Improvement of Conflict Detection and Resolution at High Densities Through 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. However, at the extreme densities envisioned for such drone applications, performance is hindered by unpredictable emergent behaviour of interacting traffic. This paper describes a study that intends to investigate how reinforcement learning techniques can be used to complement geometric methods, thus improving conflict detection and resolution at high traffic densities. Different hybrid approaches are discussed, and preliminary results are shown for a hybrid model that uses geometric methods in the training phase of a Deep Deterministic Policy Gradient (DDPG) model.**

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

I. Introduction

The aviation field must 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. The Federal Aviation Administration (FAA) ruled that an Unmanned Aircraft System (UAS) must have Sense & Avoid capability in order to be allowed in the civil airspace [1]. The International Civil Aviation Organization (ICAO) requires UAV CD&R models to be capable of detection and avoidance in both static and non-static environments. Only after meeting this requirement, will civil-UAVs be allowed to fly beyond the operator's visual line-of-sight [2].

In manned aviation research into conflict resolution algorithms, resolution models based on geometric solutions have proven 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, 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 for multi-conflict resolutions where human observation cannot. By using repetition, RL can adjust to the existent emergent behaviour. Additionally, a RL model can gradually 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. 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. To mitigate such drawbacks, a current trend is to take a hybrid approach, combining RL methods with a model-based approach. In the context of CD&R, a hybrid approach of RL with geometric methods could potentially reduce the uncertainty of RL outcome. Rewards can for instance be scaled by the efficiency of the resolution model. Moreover, by setting the initial state as the current behaviour of the geometric model, training convergence of the RL model could be improved. Our hypothesis is that having each model influence the other, can help tackle their limitations while taking advantage of their strengths. Our objective is thus to create a hybrid solution between existing geometric resolution approaches and RL, directed at improving conflict resolution performance at high densities.

Possible hybrid solutions include resorting to RL to determine an otherwise fixed parameter used in the calculation of an avoidance maneuver. Having the RL model deciding a value which is normally pre-defined, allows for having inputs generated through continuous improvement, catered to each specific conflict situation. These may include the look-ahead time (i.e. how far in advance aircraft detect intruders and initiate avoidance maneuvers); or even which maneuver to use for conflict avoidance: heading, speed, and/or altitude change.

Preference is for the maneuver, or combination of maneuvers, which minimizes flight path and flight time deviation. In addition, each geometrical resolution method has its singularities in which a solution cannot be calculated. Several provisions/rules are put in place for such events; however, we can resort to RL to decide in these cases. RL can thus create a complex mapping of these values to specific behaviours at high densities, potentially generating more efficient rules/procedures to be used by the geometric resolution models.

The aforementioned applications are 'hybrid' approaches, that we intend to research and validate as the focus of the PhD work. As preliminary work, we first looked at how geometric resolution methods can be used to accelerate the training process of a RL method, resulting in a much faster and efficient convergence of the model towards actions with high rewards. This was used to first study how to integrate the RL model into an ATM simulator tool, and to confirm that RL is able of identifying the behaviour of a conflict resolution method and improve it. Results are hereinafter discussed.

II. Experiment: Improving MVP

Preliminary experimental results were obtained by using the geometric Model Voltage Potential (MVP) model during training of a Deep Deterministic Policy Gradient (DDPG) RL model. The objective is to train the *critic*, which in turn, can then guide the *actor* more efficiently towards successful actions. The MVP model has proven to be able to reduce the effect of resolution maneuvers on flight efficiency while still guaranteeing minimal losses of separation (LoSs) [3].

A. Modified Voltage Potential (MVP) Resolution Model

The geometric resolution of the MVP model, as defined by Hoekstra [4], [5], is displayed in Fig. 1. This model makes use of the velocity obstacle theory [6], which defines the set of all velocity vectors of a moving agent which will result in a collision with a moving obstacle at some future point in time. In the MVP model, the calculated positions at the closest point of approach (CPA) 'repel' each other. This 'repelling force' is converted to a displacement of the predicted position at CPA, in a way that the minimum distance will be equal to the required minimum separation between aircraft. Such displacement results in a new advised heading and speed. Additionally, MVP is implicitly coordinated; both aircraft in a conflict will take (opposite) measures to evade the other.

