Policy Analysis of Safe Vertical Manoeuvring using Reinforcement Learning: Identifying when to Act and when to stay Idle

Groot, D.J.; Ribeiro, M.J.; Ellerbroek, Joost; Hoekstra, J.M.

Publication date
2023

Document Version
Final published version

Citation (APA)
Groot, D. J., Ribeiro, M. J., Ellerbroek, J., & Hoekstra, J. M. (2023). *Policy Analysis of Safe Vertical Manoeuvring using Reinforcement Learning: Identifying when to Act and when to stay Idle*. Paper presented at 13th SESAR Innovation Days, Sevilla, Spain.

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Policy Analysis of Safe Vertical Manoeuvring using Reinforcement Learning: Identifying when to Act and when to stay Idle.

D.J. Groot, M. Ribeiro, J. Ellerbroek, and J. Hoekstra
Control and Simulation, Faculty of Aerospace Engineering
Delft University of Technology, The Netherlands

***Abstract*—The number of unmanned aircraft operating in the airspace is expected to grow exponentially during the next decades. This will likely lead to traffic densities that are higher than those currently observed in civil and general aviation, and might require both a different airspace structure compared to conventional aviation, as well as different conflict resolution methods. One of the main disadvantages of analytical conflict resolution methods, in high-traffic density scenarios, is that they can cause instabilities of the airspace due to a domino effect of secondary conflicts. Therefore, many studies have also investigated other methods of conflict resolution, such as Deep Reinforcement Learning, which have shown positive results, but tend to be hard to explain due to their black-box nature. This paper investigates if it is possible to explain the behaviour of a Soft Actor-Critic model, trained for resolving vertical conflicts in a layered urban airspace, by interpreting the policy through a heat map of the selected actions. It was found that the model actively changes its policy depending on the degrees of freedom and has a tendency to adopt preventive behaviour on top of conflict resolution. This behaviour can be directly linked to a decrease in secondary conflicts when compared to analytical methods and can potentially be incorporated into these methods to improve them while maintaining explainability.**

***Keywords*—Air Traffic Control, Unmanned Traffic Management, Reinforcement Learning, Policy Analysis, Artificial Intelligence, Explainable AI**

I. Introduction

The demand for (un)manned air traffic operations within urban environments is expected to increase over the course of the following years. One study estimated the potential drone delivery market for Paris to grow to a total of 110 k to 275 k drones operating per hour in the city by 2035 [1]. Even at the lower end of this estimate, this will significantly exceed current aviation operations. As a result, the Federal Aviation Administration (FAA) and the International Civil Aviation Organisation (ICAO) have mandated that drones possess the ability to detect and avoid obstacles without human intervention [2]. The margin by which other aircraft must be avoided is based on a minimum horizontal and vertical separation. If any two aircraft are within these margins of each other, this is called an intrusion. In this research, a conflict between two aircraft means that the distance at the predicted closest point of approach (CPA) between these aircraft is smaller than the required separation margins, indicating a potential future intrusion.

One potential approach to comply with drone operation mandates is to use conventional analytical conflict resolution algorithms such as the Modified Voltage Potential (MVP) Algorithm [3]. However, at higher traffic densities, these algorithms may cause instability through the so-called 'domino effect' [4], which may make them unsuitable for such high-density operations. Alternatively, a learning method that is intrinsically motivated to minimize the number of secondary conflicts resulting from successive resolution manoeuvres could be used. Deep Reinforcement Learning (DRL) is one such method that has been researched in various studies for conflict resolution [5]. However, one main drawback of DRL is the 'black-box problem', which makes it challenging to certify and predict behaviour in all stages of flight. In this paper, we propose training a set of DRL models and analyzing the learned policies to identify patterns in the behaviour. These patterns could explain observed manoeuvres and increase understanding of DRL methods. Additionally, the optimal decisions found by these methods can help enhance analytical methods. This approach could leverage the benefits of DRL in enhancing safety while still maintaining the explainability of analytical methods. We note that this work employs the same experimental setup and methods as our previous publication, which focused primarily on safety [6]. However, this paper emphasises policy analysis and explainability, accordingly, the results are analyzed from a different perspective.

For this research only conflicts that arise during vertical manoeuvres within a layered urban airspace are considered in order to isolate the behaviour and improve explainability. This airspace structure is a result from the Metropolis project, which researched different ways to structure the airspace to enhance operational efficiency and intrinsic safety of the airspace [7]. A layered airspace was found to decrease the number of conflicts through separation of the traffic in different layers (segmentation effect) and by having aircraft flying in similar directions in the same layer (alignment effect) [8]. However, vertically manoeuvring aircraft do not benefit from the separation and alignment effect in layered airspace, resulting in an increase in conflicts and intrusions [9], [10]. Because of this, solely focusing on the vertical conflicts not only aids in the explainability, but is also relevant for improving the safety of the vertical manoeuvres within the airspace.

