Towards Scalable Collision Avoidance in Dense Airspaces with Deep Multi-Agent Reinforcement Learning

Thibault Roux
DTIS, ONERA, Université de Toulouse
Toulouse, France
thibault.roux@onera.fr

Filipo S. Perotto
DTIS, ONERA, Université de Toulouse
Toulouse, France
filipo.perotto@onera.fr

Gauthier Picard
DTIS, ONERA, Université de Toulouse
Toulouse, France
gauthier.picard@onera.fr

ABSTRACT

Increasing airspace congestion requires the development of robust collision avoidance systems to mitigate the risk of near mid-air collisions between aircraft. The *Airborne Collision Avoidance System-X* (ACAS-X) is a next-generation solution that provides both better conflict resolution maneuvers and fewer unnecessary actions compared to the conventional equipment (TCAS-II) currently used in most commercial and general aviation aircraft. ACAS-X is achieved through dynamic programming for one-to-one aircraft encounters. However, this solution still faces significant limitations, in particular the restriction to single intruder scenarios and the reliance on discretized state and action spaces. In this paper, we show that the naive application of ACAS-X to multi-threat scenarios leads to suboptimal and even catastrophic results. To address these issues, we formalize the multi-agent aircraft collision problem and argue for the adoption of deep multi-agent reinforcement learning (MARL) techniques, which have the potential to compute optimal maneuvers in complex multi-aircraft scenarios. Finally, we identify key challenges and open research questions for the multi-agent aircraft collision avoidance problem.

KEYWORDS

Collision Avoidance, Unmanned Aerial Vehicles, Deep Reinforcement Learning, Aeronautics, Multi-Agent Reinforcement Learning

1 INTRODUCTION

The rapid densification of airspace, fueled by emerging technologies such as unmanned aerial vehicles (UAVs), presents significant challenges for collision management. In some previsions, these systems can become prevalent in future smart city environments, supporting applications such as urban air taxis, delivery services, and infrastructure monitoring [28]. This increasing operational density underscores the critical need for robust and reliable collision avoidance systems to prevent mid-air collisions and ensure safety. Our study focuses specifically on fixed-wing UAVs, but the ideas can be generalized to all types of aircraft systems, from general aviation to propeller UAVs. In this paper, we address the critical task of collision avoidance in a dense airspace. We define the aircraft controlled by the focal agent as the "own aircraft", while all other surrounding aircraft are referred to as "intruders".

The Airborne Collision Avoidance System (ACAS-X) [10, 15, 17–19] is a next-generation solution designed to enhance aviation safety by overcoming the limitations of the currently deployed equipment, Traffic Collision Avoidance System II (TCAS-II) [11]. ACAS-X reduces both the risk of mid-air collisions and the frequency of unnecessary maneuvering alerts [20], enabling collision avoidance through cost tables computed by dynamic programming over a stochastic process that models an airspace encounter. ACAS-Xu [9, 32] is its specific version designed for unmanned fixed-wing aircraft. Nevertheless, ACAS-Xu is inherently limited by its design. On the one hand, it relies on simplified, discretized state and action spaces [34]. On the other hand, it assumes a single-intruder scenario, which is extrapolated for the multi-intruder case. While these assumptions allow for efficient computation via dynamic programming, they become dangerous in dense airspace environments with multiple intruders, where the state space grows exponentially with the number of aircraft. Thus, this method has its applicability compromised in the context of multiple aircraft operating in a dense airspace. Other previous studies have extended ACAS-X methods to multiple intruders [23, 31, 36], but always by decomposing the problem into single-intruder subproblems, which leads to suboptimal solutions. This highlights the need for a paradigm shift towards more comprehensive methods capable of handling the complexity of multi-intruder scenarios.

To achieve solutions that generalize to an unknown number of intruders, deep reinforcement learning (DRL) emerges as a promising approach. By exploiting function approximation through deep neural networks, DRL can be very effective in environments with large or continuous state and action spaces, making it well suited for the intricacies of dense airspace collision avoidance. Deep multi-agent reinforcement learning (MARL) has demonstrated its effectiveness in collision avoidance for robotics, allowing agents to learn coordination directly from raw sensor data [26]. Similarly, MARL has been successfully applied to UAV swarm control, where decentralized agents must cooperate in dynamic environments [2, 5]. These achievements position MARL as a relevant approach for addressing multi-agent collision avoidance in dense airspace.

In Section 2, we describe a standard aircraft collision avoidance scenario and present a multi-agent adaptation of the single-intruder ACAS-Xu solution, which highlights the necessity for more robust aircraft coordination approaches. Building on this, Section 3 formalizes the aircraft collision avoidance problem in its full complexity and advocates for MARL methods as a promising solution. Section 4 discusses the challenges inherent to multi-agent settings in collision avoidance. Finally, our conclusions are summarized in Section 5.

