Determining Optimal Conflict Avoidance Manoeuvres At High Densities With Reinforcement Learning

Ribeiro, M.J.; Ellerbroek, J.; Hoekstra, J.M.

Publication date
2020

Document Version
Final published version

Published in
10th SESAR Innovation Days

Determining Optimal Conflict Avoidance Manoeuvres At High Densities With Reinforcement Learning

Marta Ribeiro, Joost Ellerbroek and Jacco Hoekstra
Control and Simulation, Faculty of Aerospace Engineering
Delft University of Technology, The Netherlands

***Abstract*—The use of drones for applications such as package delivery, in an urban setting, would result in traffic densities that are orders of magnitude higher than any observed in manned aviation. Current geometric resolution models have proven to be very efficient at relatively moderate densities. However, at higher densities, performance is hindered by the unpredictable emergent behaviour from neighbouring aircraft. In this paper, we use a hybrid solution between existing geometric resolution approaches and reinforcement learning (RL), directed at improving conflict resolution performance at high densities. We resort to a Deep Deterministic Policy Gradient (DDPG) model to improve the behaviour of the Modified Voltage Potential (MVP) geometric conflict resolution method. By default, the MVP method generates avoidance manoeuvres of a geometrically-defined type, using a fixed look-ahead time. In the current study, we instead aim to use RL to determine the values for these variables, based on intruder position and traffic density. The analysis in this paper specifically addresses the difficulty of training algorithms in a cooperative multi-agent case to converge to optimal values. We prove that finding the right representation of state/rewards in a non-stationary environment is non-trivial and highly influences the learning process. Finally, we show that a variation of resolution manoeuvres can improve the safety of several scenarios at high traffic densities.**

***Keywords*—Conflict Detection and Resolution (CD&R), Reinforcement Leaning (RL), Deep Deterministic Policy Gradient (DDPG), Modified Voltage Potential (MVP), U-Space, Unmanned Traffic Management (UTM), BlueSky ATC Simulator**

I. Introduction

To prepare for the introduction of large numbers of mass-market drones, safety automation within unmanned aviation is a priority, as drones must be capable of conflict detection and resolution (CD&R) without human intervention. Both The Federal Aviation Administration (FAA) and the International Civil Aviation Organization (ICAO) have ruled that an Unmanned Aerial System (UAS) must have Sense & Avoid capability in order to be allowed in the civil airspace [1], [2].

In manned aviation research into conflict resolution algorithms, conflict resolution (CR) models based on geometric solutions have proven to be very successful at achieving a high level of safety, with only a minor impact on (path) efficiency. Yet, at the extreme traffic densities envisioned for urban drone applications, such as package delivery, these methods start to suffer from unpredictable interactions that can result in destabilising emergent patterns. For one-to-one conflicts, a set of rules can be defined which leads to implicitly coordinated optimal behaviour. However, as the number of aircraft increases, multi-actor conflicts and knock-on effects can lead to global patterns that cannot be predicted based on these single rules or analytical methods. As a consequence, with these methods we can no longer predict which characteristics lead to optimal behaviour at these higher densities.

Through continuous improvement, reinforcement learning (RL) can potentially identify trends and patterns in this otherwise unpredictable emergent behaviour. By using repetition, RL can adjust to this emergent behaviour, and develop a large set of rules and weights from the knowledge of the environment captured during training. A learning algorithm is used to automatically identify such rules, which can extend the empirical knowledge already present from geometric methods. RL therefore has the potential to help mitigate part of the decline in performance observed in geometric resolution methods when traffic density increases.

Unfortunately, RL also has its drawbacks, such as non-convergence, high dependence on initial conditions, and long training times. A RL model that is completely responsible for the definition of avoidance manoeuvres is therefore unfeasible, as it would have severe issues converging to desirable behaviour. In this paper we hypothesise that, instead, a hybrid approach, combining the strengths of geometric methods and of learning methods, has the potential to mitigate the drawbacks of each of the individual methods. In this approach, rewards are scaled by the efficiency of the resolution model, and the RL starting point is the current performance of the CR model.

When applying RL to mitigate undesirable emergent patterns several questions follow: which states should the RL model know; which CR parameters should the RL model control? Additionally, two problems arise when using RL in cooperative multi-aircraft situations. First, with each action, the next state depends not only on the action performed by the ownship, but on the combination of that action with the actions simultaneously performed by the intruders. Current research [3], [4] shows that emergent behaviour and complexity arise, not as a result of the number of agents, but from the agents interacting and co-evolving. From the point of view of each agent, the environment is non-stationary and, as training progresses, modifies in a way that cannot be explained by the agent's behaviour alone. Furthermore, this non-stationarity limits the straightforward use of *experience replay*; the data in the agent's replay memory quickly becomes obsolete as it no longer reflects the current dynamics of the environment [5]. Second, a certain action may be favourable to the ownship

but may have negative results on the intruding aircraft. The latter may, for example, have to perform bigger deviations from their nominal path to avoid loss of minimum separation with the ownship. An action with good local results may have a negative impact globally. In this paper, we start to explore these questions.

This work will employ the Modified Voltage Potential (MVP) [6] method, which has proven successful in reducing the effect of resolution manoeuvres on flight efficiency while still guaranteeing minimal losses of separation (LoSs) [7]. The Deep Deterministic Policy Gradient (DDPG) RL model [8], which has shown promising results in other studies [9], will be used to determine the look-ahead time and manoeuvre type used by MVP for each conflict resolution situation. Different ways of training the DDPG algorithm will be compared, directly evaluating the impact of rewards based on LoSs, number of conflicts, and time in conflict. Experimental results were obtained through fast-time simulations with open-source, multi-agent ATC simulation tool BlueSky [10].