To identify and highlight how different resolution manoeu-

SESAR Innovation Days
27-30 November 2023
Seville

vres result in different policies with unique strategies, a total of four DRL models with a variety of degrees of freedom will be trained in large-scale simulations, simulating both package deliveries and take-offs. The policies of the models will be evaluated by analysing the selected actions. The performance of the models will also be compared to the MVP model in terms of safety to ensure that the solutions of the methods, and therefore the policies are valid. It is decided to use the MVP algorithm as previous research has shown that it is optimal at resolving conflicts whilst minimizing additional travel distance [11].

II. Problem Formulation

To compare the effectiveness of DRL and the MVP algorithm at resolving vertical conflicts and improving the overall safety of vertical manoeuvres, vertical operations will be simulated in the BlueSky Open Air Traffic Simulator [12]. Drones will be tasked with either climb or descent commands to a specific target layer within this airspace. During these vertical manoeuvres, the goal of the model is to safely control the drone to the target layer while avoiding intrusions with other aircraft, e.g., resolve the existing conflicts before they lead to an intrusion. From now on, individually controlled aircraft will be referred to as agents, whereas the model is used to define which policy is used by these individual agents.

For this research, the DRL and MVP models are further subdivided into different models based on the freedom they have in their actions, as it is currently unknown which set of actions will result in optimal performance. In total three individual actions can be isolated: a change in vertical speed, a change in horizontal speed, and a change in heading. These actions are combined to obtain the following 'sub-models':

- 'v_z', control of the vertical speed only.
- '$v_h + v_z$', control of the vertical and horizontal speed.
- '$v_h + hdg$', control of the horizontal speed and heading.
- '$full$' ($v_h + hdg + v_z$), or 3 degrees of freedom, control of all motions.

III. Methods

Here, the methods used in the experiments will be presented. First, the Markov Decision Processes (MDPs) are formulated as the foundation for the DRL methods. Then the employed DRL algorithm, Soft Actor-Critic (SAC), will be further elaborated. Finally, an overview of MVP, the baseline resolution method, will be given.

A. Markov Decision Process

To ensure that DRL can be used for the defined problem, this problem must first be formulated as an MDP. An MDP is a mathematical framework that can be used for decision-making in systems with uncertainty. An important element of the MDP is the so-called Markov property, which entails that the future states of the system should only be dependent on the current state of the system. For the scenario of conflict resolution with MVP this Markov property holds, as for a specific conflict, the used resolution manoeuvre, and therefore future states, are independent of how these aircraft came to be in conflict. It is therefore assumed that this property also holds for DRL. This allows the problem to be formulated as an MDP, described by the quadruple (S, A, P, R): [13]

1) S, the state space of the system.
2) A, the action space of the system.
3) $P([s, a], s')$, the state transition function.
4) $R(s, a, s')$, the reward function.

The goal of the model is to learn which action $a \in A$ given a state $s \in S$ maximizes the total reward $\sum r \in R$ over all the state transitions $s, a \rightarrow s'$, where s' indicates the next state.

1) Observation Vector

The observation vector is a combination of the ownship states and the (relative) states of the intruders. Note that the number of aircraft in the vicinity of the ownship is variable, but the proposed method requires the observation vector to be constant in size. This means that the problem has to be converted to a partially observable MDP (POMDP), hence why the term observation vector is used instead of state. For this research, it is decided to include 5 aircraft in the observation vector sorted by time until the closest point of approach (T_{cpa}) with a maximum distance at the closest point of approach (D_{cpa}) of 250m, which is 5 times the minimum horizontal separation (PZ_h) between 2 aircraft. This ignores aircraft that are moving away from the agent and only includes the aircraft with the smallest T_{cpa} in the vector, e.g, the aircraft that require the most imminent action. An exception to this is made for aircraft that are in conflict, these are prioritized over other aircraft and are always included in the observation vector, again sorted by T_{cpa}. All the horizontal states considered for the observation vector are given in Figure 1. Apart from this also the vertical distance, D_z, and relative vertical velocity, V_w, are considered.

For the ownship observation, the height difference with the target layer (Δ_h), vertical speed (V_z), horizontal speed (V_{own}) and heading difference with the current layer (Δhdg_{layer}) are used. The final observation vectors for all the models are given in Table I. Not all models are given the same vector as it is assumed that a too-large observation vector containing non-relevant information will negatively impact the required training time of the models.