2 MOTIVATION

The need for MARL-based approaches to collision avoidance can be justified by illustrating the limitations of the current solutions,

Proc. of the Adaptive and Learning Agents Workshop (ALA 2025), Avalos, Aydeniz, Müller, Mohammedalamen (eds.), May 19 – 20, 2025, Detroit, Michigan, USA, ala-workshop.github.io. 2025.

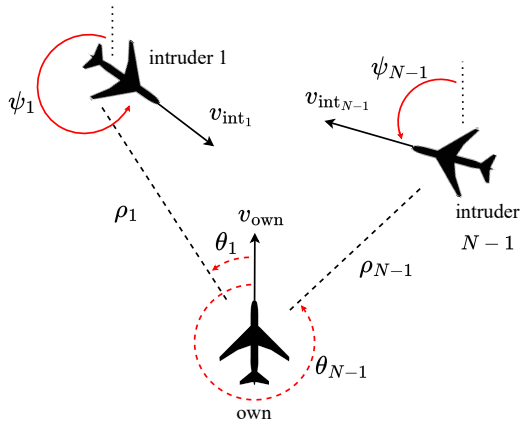

Figure 1: Illustration of the simplified 2D collision avoidance problem with several intruders.

which appear even in a simplified version of the problem. Building on the standard ACAS-Xu framework, let's consider a scenario where N aircraft operate at a fixed altitude in a 2D plane, which corresponds to the ACAS problem limited to its horizontal dimension. This abstraction allows us to focus on the core decision making process while minimizing computational overhead. The objective is to minimize the occurrence of collisions while simultaneously reducing deviations in the aircraft's heading adjustments. The situation is evaluated once per second. At each time step, each aircraft selects a heading adjustment from the set $A = \{0°/s, \pm 1.5°/s, \pm 3°/s\}$, corresponding to no change, a weak turn, or a strong turn to the left or right. A *Near Mid-Air Collision* (NMAC) is occurring when the separation between two aircraft falls to 500 feet or less (about 150 meters). Figure 1 shows the information available to the aircraft to make their decision. Each aircraft i has access to its own velocity v_i, as well as the velocity v_j, distance $\rho_{i,j}$, relative heading $\theta_{i,j}$, and relative bearing $\psi_{i,j}$ to every other aircraft. It also has information about its own last action, a_i. This constitutes an observation vector, available for each aircraft i, corresponding to $x_i = (v_i, a_i, s_{i,1}, \cdots, s_{i,i-1}, s_{i,i+1}, \cdots, s_{i,N})$, where $s_{i,j} = (v_j, \rho_{i,j}, \theta_{i,j}, \psi_{i,j})$ is the relative state of aircraft j as perceived by aircraft i.

2.1 Ad-hoc Solution to a Multi-intruder Context

The ACAS-Xu framework addresses the collision avoidance problem by focusing on scenarios involving a single intruder aircraft [32]. The solution is derived using dynamic programming, which computes an optimal policy for a specially conceived Markov Decision Process (MDP), which represents the stochastic evolution of two aircraft in the same airspace, with predefined simple behaviors for the intruder, highly penalizing with negative rewards a collision, but also including small negative rewards for strong turns and inversions of direction. The resulting table provides a mapping of state-action pairs to corresponding long-term expected costs. Specifically, for any configuration involving one intruder, the table specifies the cost associated with performing a given action in a given state. To explore the potential of extending ACAS-Xu's single-intruder solutions to a multi-intruder context, we propose six heuristic strategies that make use of this cost table. Let that cost table correspond to the function $Q(s, a)$. Also let's define the function $c(i)$ as the one that indicates the closest intruder in relation to aircraft i, so as:

$$c(i) = \underset{1 \le j \le N | j \neq i}{\arg\min} \left[\rho_{i,j} \right]. \tag{1}$$

Closest Intruder. Each aircraft selects its maneuver following the cost table based on the state information of its nearest intruder. Based on this method, the optimal action a_i^* for aircraft i is:

$$a_i^* = \underset{a \in A}{\arg\min} \left[Q(s_{i,c(i)}, a) \right], \tag{2}$$

where A is the set of possible actions (no heading change, weak or strong left or right), and $s_{i,j}$ is the state from the own aircraft when we only consider its closest intruder, $c(i)$, as defined in Eq. (1).