B. Deep Deterministic Policy Gradient (DDPG)

DDPG is a deterministic *actor-critic* policy gradient algorithm, designed to handle continuous and high dimensional state and action spaces. It has proven to outperform other RL algorithms in environments with stable dynamics [7]. However, it can become unstable, being sensitive to reward scale settings [8], [9]; rewards must then be thoroughly adjusted. DDPG

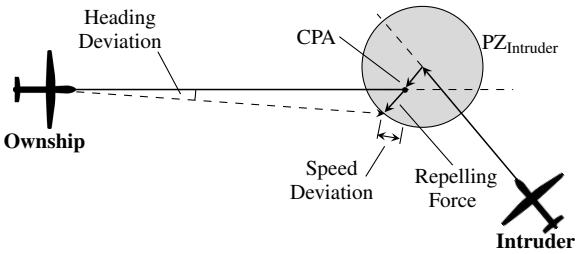

Figure 1. MVP resolution advisory based on geometry at CPA [4].

primarily uses two neural networks, one for the *actor* and one for the *critic*. The *actor* specifies the output action in response to the current state of the environment in the direction suggested by the *critic*. The *critic* estimates a correlation between the current state and the action produced by the *actor*. The output of the *critic* drives learning in both the *actor* and the *critic*. All neural networks, in the experimental simulations herein performed, 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 [10].

C. Apparatus and Aircraft Model

The Open Air Traffic Simulator BlueSky [11] was used. This tool has an Airborne Separation Assurance System (ASAS) to which different CD&R implementations can be added, thus allowing for all implementations to be tested under the same scenarios and conditions. A simulation model of a DJI Mavic Pro quadcopter was used. Speed and mass were retrieved from the manufacturers data, and common values were assumed for turn rate (max: $15^\circ/\mathrm{s}$) and acceleration/breaking ($1.0\,\mathrm{kts}/\mathrm{s}$).

D. Learning Environment

A 2D situation with aircraft flying at the same altitude is used for training. Aircraft have a pre-defined route with a set of waypoints. Their objective is to reach the final waypoint safely (i.e. without LoSs). Every time an aircraft runs into a conflict, it receives a conflict avoidance maneuver, consisting of a new heading and speed, from the DDPG model (Fig. 2). The model then awarded a reward based on the quality on this maneuver.

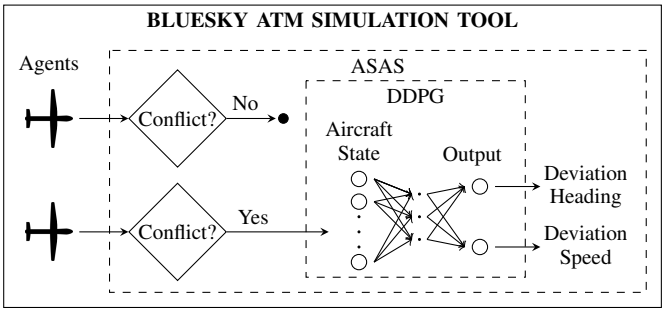

Figure 2. Integration of the DDPG model into Bluesky.

E. State of the Environment

The state received by the DDPG model defines (1) the relative bearing and (2) distance to each intruder. To limit complexity, a fixed input layer size was preferred, and thus there is a limit to the number of intruders represented in the state array. Preliminary experiments were performed with maximum 20 intruders. Given the densities used in testing, a maximum number of 20 'close' intruders at any moment is expected. In the case of fewer intruders, the distance value will be set to a maximum value and the bearing to 180°. It is expected that the model will learn to neglect these values, and to prioritize intruders with the shortest distances to the ownship. With more than 20 intruders, the closest 20 intruders will be considered.