Finally, all vector elements are normalized using z-score normalization (equation 1), which makes the distribution of all features approximate zero mean and unit variance. The values for σ_s and μ_s are determined by observing 100.000 state transitions. An exception for the normalization of the observation vector is made for the conflict boolean parameter, which is kept as a boolean.

$$S = \frac{s_i - \mu_s}{\sigma_s} \tag{1}$$

2) Action Space

For the action space, the allowable actions and their limits have to be defined. The allowable actions are dependent on

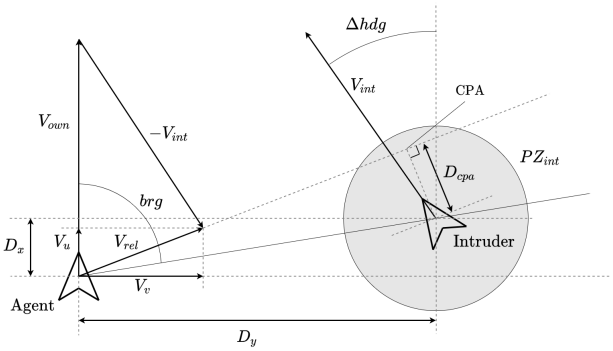


Figure 1. Visualization of the (horizontal) states related to an intruder.

TABLE I. THE RESULTING STATE VECTOR FOR THE DIFFERENT EXPERIMENTS.

	v_z	$v_h + v_z$	$v_h + hdg$	$full$
Ownship States				
V_z	x	x	x	x
V_{own}		x	x	x
Δhdg_{layer}			x	x
Δ_h	x	x	x	x
Intruder States (x5) ↓				
T_{cpa}	x	x	x	x
D_{cpa}	x	x	x	x
Conflict	x	x	x	x
D_z	x	x	x	x
D_x		x	x	x
D_y		x	x	x
V_u		x	x	x
V_v			x	x
brg			x	x
Δhdg_{int}			x	x

the different models, defined in Section II. The limits for the different actions are given in Table II. The increment column indicates the maximum change in action per time step of the simulation. Note that the sign of allowed vertical speed is bounded to the objective of the agent.

TABLE II. ALLOWED RANGE AND INCREMENTS PER TIME-STEP FOR EACH OF THE DIFFERENT ACTIONS.

Action	Range	Increment
Vertical Speed (m/s)	[-5, 5]	[-5, 5]
Horizontal Speed (m/s)	[5, 15]	[-1.5, 1.5]
Heading (deg)	[0, 360]	[-45, 45]

3) State Transition Function

The state transition function is fully determined by the underlying dynamics implemented in the BlueSky Open Air Traffic Simulator. These dynamics follow an open-source kinematic aircraft performance model developed from flight data from Automatic Dependent Surveillance-Broadcast (ADS-B) [14].

4) Reward

It is preferred to keep the reward function as simple as possible while encompassing all the requirements of the solution to the problem [15]. This leads to the reward function given in equation 2. Here s_{target} refers to a state in which the agent is in the corresponding target layer and s_{LoS} is a state in which an intrusion with the agent is present.

$$r = \begin{cases} 1 & \text{s = s}_{target} \\ -1 & \text{s = s}_{LoS} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

B. Deep Reinforcement Learning: Soft Actor Critic

To solve the (PO)MDP defined in section III-A, this study uses the Soft Actor-Critic (SAC) DRL algorithm. SAC is an off-policy, model-free, DRL algorithm, which means that it can learn from past experiences without explicitly knowing the environment dynamics or reward function. The hyperparameters used for this research are given in Table III and the same as the ones used by the original authors with a reward scale of 20 [16].

TABLE III. HYPERPARAMETERS USE FOR THE SAC ALGORITHM.

Parameter	Value
Optimizer	Adam
Learning rate	3e-4
Discount factor (γ	0.99
Memory buffer size	10e6
Sample size	256
Smoothing coefficient (τ)	5e-3
Number of layers	2
Neurons per layer	256
Network update frequency	1

C. Baseline Resolution Algorithm: Modified Voltage Potential

To provide a reference for the performance of the DRL models, all scenarios are also simulated with the MVP conflict resolution algorithm [17]. MVP determines the closest point of approach of two aircraft, and, if the distance between the two aircraft at CPA is smaller than the minimum separation distance, a repelling 'force' is determined which changes the velocity vector such that the shortest way out of the conflict is determined. To ensure a fair comparison between the MVP and the DRL models, the MVP model will have the same constraints on their degrees of freedom imposed as their DRL counterpart.

IV. EXPERIMENTAL SETUP

A. Experimental Scenario

For all conducted experiments, the goal of the agent is to traverse through the different layers in a layered airspace and reach the target layer without intrusions.

The layered airspace in question consists of 2 sets of 8 altitude layers, each having an allowed heading range of 45 degrees, covering all the possible heading angles twice. The purpose of having 2 sets of layers is that long-distance travel can be done at higher speeds in the top layers, whereas short-distance commute is allocated to the slower bottom layers [7]. For this research, however, the different layers function solely as a way to artificially generate the need for vertical manoeuvres. A transition layer is placed between each layer that can only be accessed by aircraft conducting vertical

manoeuvres, which allows the agent to adapt to the new layer before merging. All layers are 7.62m (25ft) in height.