Closest Intruder with Priority. In the previous naive method, there is no coordination between the different agents, thus combined maneuvers can result in catastrophic results. In this second method, each aircraft still selects its maneuver based on the state information of its nearest intruder. However, to avoid potential conflicts that could arise if two aircraft simultaneously adjust their heading, a priority mechanism is implemented in the form of a complete precedence order. In our experiments, the priority is assigned based on aircraft identifiers (lexicographic precedence), so as $i < j \iff i \succ j$. In this way, only the aircraft with the lowest index is allowed to adjust its heading. Using this rule, the optimal action a_i^* is therefore:

$$a_i^* = \begin{cases} \arg\min_{a \in A} \left[Q(s_{i,c(i)}, a) \right] & \text{if } i \succ c(i), \\ 0°/s & \text{otherwise.} \end{cases} \tag{3}$$

Min-Max Cost. In this third method, each aircraft takes into account the cost of all possible actions for every intruder. Then the retained cost for each action is the maximum across all intruders. The chosen action is the one that minimizes this maximum cost over all intruders. This is formalized as:

$$a_i^* = \underset{a \in A}{\arg\min} \left[\max_j \left[Q(s_{i,j}, a) \right] \right]. \tag{4}$$

Weighted Min-Max Cost. This method modifies the min-max cost approach by incorporating distance-based weighting in order to choose the intruder to consider, increasing the importance of closer intruders. Thus, the agent tends to prioritize the most immediate threats. To ensure that $w_{i,j} \in [0, 1], \forall i \forall j$, and that $w_{i,c(i)} = 1$, i.e. the maximal weight will be assigned to the closest intruder, let's consider the following weight function, based on the distances:

$$w_{i,j} = \exp\left(\frac{\rho_{i,c(i)} - \rho_{i,j}}{\rho_{i,c(i)}} \right). \tag{5}$$

The optimal action a_i^* is then given by:

$$a_i^* = \underset{a \in A}{\arg\min} \left[\max_j \left[w_{i,j} \cdot Q(s_{i,j}, a) \right] \right]. \tag{6}$$

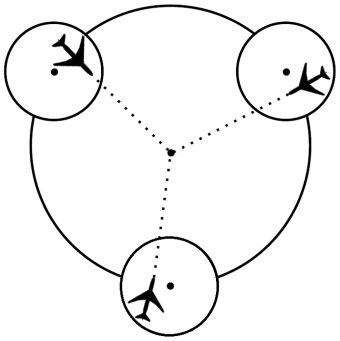

Figure 2: Aircraft random initial positions for an episode with 3 agents in the simulation.

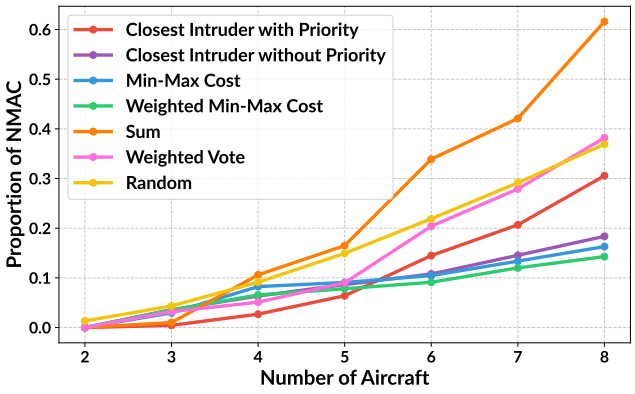

Figure 3: Comparison of the proportion of Near Mid-Air Collisions for each strategy, with regards to the number of agents in the simulation. Each point of the chart represents the proportion of collisions computed through 50 000 random simulations.

Cost Sum. In this method, each aircraft considers, for each possible action, the accumulated cost for every intruder. That is the strategy defined by the official ACAS-Xu specification [9]. The chosen action is the one that minimizes this additive cost over all intruders, which is formalized as follows:

$$a_i^* = \arg\min_{a \in A} \left[\sum_{1 \le j \le N}^{j \neq i} \left[Q(s_{i,j}, a) \right] \right]. \tag{7}$$

Weighted Majority Vote. Finally, in this last method, each aircraft gets the best action for every intruder using the Q table, then chooses the one that is the best for the majority, but considering the weight of each vote proportionally to the distance of each intruder, so as:

$$a_i^* = \arg\max_{a \in A} \left[\sum_{a \in A} \sum_{1 \le j \le N}^{j \neq i} w_{i,j} \cdot 1_{[a = \arg\min_{a \in A} Q(s_{i,j}, a)]} \right], \tag{8}$$

where $w_{i,j}$ follows Eq. (5).

2.2 Performance Comparison

To evaluate the performance of each approach, we developed a comprehensive benchmarking framework by analyzing the statistical results of millions of simulations, varying the number of aircraft, with randomized initialization. That encounter model first places each aircraft uniformly on a circle of 45 km in diameter. Then, the position of each aircraft is perturbed by a random uniform noise from 0 to 5 km, to ensure unique initial configurations at each simulation. That episode initialization is illustrated in the Figure 2. Each aircraft is initially oriented toward the center of the circle, and the speed is kept fixed at 800 km/h in this experiment. Every second, each aircraft i selects the action a_i^* based on the same analyzed strategy. Figure 3 shows, for each strategy, the proportion of Near Mid-Air Collisions (NMACs) among all simulations, as a function of the number of agents in the simulation.