F. Reward Model

Reward is based on both the *safety* and *efficiency*. An optimal reward outputs a maneuver which takes the aircraft out of a conflict situation with minimum deviation from the nominal path. The following components are included in the calculation of the reward: (1) *losses of separation*; (2) *deviation from the desired heading*; (3) *deviation from the desired speed*. The biggest weight coefficient is given to losses of separation as safety is paramount. Next, small differences from the desired heading and speed are preferred. Note that there is no reward based on the final efficiency of the episode. It is not intended for the model to learn how to improve a specific set of trajectories, the objective is for the model to be able to yield an efficient conflict avoidance maneuver for any given conflict situation. Additionally, negative rewards were employed instead of positive rewards to motivate the system to exit a conflict situation 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.

G. Independent Variables

The number of conflict situations, intruders and LoSs is highly dependent on the traffic density and trajectories. We decided not to use a single set of traffic densities and trajectories in order to prevent overfitting. Instead, several traffic densities and trajectories are tested sequentially. Traffic density varies from low to high as per Table I. High densities spend more than 10% of their flight time avoiding conflicts.

TABLE I. TRAFFIC DENSITIES USED IN TRAINING.

	#1	#2	#3	#4	#5
Traffic density [$ac/10000\,\mathrm{NM}^2$]	15000	17838	21213	25227	25226
Number of instantaneous aircraft [-]	337	401	477	567	674
Number of spawned aircraft [-]	1447	1721	2046	2433	2893

III. EXPERIMENTAL DESIGN AND PROCEDURE

For each traffic density, there are three repetitions, each with different trajectories. The model will then, successively, resolve conflict situations in this batch of 15 scenarios (i.e. 5 traffic densities x 3 repetitions each). Improvement of the rewards will be analysed over the repetition of this batch. Each scenario runs for 3 hours.

Aircraft fly at the same altitude of 300 ft. Each aircraft heading is computed with a normal distribution random number generator, varying from 0° to 360°. Total flight distance is uniformly distributed between a pre-defined minimum and maximum value, 15 NM–20 NM, based on the minimum flight time and the average True Air Speed (TAS). Note that no wind was considered. Aircraft fly within a square data collection area, with an area of $250\,\mathrm{NM}^2$. This dimension was defined based on the average TAS and the 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 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 maneuver so demands.

There is no pre-defined standard minimum separation distance for unmanned aviation. However, 50 m–400 m are values commonly used in research [12] depending on the properties of the UAVs considered. For the DJI Mavic Pro, a value of 200 m will be used in these simulations.

IV. PRELIMINARY TRAINING RESULTS

Experimental simulations were performed both with and without using MVP data to train the *critic*, in order to properly evaluate the effect. MVP values were used in the first batch of 15 scenarios. Each scenario is considered an episode. Fig. 3 displays the progression of the reward values throughout the episodes; values are also separated per traffic density for clarity. The model first trained with MVP has a more constant progression, without negative peaks. In both situations, with and without MVP, exploration of the environment was promoted. However, there is always the risk of non-safe exploration leading to states or actions negative for the learning agent. Pre-training the *critic* with MVP appears to have conditioned the model to perform within a safer range of operations. The evolution of LoSs over training is shown in Fig. 4. With MVP training, there are fewer LoSs throughout all episodes. Negative peaks on the model not pre-trained with MVP can also be seen here. However, contrary to expectations, the latter does not have lower reward values in all episodes (Fig. 3). It was observed that this model opted for not deviating

significantly from the nominal path, failing to prevent LoSs, but decreasing the number of encounters with other aircraft (deviating from path usually creates secondary conflicts), resulting in fewer events with negative rewards. The 'peaks' suggest that attempts to prevent the encountered LoSs resulted in worse rewards, discouraging the model from deviating. In comparison, the MVP trained model attempted successfully to solve LoSs. Additionally, it was faster computationally: each iteration took about half of the time in comparison with the variant without MVP pre-training.

