dynamic and challenging environment.

C. Training Setup

For this training, the PPO policy was used with a high number of steps (2048) to allow the agent to collect more data before updating the policy and small batch size (32) to reflect an approach that prioritizes more stable learning. Additionally, a low learning rate of 10^{-5} was chosen to ensure that the policy was adjusted conservatively, avoiding abrupt changes that could lead to suboptimal behavior. A high gamma value (0.99) was used so that the agent would prioritize future rewards - essential for UAV trajectory planning strategies. For this training, a total of 1 million timesteps were defined.

D. Convergence of the DRL Algorithm

During training, the metric $Explained\ Variance$ was used to evaluate the stability and effectiveness of the learned policy. As shown in Figure 3, the Explained Variance value remained around 0.7 after 150,000 timesteps, indicating that the model was able to learn a consistent policy for UAV navigation.

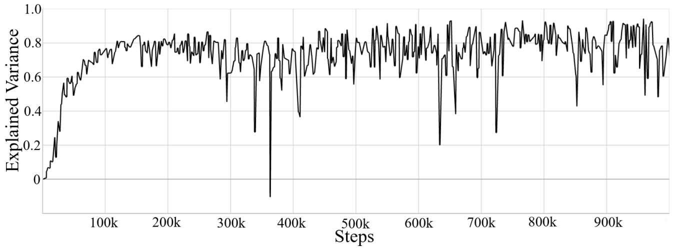

Fig. 3. Convergence of the Explained Variance metric during training.

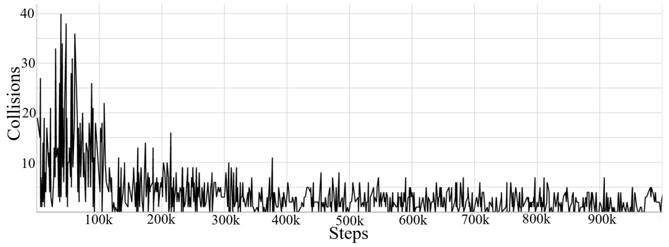

Fig. 4. Evolution of the collision rate over the timesteps.

E. Analysis of the Collision Rate

The reduction in the average number of collisions throughout the training demonstrates the effectiveness of the algorithm. As illustrated in Figure 4, the collision rate started high in the early stages but was reduced to average values between 0 and 2 collisions per episode as the UAVs learned to avoid obstacles and adjust their trajectories.

F. Average Reward Obtained

The analysis of the average accumulated reward per episode shows that the UAVs learned to navigate without straying excessively from the targets or colliding. As shown in Figure 5, the average reward stabilized around 500 points, characterizing that the trained policy maximized the efficiency of the path.

G. Loss Function Analysis

To assess the stability of the learning, the Loss, Policy Gradient Loss, and Value Loss metrics were analyzed. Figure 6 shows the evolution of the loss function during training,

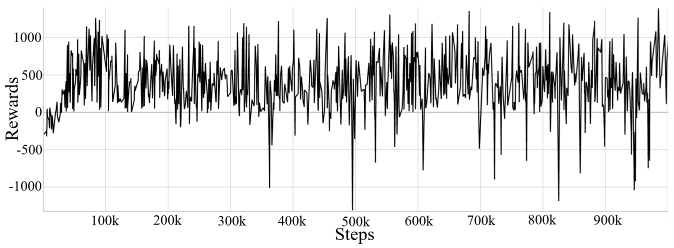

Fig. 5. Evolution of the average reward over the timesteps.

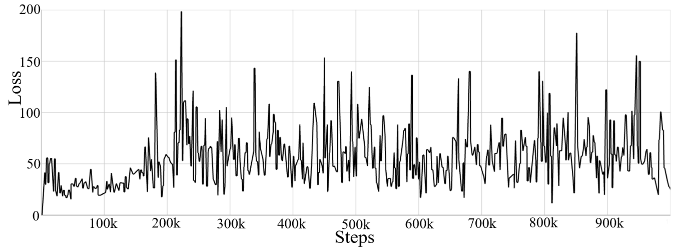

Fig. 6. Evolution of the loss functions throughout the training.

indicating that the model optimization occurred in a controlled manner, without large oscillations that could compromise the stability of the policy.

H. Discussion

The results of this first training demonstrate that the approach could train multiple UAVs to avoid collisions and navigate to the target points in a complex environment. The reduction in the collision rate and the stabilization of the reward suggest that the learned policy is sound, despite the occurrence of a residual number of collisions. To guarantee collision-free operation, fine-tuning the hyperparameters and training in more dynamic scenarios are demanded.

VI. Conclusion

The first results presented in this research demonstrate advances in the DRL-based approach for optimizing navigation and collision avoidance in multi-UAV systems. The PPO algorithm allowed UAVs to learn safe trajectories, significantly reducing collision rates. Integrating LiDAR sensors and decentralized control strategies showed promise for applications in complex environments with fixed and moving obstacles. In addition, the asynchronous training was able to ensure stability and efficiency in learning, allowing data collection from multiple UAVs in a coordinated manner. For future work, the goal is to expand the proposed approach to 3D environments, allowing UAVs to perform more complex maneuvers and improve the efficiency of path planning at different altitudes. In addition, fine-tuning the training hyperparameters can contribute to better learning stability and adaptation of UAVs in more complex scenarios for collision-free navigation. Another direction for future work is to handle failures or inaccuracies of the sensors.