II. Conflict Resolution Algorithms

A. CR Geometric Model: Model Voltage Potential (MVP)

The geometric resolution of the MVP model, as defined by Hoekstra [6], [11], is displayed in Fig. 1. When a conflict is detected, MVP uses the predicted future positions of both ownship and intruder at the closest point of approach (CPA). These calculated positions 'repel' each other, and this 'repelling force' is converted to a displacement of the predicted position at CPA. The avoidance vector is calculated as the vector starting at the future position of the ownship and ending at the edge of the intruder's protected zone, in the direction of the minimum distance vector. This displacement is thus the shortest way out of the intruder's protected zone. Dividing the avoidance vector by the time left to CPA, yields a new speed, which can be added to the ownship's current speed vector resulting in a new advised speed vector. From the latter, a new advised heading and speed can be retrieved. The same principle is used on the vertical situation, resulting in an advised vertical speed. In a multi-conflict situation, the final avoidance vector is determined by summing the repulsive forces with all intruders. As it is assumed that both aircraft in a conflict will take (opposite) measures to evade the other, MVP is implicitly coordinated.

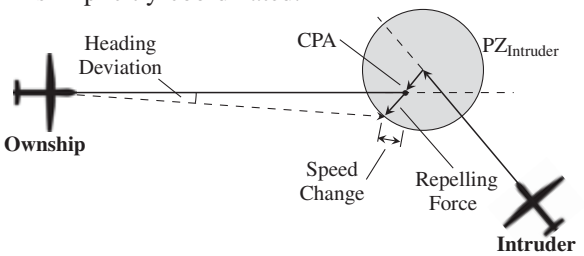

Figure 1. MVP geometric resolution. Adapted from Hoekstra [6].

B. CR Calculation: Look-Ahead Time And Manoeuvre Type

In deterministic conflict avoidance methods, the generated avoidance manoeuvre is often a result of avoiding the detected intruders up to a fixed look-ahead time by resorting to speed, heading and/or altitude variation as pre-defined. Often the same hard-coded values are used for all avoidance calculations. Nevertheless, at high traffic densities, each aircraft will face multiple conflict situations with the number of intruders and their positions varying greatly. However, there is no optimal single set of values for all of these situations. Preference over which values to use for look-ahead time and manoeuvre type may vary based upon the conditions described below.

A resolution manoeuvre can be a heading, speed, or altitude variation. One or multiple of these manoeuvres are performed to follow a conflict free path. Manoeuvres are, in many studies, restricted to the horizontal plane. Sunil [12] showed that, for a stratified airspace, having only horizontal resolutions improves stability; less conflicts are considered and accounted for with only an horizontal conflict layer. However, for resolving short-term conflicts, climb/descent is a fast and efficient action since the required vertical separation is smaller than the horizontal one. A fast vertical change can thus prevent an imminent loss of separation in an airspace where aircraft are predominantly expected to travel in the horizontal plane. Additionally, speed-only variation can be a useful tool in crossing conflicts as it does not require the ownship to occupy more airspace, which is crucial at high traffic densities given the scarcity of airspace.

The look-ahead time used for the conflict resolution defines how far in advance the ownship defends for conflicts. There are advantages and disadvantages for both low and high look-ahead times. With the latter, the ownship initiates resolution manoeuvres with more time in advance, which allows for smaller manoeuvres. For example, as per Fig. 1, a closer CPA results in a bigger heading variation. However, uncertainties increase as the current state of intruders is propagated farther into the future, thus increasing the number of false positive conflicts considered. Moreover, an optimal solution may not exist in situations with multiple intruders. Switching to a lower look-ahead time allows for a prioritization of short-term conflicts. Nevertheless, a solution to a subset of intruders may worsen the conflict with the next layer of intruders.

C. RL Model: Deep Deterministic Policy Gradient (DDPG)

A reinforcement learning model consists of an agent interacting with an environment E in discrete timesteps. At each timestep, the agent receives the current state s of the environment and performs an action a in accordance, for which it receives a reward s_t. An agent's behavior is defined by a policy, π, which maps states to a probability distribution over the actions. The goal is to learn a policy which maximizes the reward. Many RL algorithms have been researched in terms of defining the expected reward following action a. In this work, we use the Deep Deterministic Policy Gradient (DDPG) [8].

Policy gradient algorithms first evaluate the policy, and then follow the policy gradient to maximize performance. DDPG is a deterministic actor-critic policy gradient algorithm, designed to handle continuous and high dimensional state and action spaces. These models have proven to outperform other RL algorithms in environments with stable dynamics

[9]. However, they can become unstable, being particularly sensitive to reward scale settings [13], [14]. As a result, rewards must be carefully defined. Moreover, policy gradient methods tend to exhibit high variance on scenarios which include the coordination of multiple agents [3]. This will be further analyzed with the experimental results.

DDPG uses an actor-critic architecture; it primarily uses two neural networks, one for the actor and one for the critic. The actor function $\mu(s|\theta^{\mu})$ (also called policy) specifies the output action a in regard to the input, the current state s of the environment in the direction suggested by the critic. The critic $Q(s, a|\theta^{Q})$ estimates a correlation between the current state and the action produced by the actor. The critic network is updated from the gradients obtained from a temporal-difference (TD) error signal each time step. The output of the critic drives learning in both the actor and the critic. θ^{μ} and θ^{Q} represent the weights of each network. Updating the actor and critic neural network weights with the values calculated by the networks may lead to divergence. As a result, target networks are used to generate the targets. The target networks are time-delayed copies of their original networks, $\mu'(s|\theta^{\mu'})$ and target critic $Q(s', a|\theta^{Q'})$, that slowly track the learned networks. All hidden neural networks use the non-sigmoidal rectified linear unit (ReLU) activation function, as this has been shown to outperform other functions in statistical performance and computational cost [15].