The first key observation is the absence of collisions in the two-agent scenario (a single intruder case) for any of the tested heuristics, which is not the case for the random behavior. This achievement is not surprising since the optimal solution for this setting was computed using dynamic programming. However, as the number of agents increases, NMACs begin to occur in all strategies. This highlights the need for more efficient coordination strategies to enable scalability to larger numbers of agents.

The second important remark is that the strategy "sum", which chooses the action that minimizes the sum of expected long-term costs considering all the intruders, and which is the strategy recommended by the ACAS-Xu official specification, presents a quick deterioration in its safety performance, that can be seen when the number of aircraft is greater than 3, becoming worse than "random". The strategy "weighted vote", which is a common solution to aggregate multi-agent decisions, also presented a quite disappointing performance, approximating "random" when the number of aircraft was greater than 5 in the experiment.

Another notable observation is the reduced proportion of collisions performed by the "closest intruder with priority" strategy when the number of agents ranges from 3 to 5 in that experiment, highlighting the effectiveness of coordination mechanisms, even if simple. However, despite its initial advantage, this strategy exhibits a significant increase in collisions as the number of agents continues to grow, eventually becoming less performant than the others, even if still better than "random".

Finally, the figure highlights the comparable performance of the three other strategies : "closest intruder (without priority)", "min-max cost", and "weighted min-max cost". Among these, the min-max methods show a slight advantage, particularly the weighted version, which consistently outperforms the other approaches when the number of agents exceeds 6 in that experiment. This underscores that using more information allows for more efficient optimal maneuvers, as min-max cost incorporates the costs of multiple intruders, and weighted min-max cost further utilizes distance data.

These results demonstrate the value of heuristic extensions to single-intruder solutions but also highlight their limitations. The lack of explicit coordination between agents allows the occurrence of avoidable collisions and can even induce some others. The use of the distance to deduce the criticality of an encounter has also its own limits. These experimental results support the need for

more sophisticated approaches such as multi-agent reinforcement learning (MARL) to tackle the collision avoidance problem in the presence of several intruders.

3 MARL-BASED COLLISION AVOIDANCE

In this Section, we define the complete multi-agent collision avoidance problem and highlight the advantage of using a deep MARL approach to solve it.

3.1 Multi-agent Aircraft Collision Avoidance

Building on the insights gained from the simplified 2D collision avoidance setup, we now formalize the problem in a more general and realistic context. This extension incorporates 3D motion, continuous actions and partial observability, reflecting the complexity of real-world airspace operations.

Reinforcement Learning typically tries to solve Markov Decision Processes (MDPs). These can be extended to multiple agents using MMDPs [25]. In the context of aircraft collision avoidance, each agent has a partial observation of the whole state and has only access to the states of nearby aircraft. Therefore, the problem can be formulated as a Decentralized Partially Observable Markov Decision Process (Dec-POMDP) [1] which can be formally defined as a tuple $\langle I, S, A, T, R, \gamma, \Omega, O, H, \mu \rangle$ where :

- I is the set of N agents,
- S is the set of k variables $S_1, \ldots, S_k$ defining the state space,
- $A = A_1 \times A_2 \times \ldots \times A_N$ is the joint action space,
- T is the state transition function $T : S \times A \times S \to [0, 1]$,
- R is the global reward function $R : S \times A \times S \to \mathbb{R}$,
- γ is the discount factor, $\gamma \in [0, 1]$,
- $\Omega = \Omega_1 \times \Omega_2 \times \ldots \times \Omega_N$ is the set of joint observations,
- O is the observation function $O : S \times \Omega \to [0, 1]$,
- H is the time-horizon of the problem,
- μ is the initial state distribution $\mu : S \to [0, 1]$.

I is the set of all aircraft operating in the airspace. S is the set of all states representing the infinite set of possible configurations of position, speed, heading, and last action of each aircraft. μ is the initial state distribution, i.e. the encounter model. The joint action space A contains all the combinations of actions for each agent, which are two continuous values for left-right (yaw) and up-down (pitch) heading changes. The dynamics of aircraft are governed by the transition model T which updates positions and heading at each time step based on a simple 3D kinematic model known as the Dubins' aircraft model [6].

The reward function R is designed to penalize unsafe and inefficient behavior: significant costs are assigned to actions that lead to collisions, a lesser penalty is assigned to reversals (e.g., switching from a left turn to a right turn), and a small cost is given for strengthening actions in the same direction (e.g., transitioning from a weak left turn to a strong left turn). Conversely, there is a small reward for maintaining a steady heading, which encourages agents to avoid unnecessary maneuvers [20]. Reference values for immediate rewards are: -1 for collision, -0.0001 for strengthening or reversing, $+0.00001$ for going straight forward.