Within this airspace, aircraft operating in the top 8 layers will have a certain probability to obtain a descent command to one of the 8 bottom layers, simulating the delivery of a package. Similarly, aircraft flying in the bottom 8 layers have a probability to get a climb command, simulating the return to a warehouse or place outside of the city. This probability is selected such that on average 5% of the aircraft in the airspace are conducting vertical manoeuvres at any given time. This means that at any given time roughly 5% of the aircraft in the airspace will be controlled by either DRL or MVP.

B. Traffic Density and Conflict Probability

The traffic density in the airspace is selected to be $188AC/km^2$ $(55AC/NM^2)$, equally distributed over all of the heading layers, such he conflict probability between an agent and any other aircraft, based on the equations in Sunil [4], equals 10.0%.

C. Control Variables

1) Simulation time-steps

The simulation is run with time-steps of 1.5 seconds. Thus, the DRL agent selects an action for the aircraft every 1.5 seconds. The MVP agent selects an action for the aircraft every 1.5 seconds only when in conflict.

2) Minimum Separation

The protected zone around all aircraft is set at 50m horizontally (R_{pz}) and 7.62m (25ft) vertically (h_{pz}). These values are based on comparable work [18], as currently no standard for separation requirements has been specified for unmanned aviation.

3) Conflict Detection

For all experiments, instead of look-ahead time use is made of a 'search cylinder' with a radius of 500m, spanning from the agent's altitude to the altitude of the target layer. All aircraft within this cylinder with a $D_{cpa} < PZ_h$ are evaluated for potential conflicts. This is done by comparing the times in and out of the horizontal and vertical minimum separation. If there is an overlap between these times the aircraft are labelled as in conflict. The choice for a look-ahead distance instead of look-ahead time is made to ensure that aircraft that are flying (almost) parallel to the agent, but that are very close in absolute distance, will not be overlooked for state inclusion. This has as a drawback that aircraft with a very high relative speed, and therefore a much smaller T_{cpa} than other aircraft, might initially be ignored.

4) Default Speeds

All cruising aircraft will be flying at a constant horizontal speed of 10m/s. The default vertical speed for the baseline and MVP during climb or descent is 4m/s.

5) Conflict Resolution

Conflict resolution is only performed by aircraft that are vertically manoeuvring. For all aircraft that are not conducting vertical manoeuvres conflict resolution is turned off.

D. Dependent Variables

Three safety metrics are used: the average number of conflicts encountered during a vertical manoeuvre, the average time spent in conflicts, and the average number of intrusions or losses of minimum separation. The latter is the most important as it directly relates to the safety of the operations. The number of conflicts encountered can give a good indication of the relative stability between the different methods and the percentage of time spent in conflict can be related to the efficacy of the performed resolution manoeuvres.

E. Experimental Hypotheses

1) DRL Model Policies

It is hypothesized that all the models that can control the vertical speed will opt for a high mean vertical speed. This hypothesis is based on the findings of Sunil and Tra where it is shown that lower vertical speeds lead to more intrusions [9], [10]. Heading changes on the other hand, are expected to be relatively small in magnitude regardless of the available degrees of freedom, as large heading changes will also change the relative orientation of the observed aircraft by the agent considerably. From a predictability point of view, this is unfavourable, as the agent has less control over the next state.

For the range of magnitudes for the selected actions, the hypothesis is that this range will become smaller when more degrees of freedom are introduced to the model. This is because it can use coupling of the actions to resolve conflicts with smaller changes to individual states.

Finally, it is expected that all policies will be predominantly driven by the conflict boolean in order to minimize flight path deviations. This is because flight path deviations can result in secondary conflicts, which, when assuming a uniform distribution of intrusion probability for all conflicts, will likely also result in an increase in the number of intrusions.

2) Safety

In terms of safety, it is hypothesized that the models with more degrees of freedom will have fewer intrusions than the models with fewer degrees of freedom at the cost of higher training time. This is because it was shown that having more degrees of freedom increases the safety of a DRL model in a lane-changing and merging task on the highway [19]. Simultaneously it is hypothesized that the total number of conflicts will increase due to the Domino Effect of conflict resolution manoeuvres [20].

V. Results

A. Policy Analysis

One of the main issues with DRL, or methods utilizing deep neural networks in general, is the 'black box' that underlies the decision-making process. In an attempt to demystify the behaviour of the trained models, this section will evaluate the policies by analyzing the selected actions based on the T_{cpa} and D_{cpa} of the first aircraft in the state array. For the heading