Experience replay is used in order to improve the independence of samples in the input batch. Past experiences are stored in a *replay buffer*, a finite sized cache R. At each timestamp, the actor and critic are updated by sampling data from this buffer. Because DDPG is an off-policy algorithm, the replay buffer can be large. However, if the *replay buffer* becomes full the oldest samples are discarded. DDPG uses target networks for both the actor, the critic and experience replay. Finally, *exploration noise* is used in order to promote exploration of the environment; an Ornstein-Uhlenbeck process [16] is used in parallel to the authors of the DDPG model.

III. Hybrid Approach: Geometric CR solution + RL

In this work we resort to a hybrid approach combining the strengths of both model-based approaches and RL. The objective is to improve the efficiency of a geometric resolution approach, by having RL determining the optimal look-ahead time and manoeuvre type to be used for every avoidance manoeuvre calculation. This results in avoidance manoeuvres catered to each specific conflict situation. Moreover, having the RL model deciding both these values simultaneously allows for two degrees of freedom where not only the type but the 'magnitude' of the manoeuvre can be controlled.

IV. Experiment: Improving MVP with RL

A. Learning Environment

Aircraft have a pre-defined route with a set of waypoints, and their objective is to reach the final waypoint safely (i.e. no losses of separation). All waypoints are set at the same altitude. The conflict evaluation interval is set to one second; each second, the current conflicts and LoSs are detected, and the necessary avoidance manoeuvres are calculated. When an aircraft is detected to be in conflict, it will receive an action value from the DDPG model.

A sigmoid activation function is used to transform the value calculated by the hidden neural networks into a value within the interval $[0, 1]$, which is then multiplied by 300 to produce a fitting look-ahead value between 0 and 300 seconds. Regarding the manoeuvre type, a softmax activation function is used to select between **S**peed only variation, **A**ltitude only variation, **S**peed + **H**eading variation, and **S**peed + **H**eading + **A**ltitude; the type with the highest probability value is used.

The look-ahead time and manoeuvre type are used to generate the avoidance manoeuvre to be performed by the aircraft for the next 60 seconds. Afterwards, if the aircraft is still in conflict, a new look-ahead time and manoeuvre type are requested with the aircraft's state at that point. A 60 second time period for updating the resolution manoeuvre was considered sufficient to correctly assess the consequences of the manoeuvre, while still allowing the model to perform different manoeuvre types during a conflict situation (instead of one single manoeuvre) until the aircraft is out of conflict.

B. State of the Environment

The state of the environment received by the training model defines the current state of the aircraft in conflict situation. It should provide enough information to allow the RL model to correctly respond to the emergent behaviour. However, we are limited by the information available to each agent, as well as by the computational effort. Within this first stage of exploration, simplicity was preferred. Note that making more information available can potentially improve the results of the model; however, it also increases complexity of the training process and thus decreases the efficiency.

We employ a state representing the location and quantity of intruders. The area surrounding the ownship is divided into four regions regarding the relative bearing to intruders (i.e. front: $[-315, 45]$, right: $]45, 135[$, back: $[135, 225]$, and left: $]225, 315[$), as represented in Fig. 2. The state array's size is equal to 4, where each position on the array corresponds to a different sector. The integer value in each element of the state array represents the number of intruders in that sector. All the intruders can thus be represented within this fixed size array.

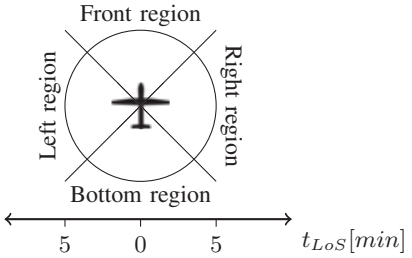

Figure 2. Separation of intruders by relative bearing to the ownship to form the current state of the environment.

C. Reward Model

A policy gradient model tweaks the policy parameters following the gradient towards higher rewards. As a result, the

training model is highly influenced by the reward structure.

In this work, reward is based on the safety of the manoeuvre, which may be based upon the number of LoSs, number of conflicts, and/or time in conflict. However, the relationship between these elements is not trivial: the number of conflicts and LoSs are not directly proportional [7]; reducing conflicts may not have a direct impact on reducing LoSs. Second, the MVP excels in performance as it calculates the minimum deviation required to avoid LoS. A higher deviation to avoid more conflicts may have a negative effect - when free airspace is scarce, the number of LoSs increases when aircraft do not limit the portion of airspace used [7]. Finally, one has to make the decision: how do the penalties for conflicts and LoSs compare? A certain number of conflicts equals having one LoS; this relationship has to be tuned as to prevent the RL model from often choosing to have one LoS in favour of resolving a high number of conflicts. In this work we compare three different reward approaches: (1) only LoSs are considered, the model receives -1 for every loss of separation; (2) both LoSs and conflicts are considered, the model receives -1000 for every LoSs and -1 for every conflict; (3) LoSs and time in conflicts are considered, -100 for every LoSs and -1 for every second in conflict. The magnitude of the previous values were achieved through empirical testing and the best values are herein used.

Note also that negative rewards were employed instead of positive rewards. The reason for this is to motivate the system to reach the destination point of each aircraft as quickly as possible to avoid accumulating penalties. Positive rewards could potentially lead the system to try to accumulate rewards by continuously resolving conflict situations.