The observation space Ω is shared by all agents, where $\Omega = \Omega_1^n$. Each agent's observation consists of a partial view of the global state. Specifically, an agent perceives the relative states of nearby agents within a fixed radius r, including their velocity, distance, relative heading, relative altitude, and relative bearing. Moreover, to account for sensor inaccuracy, a zero-mean Gaussian noise is added to each parameter of the perfect observation to reproduce the uncertainty relative to the estimated position and dynamics of intruders [7]. In addition, agents have access to their own speed and the last action they performed.

Finally, γ represents the discount factor, applied to future rewards to account for their decreasing importance over time. The problem runs over a finite horizon H, typically set to 500 time steps, where each time step corresponds to one second of simulated time.

3.2 Deep MARL for Collision Avoidance

Deep learning methods offer a powerful approach to manage the high-dimensional, continuous state space inherent in aircraft collision avoidance, while avoiding discretization assumptions and effectively generalizing to unseen states. Moreover, the real-time and embedded decision process required for this application is consistent with deep MARL, where actions are efficiently computed via a single forward pass through the actor network, which in addition presents a smaller memory footprint, compared to the storage of the entire Q-table.

Furthermore, deep MARL leverages the centralized training and decentralized execution paradigm (CTDE), allowing aircraft to learn coordinated behaviors by sharing global information through a centralized critic during training, and then operating autonomously during execution based solely on their local observations. This framework is particularly suitable for collision avoidance, where each aircraft has only a partial view of the global state. This is typically done in policy gradient actor-critic algorithms with a centralized critic which takes as input global state information to compute either the value or state-value function during learning. During execution, each agent follows its policy through its actor network, which only needs the local observations and not any centralized information. This is particularly useful in our problem because aircraft have to make their decisions based only on their local observations.

Although independent learning (IL), where single-agent RL algorithms are applied to each agent independently, is widely used for MARL collision avoidance [2, 4], policy gradient algorithms with centralized critics, such as MAPPO [38] and MADDPG [27], have demonstrated superior performance in collaborative environments [33] by effectively using shared global information for state evaluation. In addition, the scalability required for the projected density of future airspace can be addressed by algorithms such as MADDPG with attention mechanisms that are effective for controlling swarms of UAVs [5]. The state-of-the-art PPO algorithm [35] performs surprisingly well in multi-agent environments, and was used in [2] for effective control and collision avoidance in UAV swarms, as well as in [4] for aircraft deconfliction in a simulator by increasing or decreasing the speed of aircraft.

Attention mechanisms have been widely applied in collision avoidance methods [2, 4, 5], leveraging their ability to assign varying levels of importance to nearby aircraft based on their potential impact. These mechanisms enable agents to identify and prioritize intruders that pose an imminent collision risk. Often using

recurrent neural networks like LSTMs, attention mechanisms effectively capture spatial and temporal dependencies, making them particularly well-suited for partially observable environments.

4 CHALLENGES AND OPEN QUESTIONS

Developing autonomous aircraft that can safely navigate complex airspace is a challenging task. Multi-agent deep reinforcement learning is a promising approach to address this challenge. However, implementing MARL in aviation raises significant concerns related to system safety, robustness, and scalability. To ensure the safe and reliable operation of autonomous aircraft, we must carefully consider these challenges and explore solutions that can mitigate risks and improve system performance.

How can robust and effective behaviors be developed in a continuously changing environment where other agents change their behavior as they learn? MARL suffers from the non-stationarity of the environment due to the continuous change of each agent's policy during training. Hence, an optimal action in one training step could be suboptimal in the next step because the other agents have changed their behavior. Some recent advances may help to address this issue [24, 30, 37].

How can we scale collision avoidance techniques to handle the increasing density of autonomous aircraft in future airspace? We envision a future where not just dozens, but hundreds or even thousands of autonomous aircraft share the sky. Therefore, the solutions to collision avoidance need to scale efficiently to run in real-time and to scale in terms of memory usage for more modest systems. Current research on collision avoidance for unmanned aircraft tends to study cases with only two to ten agents [2, 4], which is a very low bound compared to what is expected in the future of airspace management. Hierarchical reinforcement learning could appear as a relevant approach. Decomposing the problem into hierarchical levels related to sub-airspace (such as separated flight planes) could help manage complexity and improve scalability [12].