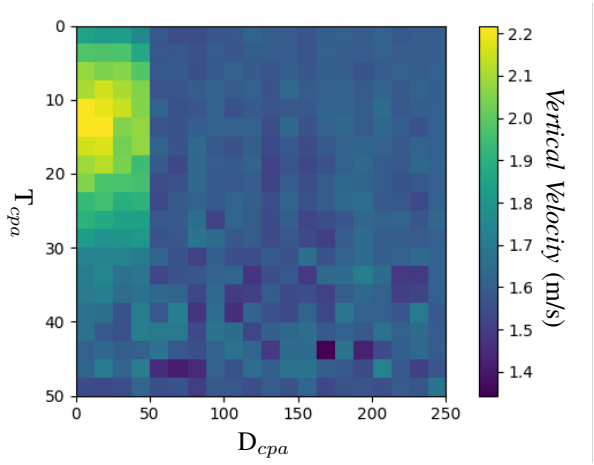


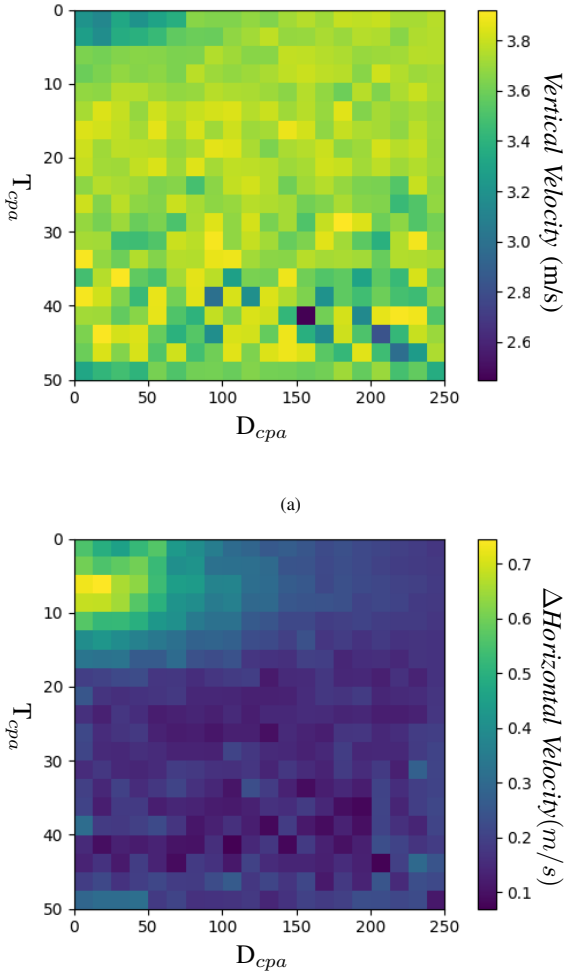
Figure 2. Mean selected vertical velocities for the 'v_z' model. The figure shows a clear separation between conflict and non-conflicting flights, indicated by the sharp line at $D_{cpa} = 50$m.

and horizontal velocity policy maps the mean absolute value is used, as the distribution for both negative and positive values is relatively equal, which would result in a mean close to zero for all values of T_{cpa} and D_{cpa}.

The policy heatmap for the 'v_z' model is given in Figure 2. For the 'v_h+v_z' and 'v_h+hdg' models the policy heatmaps are shown in Figures 3 and 4, respectively. Finally, the heatmap for the '$full$' model can be seen in Figure 5.

Figure 2 shows the selected vertical velocities for the 'v_z' model. In this figure, it is clearly visible that the conflict boolean dictates the behaviour of the model. This is indicated by the fact that the vertical velocity changes based on whether or not $D_{cpa} < 50m$. This is in contrast with the vertical velocities selected by the 'v_h+v_z' and '$full$' methods (Figures 3,5), which do not show this distinction. This can be explained by the fact that the 'v_z' model is unable to influence the value of the D_{cpa} variable, as it is only based on the closest horizontal distance encountered, which is not affected by the vertical velocity. All other methods (Figures 3, 4 & 5) do have control over D_{cpa} through horizontal velocity and heading commands. In these cases, the sharp contrast between conflict and no conflict also disappears in their respective policy heatmaps. This is interesting as there is no intrinsic motivation for the agents to select actions that change the trajectory if there is no conflict, but the behaviour is present in all methods that were able to influence D_{cpa}.

Comparing the policies of the '$v_h + v_z$' and the '$v_h + hdg$' models, which both have an action space of size = 2, a clear difference can also be observed. From the '$v_h + hdg$' policy maps (Figure 4) it can be seen that the policies for the change in horizontal velocity and heading are highly correlated. The '$v_h + v_z$' model (Figure 3) however does not show the same correlation between the two actions. Instead, the model keeps the vertical velocity relatively constant (except for some outliers), and only uses the vertical velocity as a last resort, indicated by the lower values close to $T_{cpa} = 0$ in Figure 3a. This is in line with the hypothesis that a constant