D. Apparatus and Aircraft Model

The open Air Traffic Simulator BlueSky [10] was used. This tool has an Airborne Separation Assurance System (ASAS) to which different CD&R implementations can be added; therefore, allowing for all CD&R to be tested under the same scenarios and conditions. The results obtained should be directly associated with using this specific tool and scenarios. Additionally, Bluesky allows for multiple clients working simultaneously while communicating with one server. By implementing the training model in the server, we can have multiple episodes (each client runs one episode at a time) receiving actions and updating the same training model in the server. Such enables a faster training process.

All vehicles in the simulation used a DJI Mavic Pro quadcopter performance model. The data for this model, speed ($-35\,\mathrm{kts}$ to $35\,\mathrm{kts}$) and mass were retrieved from manufacturer data. Although exact turn rate and acceleration/braking values are not available, common values of $15\,^\circ/\mathrm{s}$ and $1\,\mathrm{kts/s}$ were assumed, respectively.

E. Independent Variables

CR model, *traffic density* are set as independent variables.

During training phase, three different *CR* approaches will be experimented with. These differ on the reward given to the RL model: (1) only LoSs are considered; (2) both LoSs and conflicts are considered; (3) both LoSs and time in conflict are considered. These models will be responsible for defining the resolution look-ahead and manoeuvre type to be used by the MVP to generate avoidance manoeuvres. The model with the best performance is then used in the testing phase.

Traffic density ranges from low to high, as shown in Table I. High densities were considered based on the rule of thumb that aircraft should spend more than 10% of their flight time avoiding conflicts [17]. The instantaneous aircraft count defines the number of aircraft expected at any given moment during the measurement period. Given the duration of the measurement and the average flight time, the simulator constantly spawns (adds to the simulation) aircraft at the same rate as they are removed from simulation, in order to keep a constant traffic density. A scenario of 211 instantaneous aircraft is used for training. During testing, the trained RL model will be used with identical traffic densities, as well as with lower and higher traffic densities.

TABLE I. TRAFFIC VOLUME USED IN SIMULATION.

Training Scenarios			
Traffic density $[ac/10\,000\,\mathrm{NM}^2]$		14697	
Number of instantaneous aircraft [-]		211	
Number of spawned aircraft [-]		2442	
Testing Scenarios			
	Low	**Moderate**	**High**
Traffic density $[ac/10\,000\,\mathrm{NM}^2]$	12000	14697	18000
Number of instantaneous aircraft [-]	172	211	259
Number of spawned aircraft [-]	1994	2442	2991

F. Simulation Scenarios

Aircraft fly within a square data collection area, whose dimensions were defined based on the average True Air Speed (TAS), multiplied by the expected average flight time. Aircraft are spawned just outside of the data collection area; this prevents the logging of very short-term conflicts between just spawned aircraft and pre-existing cruising traffic. Spawn locations (origins) are spaced at a distance of the minimum separation distance plus a 10% margin, to avoid conflicts between spawn aircraft and aircraft arriving at their destination. The initial heading of each aircraft varies from 0° to 360°, computed with a normal distribution random number generator. All aircraft are initially set to fly at the same level.

The data collection area is inside a larger square area designated the simulation area. An aircraft is removed from the simulation once it exits this simulation area. This second area is used as we do not want to delete aircraft as soon as they leave the data collection area; they may temporary exit it in case a conflicting manoeuvre demands it to.

Each scenario consists of a build-up period to reach a steady state in terms of traffic volume and traffic pattern. The build-up is followed by the logging phase, during which traffic volume is held constant, and a build-down period, allowing for aircraft created during the logging period to finish their flights. During testing of the model, the experiment is repeated multiple times with different origin-destination combinations. More details are displayed in Table II. No wind was considered.

TABLE II. PROPERTIES OF THE SCENARIOS USED IN SIMULATION.

Training + Testing	
Data Logging Area [NM^2]	144
Simulation Area [NM^2]	250
Minimum Flight Distance [NM]	6
Maximum Flight Distance [NM]	7
Flight Level [ft]	30
Average TAS [kts]	30
Scenario Duration [h]	2
Testing Only	
Number of Repetitions [-]	3

G. Minimum Separation

There is no pre-defined standard separation distance for minimum separation distance for unmanned aviation. However, for horizontal separation, $50\,\mathrm{m}$ to $400\,\mathrm{m}$ are values commonly used in research [18], [19] depending on the properties of the UASs considered. In order to emphasize the effect of RL in losses of separation, a conservative value of $400\,\mathrm{m}$ was used. For vertical separation, $30\,\mathrm{ft}$ was used.

H. Conflict Detection and Resolution

A look-ahead time of five minutes is used for conflict detection in both training and testing. The number of conflicts can thus be directly compared. However, the look-ahead time used for conflict resolution is decided by the training model.

I. Dependent Measures

Two different categories are used to compare the simulated conflict resolution methods: *safety* and *efficiency*. *Safety* is defined in terms of the number and duration of conflicts and LOSs. Naturally, fewer conflicts and LOSs are expected to be safer. *Efficiency* is evaluated in terms of distance travelled and duration of flight. An off-CR situation has better performance in terms of flight distance and time, as the flight path is a straight line. CR models move aircraft out of their intended trajectory in order to avoid conflicts/LOSs; thus, making the path longer. However, a CR model which results in considerable path deviations, significantly increasing the path travelled and/or the duration of the flight is considered inefficient.

V. EXPERIMENT: RESULTS

A. Training the Model

In total, 200 episodes were run; one episode is a full execution of the simulation environment, which runs for 2 hours. The episodes do not all have the same number of calls to the DDPG model, as this is proportional to the number of conflicts throughout the episode. As a reference, between $70\,000$ and $125\,000$ calls were performed in each episode.