How can we ensure the safety and reliability of autonomous systems in the face of sensor noise and potential adversarial attacks? Solutions must satisfy strong safety requirements on an infinite set of different aircraft configurations. Reinforcement learning with safety guarantees in the mono-agent case is still an active research area, and only a few works have tackled the multi-agent case [8, 13]. The embedded solutions need to be robust to uncertainty in the state estimation to account for sensor inaccuracy and noise [7]. Uncertainty may also come from the behavior of other aircraft, which may be cooperative, non-cooperative, or even adversarial, in which case each aircraft must recognize the intruder's profile and intentions, and must act accordingly. This should require a specific offline and/or online adversarial training protocol [14].

What communication protocols can minimize latency and maximize reliability in real-time, multi-agent systems? Although current systems such as TCAS and ACAS-X rely on transmitted position data, they do not fully exploit the potential of shared messages about action intentions or preferences. Explicit communication can be quite helpful in a multi-agent scenario. These messages can be added to the action space and observed by nearby aircraft to promote better coordination [3]. Agents can share information concerning their perception, their future actions, their flight objectives, and eventually negotiate to choose a satisfying combined set of maneuvers.

How to design a reliable encounter model to test the proposed methods in a multi-aircraft scenario? The difficulty lies in how to test whether the proposed solutions meet safety standards through a reliable encounter model that encapsulates most of the possible cases that could occur in real life. Encounter models for one-to-one interactions [21] and multi-threat situations with up to three aircraft [22] have been well-studied. However, scaling these models to include more aircraft presents a significant theoretical and computational challenge.

How can we make deep learning-based multi-agent systems interpretable and explainable for certification and human-interaction purposes? Airborne systems must be reliable and comply with stringent constraints. In particular, a large open and active issue is the certification of neural networks, which are often considered as black boxes. Although a promising approach has been developed to verify and prove the properties of neural networks for aircraft collision avoidance systems [16], multi-agent explainability is a very under-researched domain, and could open the door to find and understand new ways of cooperation. Plus, integrating the human in the loop of MARL, which is particularly important for airborne and traffic control systems, is still a challenging task [29].

5 CONCLUSION AND NEXT STEPS

Through simulations, we showed that current optimal policies derived via dynamic programming do not scale well as the number of aircraft in the airspace increases, motivating the need for a multi-agent perspective. Thus, we formalized the multi-agent aircraft collision avoidance problem within the framework of a decentralized partially observable Markov decision process (Dec-POMDP). By taking advantage of recent deep multi-agent reinforcement learning (MARL) approaches, we emphasized the potential of these algorithms to learn coordinated behaviors in response to the challenges posed by increasingly dense airspace. However, there are still several open research questions that must be addressed by the multi-agent community, for which investigation tracks such as safe and hierarchical multi-agent reinforcement learning, adversarial training, and intruder intent recognition.

In this paper, we developed an experimental scenario and tested diverse strategies to demonstrate the limits of ad-hoc heuristic solutions that could be used to extend the single intruder collision avoidance solution to the multi-intruder case. In the next steps of this research, we intend to test and adapt different state-of-the-art MARL algorithms to verify whether those techniques can effectively tackle the multi-agent version of the CAS problem, potentially allowing the agents to learn autonomously on how to coordinate their actions, with the advantages of increasing the complexity of the observation and action spaces, not only by allowing to operate directly in the continuous dimensions, but also by adding other informative variables concerning the other agents (flight plan, dynamics, communicated intentions or demands), and an augmented control, including the simultaneous decision over horizontal, vertical, and acceleration/deceleration maneuvers.

REFERENCES

[1] Christopher Amato, Girish Chowdhary, Alborz Geramifard, N. Kemal Üre, and Mykel J. Kochenderfer. 2013. Decentralized control of partially observable Markov decision processes. In *Proceedings of the IEEE Conference on Decision and Control*. https://doi.org/10.1109/CDC.2013.6760239

[2] Sumeet Batra, Zhehui Huang, Aleksei Petrenko, Tushar Kumar, Artem Molchanov, and Gaurav S. Sukhatme. 2021. Decentralized Control of Quadrotor Swarms with End-to-end Deep Reinforcement Learning. In *Proceedings of Machine Learning Research*, Vol. 164.

[3] Rohit Bokade, Xiaoning Jin, and Christopher Amato. 2023. Multi-Agent Reinforcement Learning Based on Representational Communication for Large-Scale Traffic Signal Control. *IEEE Access* 11 (2023), 47646–47658. https://doi.org/10.1109/ACCESS.2023.3275883

[4] Marc Brittain, Xuxi Yang, and Peng Wei. 2020. A Deep Multi-Agent Reinforcement Learning Approach to Autonomous Separation Assurance. arXiv:2003.08353 [cs.LG] https://arxiv.org/abs/2003.08353