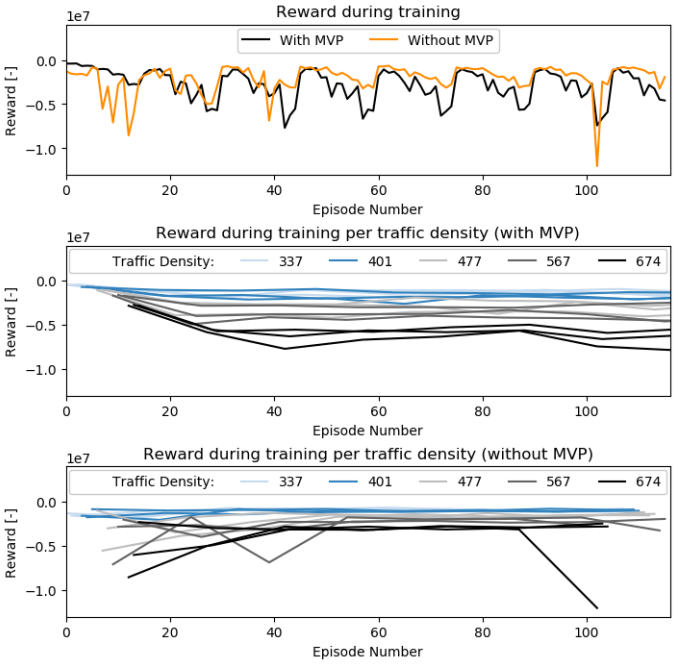

Figure 3. Total reward per episode during training.

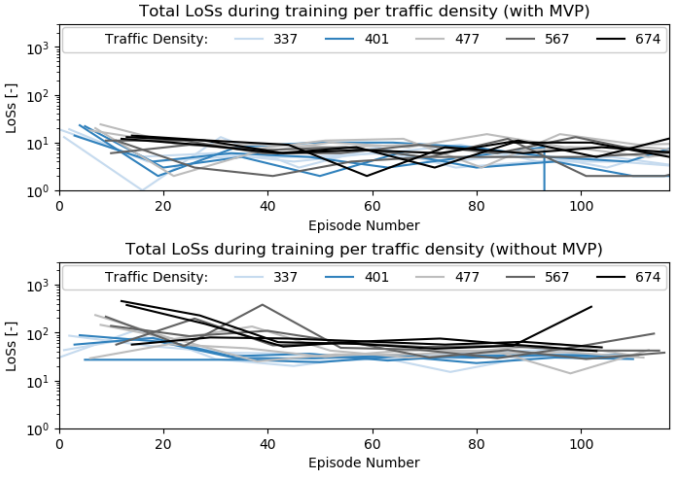

Figure 4. Total LoSs per episode during training.

In spite of the previous positive results, during testing, situational cases were observed which showed that the DPPG model must be further tuned. An example is a typical issue with geometric resolution models: a head-on conflict. The model first resolves in the same way for both aircraft. The conflict is thus not resolved, and both aircraft continue moving in the same direction waiting for the other aircraft to change its trajectory, resulting in a permanent conflict. In the future, factors such as time in conflict, intrusion severity (i.e. how close aircraft get to each other at CPA), and flight path/time deviation, will be added to the reward in order to tackle these situations.

V. Conclusion

We presented a first attempt at creating hybrid solutions combining both reinforcement learning (RL) and known geometric conflict resolution models. Pre-training the *critic* of a Deep Deterministic Policy Gradient (DDPG) model, with values calculated by Modified Voltage Potential method, improved computational speed and stability during the training of the DDPG model. Future work will use this acquired knowledge to optimize training conditions for hybrid solutions, where RL will be used to determine parameters used in a geometric resolution calculation, such as look-ahead time and which maneuver to follow, in order to improve safety and efficiency of conflict resolution at high densities. Furthermore, non-linear trajectories, which are a better representation of routes in an urban settings (e.g. for package delivery), will be studied.

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] M. Ribeiro, J. Ellerbroek, and J. Hoekstra, "Analysis of Conflict Resolution Methods for Manned and Unmanned Aviation Using Fast-Time Simulations," in *9th SESAR Innovation Days*, 2019.

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

[5] 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.

[6] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *The International Journal of Robotics Research*, vol. 17, no. 7, pp. 760–772, jul 1998.

[7] P. Henderson, R. Islam, P. Bachman, J. Pineau, D. Precup, and D. Meger, "Deep Reinforcement Learning that Matters," sep 2017.

[8] 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.

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

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

[11] 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.

[12] 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.