Figs. 3 to 5 show the progression of the number of LoSs, number of pairwise conflicts, and average time in conflict per aircraft during training for all the trained models. The models differ in the form of the reward: with **LoSs** only losses of separation are considered; with **LoSs+Conf** both LoSs and conflicts are considered; and lastly **LoSs+TConf** considers both LoSs and time in conflict.

As expected, when the number of conflicts is included in the reward calculation, the model excels in reducing the number of conflicts (see Fig. 4). However, this does not have a direct impact on reducing the number of LoSs (see Fig. 3). In fact, having more conflicts seems to be beneficial for preventing LoS. This behaviour has been previously observed with the MVP method; Hoekstra [6] argues that a moderately positive number of secondary conflicts can be beneficial on a global scale. The effect of sequentially running into a new conflict creates a wave-like pattern, spreading the aircraft out in the available airspace thus 'creating' more airspace. This effect might be particular to the MVP model, however it shows how adding rewards in favour of preventing conflicts may have an adverse effect on the total number of LoSs. This should also be considered with other conflict resolution methods.

Considering either the number of conflicts or time in conflict results in similar reductions of the number of conflicts and average time in conflict (see Figs. 4 and 5). Surprisingly, reducing the number of conflicts results in fewer LoSs than reducing time in conflict (see Fig. 3). In the case of non-simultaneous conflicts, reducing the number of conflicts directly reduces the time in conflict. However, although simultaneous conflicts directly increase the conflict count, they may not impact the total time in conflict. As a result, although the final number of conflicts is similar, it results from preventing different conflicts. To reduce time in conflict, it is preferable to avoid single-conflict situations over some conflicts in multi-conflict situations. The downside of this approach is that it results in more LoSs.

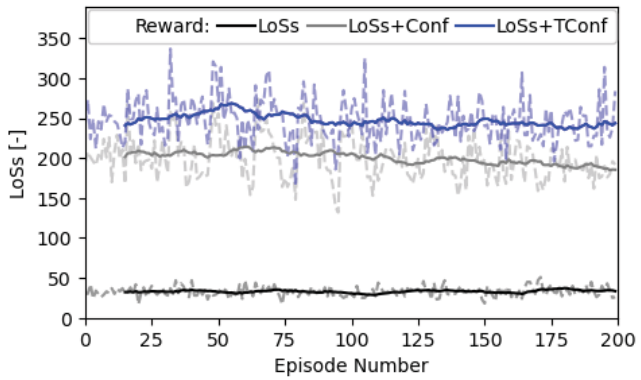

Figure 3. Total number of losses of separation (LoSs) during training.

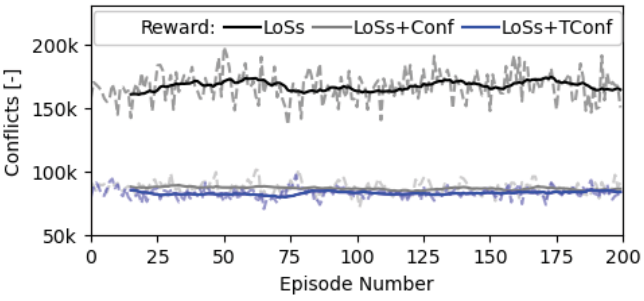

Figure 4. Total number of pairwise conflicts during training.

The non-stationarity of the environment is evident in the observed high variation of total LoSs/conflicts throughout the episodes. In stationary environments, where an agent's

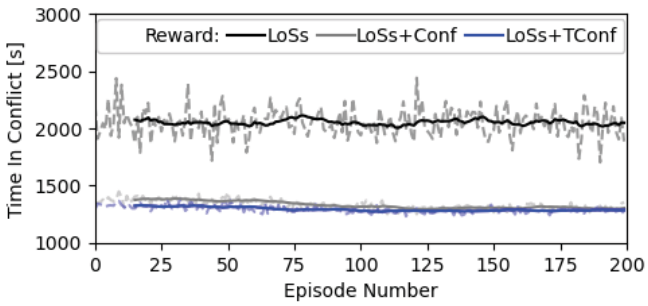

Figure 5. Average time in conflict per aircraft during training.

behaviour is optimized based on its own actions, the RL model is often capable of an almost linear progression. However, in our case, as the agent is also dependent on the actions of the surrounding agents, its optimization exhibits more variability.

Both **LoSs+Conf** and **LoSs+TConf** show an improvement as training progresses, however at a slow pace and marked by a high variation of values. In comparison, **LoSs** does not have a significant improvement over training. According to results, the best reward setting is to focus solely on the number of LoSs. However, the number of LoSs per calls to the RL model might be too sparse to favour a faster convergence to an optimal solution. In the **LoS** only reward situation, almost 99% of the calls to the RL model result in no LoSs. As a result, it is very hard for the model to find occasions to improve (see Fig. 6). In comparison, there is more information regarding conflicts and time in conflict for the models trained with **LoSs+Conf** and **LoSs+TConf** to evolve.