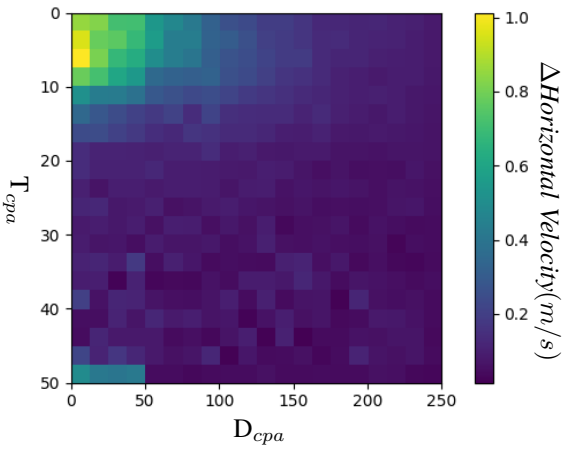
(a)

(b)

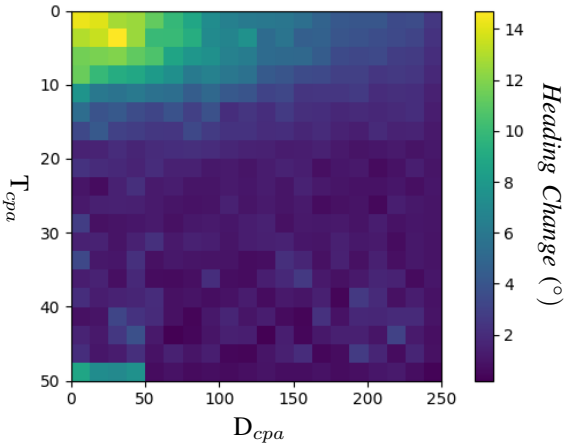
Figure 3. Policy heatmaps for the '$v_h + v_z$' model. (a) Selected vertical velocities. (b) Selected horizontal velocity changes.

higher vertical velocity is more favourable for minimizing the number of intrusions, however, both the 'v_z' and '$full$' have a lower mean vertical velocity which contradicts this hypothesis. Furthermore, the correlation for the heading and horizontal velocity changes present in the '$v_h + hdg$', and to a lesser extent in the '$full$' model, can potentially be explained by the fact that both affect the relative horizontal velocity and can therefore be effectively used in conjunction. However, more research is necessary to strengthen this hypothesis.

Finally, analyzing the policy for the '$full$' method (Figure 5) and comparing it with the policies for the '$v_h + v_z$' and '$v_h + hdg$' methods (Figures 3,4), some differences are again observed. Most notably the policy for the '$full$' method seems to be more homogeneous in the selected action, showing less contrast between clear areas of danger and areas of relative safety. This is also indicated by the lower overall range in magnitudes in the heatmaps. This can partially be explained by the fact that more degrees of freedom allow conflicts to



(a)



(b)

Figure 4. Policy heatmaps for the '$v_h + hdg$' model. (a) Selected horizontal velocity changes. (b) Selected heading changes.

be resolved with fewer extreme actions through a coupling of the actions, and is in line with the initial hypothesis. It can, however, also indicate that it is harder to generate a proper policy with too many degrees of freedom, resulting in less resolute behaviour. This observation is supported by the relatively unstructured policy maps in comparison with the other policies.

B. Safety Analysis

Evaluating the total number of conflicts, which are predicted intrusions and shown in Figure 7, it is observed that for all cases the total number of conflicts encountered during operations increases in comparison to the 'No Resolution' method. This is a frequent occurrence often referred to as the Domino Effect [4], [20]. Essentially, the manoeuvres employed by the agents to resolve conflicts lead to a larger airspace volume being used by the state-based conflict detection method. This in turn increases the number of potential conflict pairs when compared to flying in a straight line. The most significant increase in the number of conflicts is observed in the 'v_z' and '$full$' models, which may be partially attributed to the overall lower vertical speed demonstrated by these models during operations, as illustrated in Figures 2 and 5a. This is because these lower vertical speeds also increase the overall duration of the vertical manoeuvres, which can result in an increase in the number of conflicts encountered. A noteworthy observation is that the aforementioned Domino Effect appears to be less prominent in the '$v_h + v_z$' and '$v_h + hdg$' DRL models compared to their corresponding MVP models. It is hypothesized that this can be attributed to the ability of the DRL models to act when not in conflict. This allows it to increase the D_{cpa} margins whilst not in conflict, allowing more room for resolution manoeuvres that do not result in secondary conflicts.