[5] Jinchao Chen, Tingyang Li, Ying Zhang, Tao You, Yantao Lu, Prayag Tiwari, and Neeraj Kumar. 2024. Global-and-Local Attention-Based Reinforcement Learning for Cooperative Behaviour Control of Multiple UAVs. *IEEE Transactions on Vehicular Technology* 73, 3 (2024), 4194–4206. https://doi.org/10.1109/TVT.2023.3327571

[6] Hamidreza Chitsaz and Steven M. LaValle. 2007. Time-optimal paths for a dubins airplane. In *Proceedings of the IEEE Conference on Decision and Control*. https://doi.org/10.1109/CDC.2007.4434966

[7] James P. Chryssanthacopoulos and Mykel J. Kochenderfer. 2011. Accounting for state uncertainty in collision avoidance. *Journal of Guidance, Control, and Dynamics* 34 (2011). Issue 4. https://doi.org/10.2514/1.53172

[8] Ingy Elsayed-Aly, Suda Bharadwaj, Christopher Amato, Rüdiger Ehlers, Ufuk Topcu, and Lu Feng. 2021. Safe Multi-Agent Reinforcement Learning via Shielding. *CoRR* abs/2101.11196 (2021). arXiv:2101.11196 https://arxiv.org/abs/2101.11196

[9] EUROCAE. 2020. *ED-275: Minimal Operational Performance for Airborne Collision Avoidance System Xu (ACAS-Xu) – volumes I and II.* Technical Report. EUROCAE.

[10] EUROCONTROL. 2022. Airborne Collision Avoidance System (ACAS) guide.

[11] FAA. 2011. *Introduction to TCAS II version 7.1.* Technical Report. FAA.

[12] Minghong Geng, Shubham Pateria, Budhitama Subagdja, and Ah-Hwee Tan. 2024. HiSOMA: A hierarchical multi-agent model integrating self-organizing neural networks with multi-agent deep reinforcement learning. *Expert Syst. Appl.* 252, PA (July 2024), 11. https://doi.org/10.1016/j.eswa.2024.124117

[13] Shangding Gu, Jakub Grudzien Kuba, Yuanpei Chen, Yali Du, Long Yang, Alois Knoll, and Yaodong Yang. 2023. Safe Multi-Agent Reinforcement Learning for Multi-Robot Control. *Artificial Intelligence* (2023), 103905.

[14] Songyang Han, Sanbao Su, Sihong He, Shuo Han, Haizhao Yang, Shaofeng Zou, and Fei Miao. 2024. What is the Solution for State-Adversarial Multi-Agent Reinforcement Learning? arXiv:2212.02705 [cs.AI] https://arxiv.org/abs/2212.02705

[15] Holland, M.J. Kochenderfer, and Olson. 2013. Optimizing the Next Generation Collision Avoidance System for Safe, Suitable, and Acceptable Operational Performance. *Air Traffic Control Quarterly* 21, 3 (2013).

[16] Guy Katz, Clark Barrett, David L. Dill, Kyle Julian, and Mykel J. Kochenderfer. 2017. Reluplex: An efficient smt solver for verifying deep neural networks. In *LNCS*, Vol. 10426 LNCS. https://doi.org/10.1007/978-3-319-63387-9_5

[17] M.J. Kochenderfer and J.P. Chryssanthacopoulos. 2011. *Robust airborne collision avoidance through dynamic programming. Project Report ATC-371.* Technical Report. MIT, Lincoln Lab.

[18] M.J. Kochenderfer, L.P. Espindle, J.K. Kuchar, and J.D. Griffith. 2008. *Correlated encounter model for cooperative aircraft in the national airspace system. Project Report ATC-344.* Technical Report. MIT, Lincoln Lab.

[19] M.J. Kochenderfer, J.E. Holland, and J.P. Chryssanthacopoulos. 2012. Next-Generation Airborne Collision Avoidance System. *Lincoln Lab Journal* 19, 1 (2012), 17–33.

[20] Mykel J Kochenderfer and JP Chryssanthacopoulos. 2011. Robust airborne collision avoidance through dynamic programming. *Massachusetts Institute of Technology, Lincoln Laboratory, Project Report ATC-371* 130 (2011).