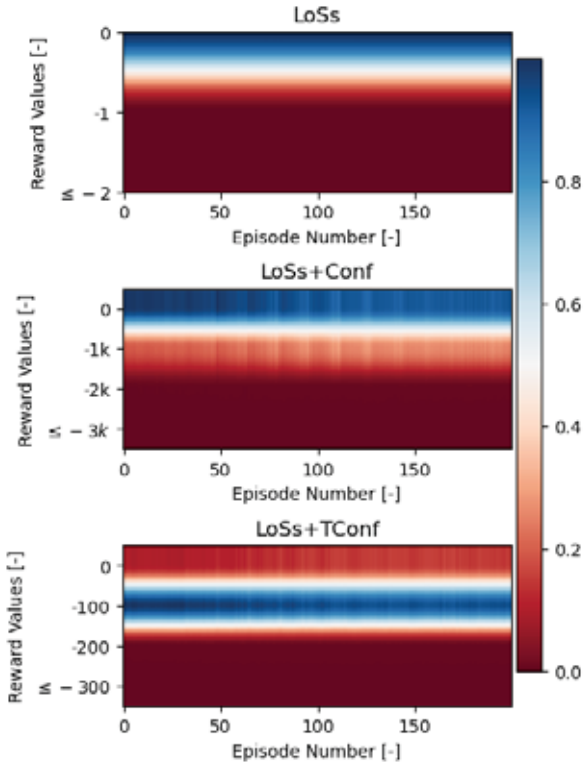

Figure 6. Division of rewards values during training per percentage.

Fig. 7 shows the actions performed by each model in the episode that achieved the fewest LoSs. For clarity, outliers were removed. In the model trained with LoSs only, roughly 99% of the actions of the avoidance manoeuvres are based on **S**peed+**H**eading separation with an optimal look-ahead time of almost 5 minutes. Sporadically, a fast vertical deviation is used. As proven by the LoSs+Conf and LoSs+TConf models, vertical deviation in an environment where aircraft are mostly expected to deviate in the horizontal plane, helps to efficiently resolve horizontal conflicts, as aircraft move out of the main traffic layer. However, the more vertical manoeuvres are performed, the more traffic is dispersed into different layers. Aircraft now have to take into account not only horizontal, but also an increasing number of vertical conflicts. When performed often, it leads to more LoSs as the extra layer of conflicts increases the complexity of the multi-conflict resolution.

For both the LoSs+Conf and LoSs+TConf models, about 80% of the manoeuvres are based on altitude deviation. The LoSs+TConf model has on average shorter look-ahead times than the other models. These allow the model to focus on short-term conflicts, which helps prevent single-conflict situations which in turn reduces the total time in conflict.

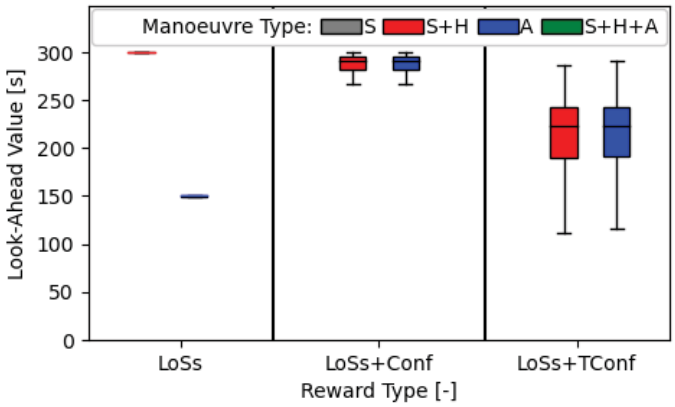

Figure 7. Actions taken by each model in the episode with the fewest LoSs.

B. Testing the Model

The DDPG model trained with only LoSs as a reward, was used with the testing scenarios and is defined with symbol **MVP+RL**. The main objective of having a RL model producing varying avoidance manoeuvres, was to achieve better safety compared to the best combination of a static look-ahead time and manoeuvre type **MVP-Hor**. The latter employs horizontal (speed+heading) only deviation with a resolution look-ahead of 300 seconds. This is the best combination we have previously empirically found and employed in our work [7]. Note that this look-ahead time is directly associated with linear routes in a non-limited space scenario. In a constrained airspace (e.g. an urban environment), smaller look-ahead values are preferable [20]. Results for **MVP-Hor** and **MVP+RL** are also compared with an off-CR situation, where aircraft follow their nominal path without conflict resolution.

The total number of LoSs for all cases is presented in Fig. 8. The manoeuvre variation inputted by the trained RL model

MVP+RL, was able to prevent more losses of separation than the static manoeuvre option **MVP-Hor** for the low and medium traffic densities. The latter, with which the RL model was trained, has the best improvement. However, at high densities it performed worse. Such suggests that the complexity in coordination in higher traffic densities, and consequent emergent behaviour, is not something we can prepare for with lower densities. With higher densities, we expect a change in the most common set of states seen during an episode. The trained RL model was able to optimize manoeuvres for the most common states in the medium density, and is missing optimality in the states representing a higher number of intruders.

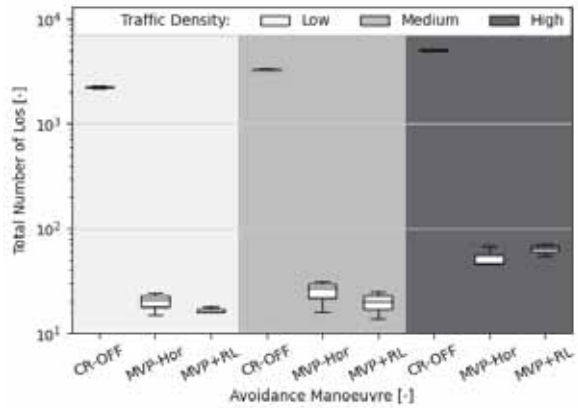

Figure 8. Total number of LoSs of when using: MVP-Hor the best static manoeuvre scenario; MVP+RL the actions defined by the DDPG model.