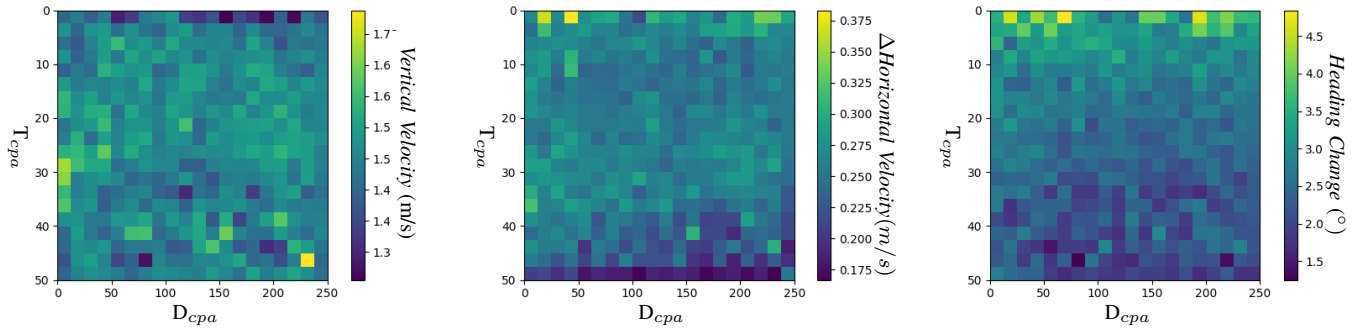
Finally, observing the total number of intrusions per flight given in Figure **??**, it is evident that all methods successfully reduce the total number of intrusions when compared with the no-conflict resolution scenarios. Further inspection of Figure **??** also shows that increasing the degrees of freedom does not necessarily result in a safer policy for the DRL model. This observation is intriguing since the policy of the DRL 'v_z' model constitutes a part of the solution space of the '$v_h + v_z$' and '$full$' models, and similarly, the policy of the MVP models is a part of the solution space of their corresponding DRL models. The performance of the '$v_h + v_z$' and '$full$' models, however, fail to match that of the better available policies, suggesting that these models may be stuck in a local optimum, or require longer training time. This highlights one of the drawbacks of using Deep Reinforcement Learning for higher-dimensional problems. With more actions, the required exploration increases exponentially, increasing the required training time whilst decreasing the guarantee of convergence to the global (or a more optimal local) optimum.

A final remark is that the DRL model found a horizontal resolution method that outperforms the MVP model in terms of safety. As already shown in Section V-A, the DRL models also acted when not in conflict. It is possible that this results in fewer conflicts during vertical manoeuvres and larger buffer areas around other aircraft, which in turn leads to fewer potential conflicts leading to an intrusion and increases the available solution space in the scenarios where conflicts present themselves.

VI. Discussion

A. Policy Analysis

The findings demonstrate significant variability in observed policies for different conflict resolution DRL models, depending on their degrees of freedom. However, the study also found similarities between the '$v_h + v_z$' and '$v_h + hdg$' models, indicating the possible existence of common ground rules that result in fewer intrusions and secondary conflicts. Although it might be difficult to directly implement DRL-based methods as a resolution method due to the black box nature of DRL,



(a) (b) (c)

Figure 5. Policy heatmaps for the '$full$' model. (a) Selected vertical velocities. (b) Selected horizontal velocity changes. (c) Selected heading changes.

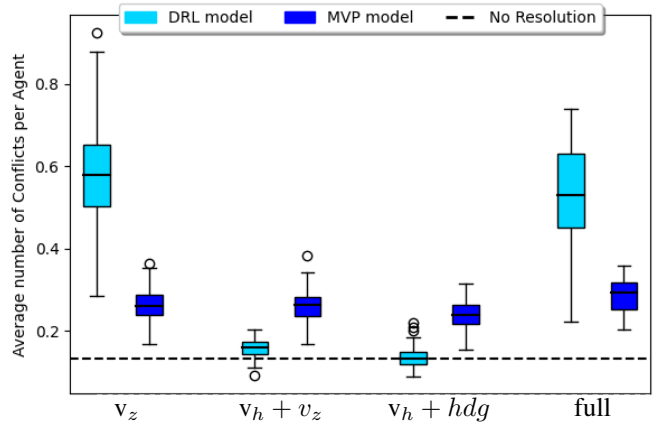


Figure 6. Average number of conflicts encountered during a vertical manoeuvre.

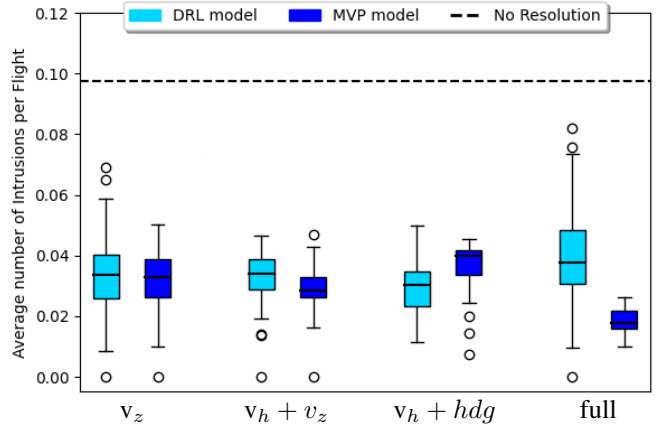


Figure 7. Number of intrusions per vertical manoeuvre.

analyzing the resulting behaviour can still provide valuable insights for enhancing analytical methods.