Acknowledgment

This study was financed in part by the Brazilian Federal Agency for Support and Evaluation of Graduate Education (CAPES) - Finance Code 001, partly by the Brazilian National Council for Scientific and Technological Development (CNPq) with the Projects 309505/2020-8 and 311773/2023-0, and partly by the Competence Center for Digital Agriculture Hardware - Senai-RS, with financial resources from the PPI IoT/Manufatura 4.0 / PPI HardwareBR of the MCTI grant number 056/2023, signed with EMBRAPII, and in part by the ELLITT Strategic Research Network, Sweden.

References

[1] M. I. Ilyas, I. Winarno, and D. Pramadihanto, "Drone-to-drone collision detection algorithm in flight traffic management," in *2023 International Electronics Symposium (IES)*. IEEE, 2023, pp. 575–580.

[2] Y.-H. Hsu and R.-H. Gau, "Reinforcement learning-based collision avoidance and optimal trajectory planning in uav communication networks," *IEEE Transactions on Mobile Computing*, vol. 21, no. 1, pp. 306–320, 2020.

[3] J. N. Yasin, S. A. Mohamed, M.-H. Haghbayan, J. Heikkonen, H. Tenhunen, and J. Plosila, "Unmanned aerial vehicles (uavs): Collision avoidance systems and approaches," *IEEE access*, vol. 8, pp. 105 139–105 155, 2020.

[4] L. C. Garaffa, M. Basso, A. A. Konzen, and E. Pignaton de Freitas, "Reinforcement learning for mobile robotics exploration: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 8, pp. 3796–3810, 2023.

[5] D. Wang, T. Fan, T. Han, and J. Pan, "A two-stage reinforcement learning approach for multi-uav collision avoidance under imperfect sensing," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3098–3105, 2020.

[6] S. Ouahouah, M. Bagaa, J. Prados-Garzon, and T. Taleb, "Deep-reinforcement-learning-based collision avoidance in uav environment," *IEEE Internet of Things Journal*, vol. 9, no. 6, pp. 4015–4030, 2021.

[7] B. Ince, V. C. Martinez, P. K. Selvam, I. Petrunin, M. Seo, and A. Tsourdos, "Sense and avoid considerations for safe suas operations in urban environments," *IEEE Aerospace and Electronic Systems Magazine*, 2024.

[8] P. R. Gervi, A. Harati, and S. K. Ghiasi-Shirazi, "Vision-based obstacle avoidance in drone navigation using deep reinforcement learning," in *2021 11th International Conference on Computer Engineering and Knowledge (ICCKE)*, 2021, pp. 363–368.

[9] S.-Y. Shin, Y.-W. Kang, and Y.-G. Kim, "Obstacle avoidance drone by deep reinforcement learning and its racing with human pilot," *Applied sciences*, vol. 9, no. 24, p. 5571, 2019.

[10] K. Jaiswal and A. Vashisth, "A comprehensive analysis of uav collision avoidance techniques for enhanced aerial safety," in *2023 4th International Conference on Computation, Automation and Knowledge Management (ICCAKM)*. IEEE, 2023, pp. 1–6.

[11] X. Liu, A. Prabhu, F. Cladera, I. D. Miller, L. Zhou, C. J. Taylor, and V. Kumar, "Active metric-semantic mapping by multiple aerial robots," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, May 2023. [Online]. Available: http://dx.doi.org/10.1109/ICRA48891.2023.10161564

[12] Y. Tao, Y. Wu, B. Li, F. Cladera, A. Zhou, D. Thakur, and V. Kumar, "Seer: Safe efficient exploration for aerial robots using learning to predict information gain," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, May 2023. [Online]. Available: http://dx.doi.org/10.1109/ICRA48891.2023.10160295

[13] S. Ivić, B. Crnković, L. Grbčić, and L. Matleković, "Multi-uav trajectory planning for 3d visual inspection of complex structures," *Automation in Construction*, vol. 147, p. 104709, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0926580522005799

[14] H. Shraim, A. Awada, and R. Youness, "A survey on quadrotors: Configurations, modeling and identification, control, collision avoidance, fault diagnosis and tolerant control," *IEEE Aerospace and Electronic Systems Magazine*, vol. 33, no. 7, pp. 14–33, 2018.

[15] Y. Tao, E. Iceland, B. Li, E. Zwecher, U. Heinemann, A. Cohen, A. Avni, O. Gal, A. Barel, and V. Kumar, "Learning to explore indoor environments using autonomous micro aerial vehicles," 2023.

[16] L. Wang, K. Wang, C. Pan, W. Xu, N. Aslam, and L. Hanzo, "Multi-agent deep reinforcement learning-based trajectory planning for multi-uav assisted mobile edge computing," *IEEE Transactions on Cognitive Communications and Networking*, vol. 7, no. 1, pp. 73–84, 2021.

[17] D. Wang, T. Fan, T. Han, and J. Pan, "A two-stage reinforcement learning approach for multi-uav collision avoidance under imperfect sensing," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3098–3105, 2020.

[18] Z. Xue and T. Gonsalves, "Vision based drone obstacle avoidance by deep reinforcement learning. ai 2021, 2, 366–380," 2021.

[19] MAVSDK, "MAVSDK - MAVLink API Library," 2025, accessed: 2025-02-07. [Online]. Available: https://mavsdk.mavlink.io/main/en/index.html