Fig. 9 shows the total number of pairwise conflicts with the testing scenarios. A pairwise conflict is only counted once, independent of its duration. The increase in number of conflicts, compared to the off-CR situation, is due to secondary conflicts created by the tactical resolution manoeuvres. Interestingly, even tough reduction of conflicts was not part of the reward for the trained model, it was able to diminish the number of conflicts for every traffic density.

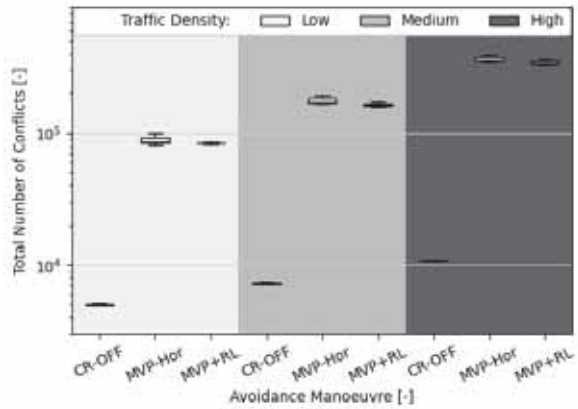

Figure 9. Total number of pairwise conflicts when using: MVP-Hor the best static manoeuvre scenario; MVP+RL the actions defined by the DDPG model.

Fig. 10 shows the amount of time spent in 'conflict mode' per aircraft. An aircraft enters 'conflict mode' when it adopts a new state computed by the CR method. The aircraft will exit this mode, once it is detected that it is past the previously calculated time to CPA. The time to recovery is not included in total time in conflict. With the trained RL model, aircraft spent in average more time in conflict. Such is likely related to a lower look-ahead time for CR; as aircraft initiate avoidance manoeuvres later, conflicts are also resolved later, increasing the total time spent in conflict.

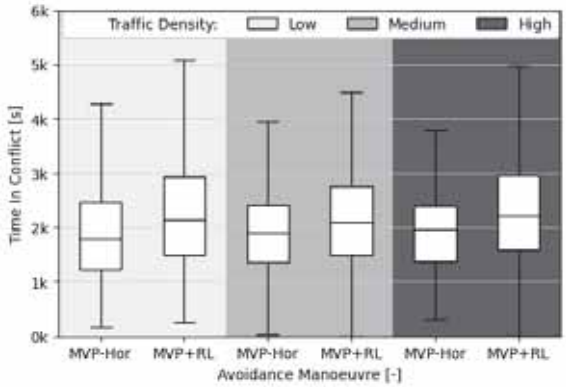

Figure 10. Total time in conflict per aircraft when using: MVP-Hor the best static manoeuvre scenario; MVP+RL the actions defined by the DDPG model.

Figs. 11 and 12 display the extra flight path and time performed by aircraft compared with an off-CR situation. With the trained RL mode, aircraft travelled more to guarantee safety. A bigger path resulted in longer travel times. Given the similarities in values with Fig. 10, this is likely a consequence of the increase in time in conflict. The longer aircraft remained in 'conflict mode', the farther they deviated from their nominal path, increasing their flight time.

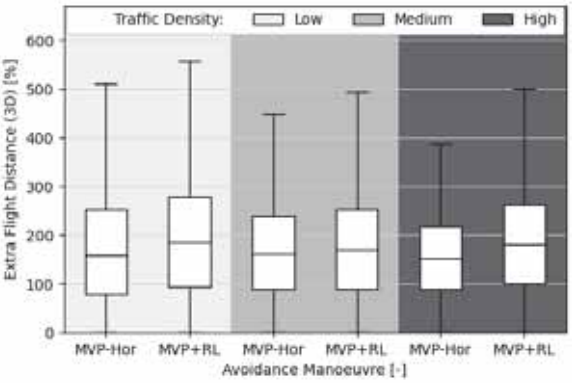

Figure 11. Extra flight path (3D) when using: MVP-Hor the best static manoeuvre scenario; MVP+RL the actions defined by the DDPG model.

VI. Discussion

Similarly to existing work [4], our experimental results show the difficulty of algorithms in a multi-agent environment to converge towards optimal values. Due to the constant changing of actions by the other agents, the environment is non-stationary outside the range of influence of each agent. This results in a difficult and slow convergence to optimal values. In these situations, research into the best state and reward representations is essential.

The final outcome of the trained RL model was as expected given past experience with the MVP method: with hard-coded values, having only horizontal manoeuvres proved

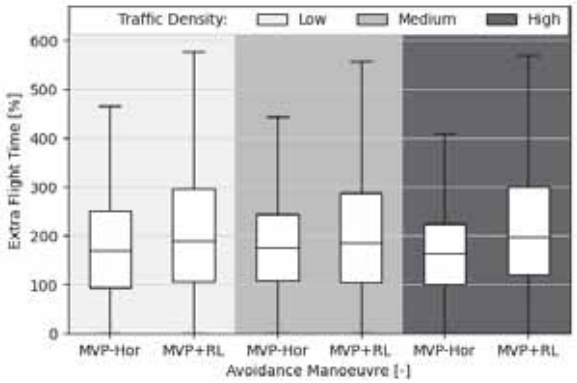

Figure 12. Extra flight time when using: MVP-Hor the best static manoeuvre scenario; MVP+RL the actions defined by the DDPG model.

more efficient safety-wise. Moreover, fast singular vertical manoeuvres can be efficient in an airspace where aircraft are predominantly expected to move with the horizontal plan. However, training in a specific traffic density proved inefficient for higher densities. The model should therefore at least be trained at the highest traffic density that is expected under actual operations. It may also be that different traffic densities require different resolution strategies, as was also hypothesised in the Metropolis project [21]. In this case, the RL model must learn different responses per complexity of emergent behaviour resulting from increasing traffic densities. As always, these results are highly dependent on the conflict resolution method and type of aircraft. A different method may prove more efficient with a different set of manoeuvres. Additionally, the efficiency of all resolutions manoeuvres is dependent on the speed/acceleration of the involved aircraft. RL training with a different resolution method, and/or aircraft type, may produce different results.