[21] Mykel J. Kochenderfer, Matthew W.M. Edwards, Leo P. Espindle, James K. Kuchar, and J. Daniel Griffith. 2010. Airspace encounter models for estimating collision risk. *Journal of Guidance, Control, and Dynamics* 33 (2010). Issue 2. https://doi.org/10.2514/1.44867

[22] Lincoln Laboratory, Thomas B Billingsley, Leo P Espindle, John Daniel Griffith, Thomas B Billingsley, Leo P Espindle, J Daniel, and Griffi Th. 2009. TCAS Multiple Threat Encounter Analysis. https://api.semanticscholar.org/CorpusID:106711466

[23] Sheng Li, Maxim Egorov, and Mykel Kochenderfer. 2019. Optimizing Collision Avoidance in Dense Airspace using Deep Reinforcement Learning. arXiv:1912.10146 [cs.LG] https://arxiv.org/abs/1912.10146

[24] Wenhao Li, Xiangfeng Wang, Bo Jin, Junjie Sheng, and Hongyuan Zha. 2022. Dealing with Non-Stationarity in MARL via Trust-Region Decomposition. In *International Conference on Learning Representations*. https://openreview.net/forum?id=XHUxf5aRB3s

[25] Michael L. Littman. 1994. Markov games as a framework for multi-agent reinforcement learning. In *Proceedings of the 11th International Conference on Machine Learning, ICML 1994*. https://doi.org/10.1016/B978-1-55860-335-6.50027-1

[26] Pinxin Long, Tingxiang Fan, Xinyi Liao, Wenxi Liu, Hao Zhang, and Jia Pan. 2018. Towards Optimally Decentralized Multi-Robot Collision Avoidance via Deep Reinforcement Learning. arXiv:1709.10082 [cs.RO] https://arxiv.org/abs/1709.10082

[27] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. 2020. Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments. arXiv:1706.02275 [cs.LG] https://arxiv.org/abs/1706.02275

[28] Nader Mohamed, Jameela Al-Jaroodi, Imad Jawhar, Ahmed Idries, and Farhan Mohammed. 2020. Unmanned aerial vehicles applications in future smart cities. *Technological Forecasting and Social Change* 153 (2020). https://doi.org/10.1016/j.techfore.2018.05.004

[29] Eduardo Mosqueira-Rey, Elena Hernández-Pereira, David Alonso-Ríos, José Bobes-Bascarán, and Ángel Fernández-Leal. 2022. Human-in-the-loop machine learning: a state of the art. *Artif. Intell. Rev.* 56, 4 (Aug. 2022), 3005–3054. https://doi.org/10.1007/s10462-022-10246-w

[30] Hadi Nekoei, Akilesh Badrinaaraayanan, Amit Sinha, Mohammad Amini, Janarthanan Rajendran, Aditya Mahajan, and Sarath Chandar. 2023. Dealing With Non-stationarity in Decentralized Cooperative Multi-Agent Deep Reinforcement Learning via Multi-Timescale Learning. arXiv:2302.02792 [cs.LG] https://arxiv.org/abs/2302.02792

[31] Hao Yi Ong and Mykel J. Kochenderfer. 2015. Short-term conflict resolution for unmanned aircraft traffic management. In *AIAA/IEEE Digital Avionics Systems Conference - Proceedings*. https://doi.org/10.1109/DASC.2015.7311424

[32] Michael P. Owen, Adam Panken, Robert Moss, Luis Alvarez, and Charles Leeper. 2019. ACAS Xu: Integrated Collision Avoidance and Detect and Avoid Capability for UAS. In *AIAA/IEEE Digital Avionics Systems Conference - Proceedings*, Vol. 2019-September. https://doi.org/10.1109/DASC43569.2019.9081758

[33] Georgios Papoudakis, Filippos Christianos, Lukas Schäfer, and Stefano V. Albrecht. 2020. Comparative Evaluation of Multi-Agent Deep Reinforcement Learning Algorithms. *CoRR* abs/2006.07869 (2020). arXiv:2006.07869 https://arxiv.org/abs/2006.07869

[34] Filipo Perotto. 2024. Using Deep RL to Improve the ACAS-Xu Policy: Concept Paper. In *Proceedings of the 2nd International Conference on Cognitive Aircraft Systems - Volume 1: ICCAS*. INSTICC, SciTePress, 110–117. https://doi.org/10.5220/0013023200004562

[35] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal Policy Optimization Algorithms. arXiv:1707.06347 [cs.LG] https://arxiv.org/abs/1707.06347

[36] Rachael E. Tompa, Blake Wulfe, Michael P. Owen, and Mykel J. Kochenderfer. 2016. Collision avoidance for unmanned aircraft using coordination tables. In *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*. 1–9. https://doi.org/10.1109/DASC.2016.7777958

[37] Jianan Wei, Liang Wang, Xianping Tao, Hao Hu, and Haijun Wu. 2022. Tackling Non-stationarity Decentralized Multi-Agent Reinforcement Learning with Prudent Q-Learning. In *Web Information Systems and Applications: 19th International Conference, WISA 2022, Dalian, China, September 16–18, 2022, Proceedings* (Dalian, China). Springer-Verlag, Berlin, Heidelberg, 403–415. https://doi.org/10.1007/978-3-031-20309-1_35

[38] Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu. 2022. The Surprising Effectiveness of PPO in Cooperative Multi-Agent Games. In *Advances in Neural Information Processing Systems*, Vol. 35.