The training of the RL model is highly influenced by the reward values used. Within aircraft safety this can be a complex decision: safety is often set in terms of both losses of separation and conflicts. On a first approach, it can be intuitive to consider that fewer conflicts will equal fewer losses of separation. However, multiple studies [6], [7] have shown that focusing on diminishing the number of conflicts may be counter-productive for the MVP. Next to considering LoSs, the best solution may be to consider more efficient manoeuvres, which result in a smaller path deviation. We leave this investigation for future work.

VII. Conclusion

This paper analysed how reinforcement learning (RL) can help surpass the limitations of geometric conflict resolution models. Results show that a variation of avoidance manoeuvres catered to each specific conflict situation is beneficial to improving safety at high traffic densities. More specifically, our trained model found that focusing on horizontal conflict avoidance manoeuvres, with occasional fast vertical deviations, improves prevention of losses of minimum separation.

Translating the success of deep learning on single agent RL to a multi-agent environment continues to be a key challenge. This works constitutes a first exploratory phase with using RL in a cooperative multi-aircraft environment with promising results. Future work will be oriented towards further stabilization of *experience replay* and additional training on a variety of cooperative and competitive multi-aircraft environments.

References

[1] FAA, "FAA Modernization and Reform Act of 2012, Conference Report," FAA, Tech. Rep., 2012.

[2] I. C. A. Organization, "ICAO circular 328 - Unmanned Aircraft Systems (UAS)," ICAO, Tech. Rep., 2011.

[3] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," *Neural Information Processing Systems (NIPS)*, 2017.

[4] L. Matignon, G. J. Laurent, and N. Le Fort-Piat, "Independent reinforcement learners in cooperative Markov games: a survey regarding coordination problems," *The Knowledge Engineering Review*, vol. 27, no. 1, pp. 1–31, feb 2012.

[5] J. N. Foerster, N. Nardelli, G. Farquhar, T. Afouras, P. Torr, P. Kohli, and S. Whiteson, "Stabilising experience replay for deep multi-agent reinforcement learning," in *ICML*, 2017.

[6] J. Hoekstra, R. van Gent, and R. Ruigrok, "Designing for safety: the 'free flight' air traffic management concept," *Reliability Engineering & System Safety*, vol. 75, no. 2, pp. 215–232, feb 2002.

[7] M. Ribeiro, J. Ellerbroek, and J. Hoekstra, "Review of conflict resolution methods for manned and unmanned aviation," *Aerospace*, vol. 7, no. 6, p. 79, jun 2020.

[8] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," in *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*. International Conference on Learning Representations, ICLR, 2016.

[9] P. Henderson, R. Islam, P. Bachman, J. Pineau, D. Precup, and D. Meger, "Deep Reinforcement Learning that Matters," sep 2017. [Online]. Available: http://arxiv.org/abs/1709.06560

[10] J. Hoekstra and J. Ellerbroek, "Bluesky ATC simulator project: an open data and open source approach," in *Conference: International Conference for Research on Air Transportation*, 2016.

[11] J. M. Hoekstra, "Free flight in a crowded airspace?" 2000.

[12] E. Sunil, J. Ellerbroek, and J. M. Hoekstra, "Camda: Capacity assessment method for decentralized air traffic control," in *International Conference on Air Transportation (ICRAT)*, 2018.

[13] Y. Duan, X. Chen, C. X. B. Edu, J. Schulman, P. Abbeel, and P. B. Edu, "Benchmarking Deep Reinforcement Learning for Continuous Control," Tech. Rep., 2016.

[14] R. Islam, P. Henderson, M. Gomrokchi, and D. Precup, "Reproducibility of Benchmarked Deep Reinforcement Learning Tasks for Continuous Control," aug 2017.

[15] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *AISTATS*, 2011.

[16] "On the theory of the Brownian motion," *Physical Review*, vol. 36, no. 5, pp. 823–841, 1930.

[17] R. Golding, "Metrics to characterize dense airspace traffic," Altiscope, Tech. Rep. 004, 2018.

[18] D. Alejo, R. Conde, J. Cobano, and A. Ollero, "Multi-UAV collision avoidance with separation assurance under uncertainties," in *2009 IEEE International Conference on Mechatronics*. IEEE, 2009.

[19] J. Yang, D. Yin, Y. Niu, and L. Shen, "Distributed cooperative onboard planning for the conflict resolution of unmanned aerial vehicles," *Journal of Guidance, Control, and Dynamics*, vol. 42, no. 2, pp. 272–283, feb 2019.

[20] S. C. Johnson, A. Petzen, and D. Tokotch, "Exploration of Detect-and-Avoid and Well-Clear Requirements for Small UAS Maneuvering in an Urban Environment," in *17th AIAA Aviation Technology, Integration, and Operations Conference*. Reston, Virginia: American Institute of Aeronautics and Astronautics, jun 2017.

[21] E. Sunil, J. Hoekstra, J. Ellerbroek, F. Bussink, D. Nieuwenhuisen, A. Vidosavljevic, and S. Kern, "Metropolis: Relating Airspace Structure and Capacity for Extreme Traffic Densities," in *ATM seminar 2015, 11th USA/EUROPE Air Traffic Management R&D Seminar*, Lisboa, Portugal.