It is observed that the '$v_h + v_z$' and '$v_h + hdg$' models act even when the D_{cpa} is larger than 50 meters, the conflict cut-off. This behaviour effectively increases the minimum horizontal separation used in decision-making, which is considered to decrease airspace stability by increasing the number of conflict resolution manoeuvres, and thus secondary conflicts [4]. This increase in secondary conflicts is however not observed for these models, indicating that this acting while not in conflict behaviour effectively functions as an additional conflict prevention layer, not present in conventional conflict resolution methods. This has two potential benefits. First, conflict prevention requires smaller deviations from the current flight path than resolving conflicts. Second, this creates larger margins with other aircraft, increasing the available solution space in the case of new conflicts, and potentially reducing the occurrence of multi-conflict scenarios where finding a solution is challenging.

An interesting next step would be to use the observed policies to generate a set of rules or alterations to existing conflict resolution methods, incorporating the trends observed in the DRL methods. If this can be successfully done it will allow conventional analytical methods to benefit from the exploratory nature of DRL methods.

B. Safety Analysis

The safety analysis showed that the DRL model outperforms the MVP model for the '$v_h + hdg$' cases. This difference may be attributed to the lower total number of conflicts observed by the DRL method, as a lower number of conflicts directly influences the total number of potential intrusions. When actually comparing the number of intrusions to the number of conflicts, the MVP model still resolves a higher fraction of conflicts (0.33% vs 0.15% of conflicts resulting in an intrusion for the DRL and MVP method respectively). The main reason for the higher effectiveness of the DRL method for the '$v_h + hdg$' case should therefore be attributed to conflict prevention, rather than conflict resolution. Additionally, the DRL model's capability to execute conflict resolution manoeuvres at different moments depending on the conflict geometry allows for more optimal

conflict resolution timing when compared to the MVP model, which has a fixed resolution time step for all geometries. Previous research has indicated that a constant look-ahead distance or time may not be the most effective approach [21].

The reduction in conflicts is not observed in the v_z and the $full$ model, both of which have a high increase in the number of conflicts. This can be explained by the vertical velocity of these methods shown in Figures 2 and 5a respectively. Both of these methods use a low vertical velocity during 'safe operation', resulting in longer duration of the manoeuvres, and hence more conflicts. This result is in agreement with the results of the study done by Tra. Et al. which found that a higher vertical velocity is better for the overall safety [10]. This also indicates that the models have likely converged to a local optimum, resulting from the fact that a lower vertical velocity results in a lower number of intrusions per unit time, but not per episode. This can potentially be mitigated by increasing the γ hyperparameter to ensure a higher weight is given to future penalties.

C. Limitations and Recommendations

Finally, it is important to acknowledge the limitations of the results due to the experimental setup and the scope of the research. For instance, the traffic scenarios used in the study can be modified to include higher or variable traffic densities and aircraft flying at different cruising velocities. Furthermore, it is challenging to predict the performance of the DRL model in more complex traffic scenarios where not all aircraft adhere to altitude layers, or how the method and corresponding policies would change in non-vertical conflict scenarios such as during cruise or for general/commercial aviation applications. To estimate the true effectiveness of DRL for safe manoeuvring, it should be trained and tested in a variety of different traffic scenarios consisting of operations during all stages of flight (potentially using different models/policies for different conditions) and at various traffic densities.

Another limitation of the research was the non-resolution behaviour of the cruising aircraft, this is analogous to merging on the highway without other road vehicles giving way, and is likely, not optimal in terms of overall safety. Therefore a next step of the research could be to also activate conflict resolution for cruising aircraft. This will remove the stationarity and predictability from the environment and better demonstrate the DRL model's ability to handle emergent behaviour. However, this will also result in a massive multi-agent operation that may adversely affect the stability and duration of training, and will hinder proper policy analysis, as is done in this research, potentially decreasing the explainability of the methods. Nevertheless, the obtained results show that DRL can be a potential solution for improving safety and providing novel insights into safe operations through analysis of the trained policies.

VII. Conclusion

This paper tried to enhance the explainability of Deep Reinforcement Learning (DRL) methods for the task of conflict resolution. It was found that the methods actively changed the trajectory, even in non-conflicting states, which is different from conventional analytical conflict resolution algorithms such as the Modified Voltage Potential (MVP) Algorithm. Moreover, a few of the DRL methods obtained fewer secondary conflicts than their respective MVP counterparts, indicating that the learned policy effectively learned to decrease the domino effect commonly observed in conflict resolution methods. Although not all trained models obtained satisfactory behaviour, the low number of secondary conflicts observed in two of the models shows that it is possible to reduce the number of intrusions while minimizing the domino effect. Because this domino effect is linked to airspace stability, future studies should investigate how and if analytical methods can be enhanced with pre-emptive acting as observed in the policies of the DRL models to ensure that safety at higher traffic densities is maintained.

References

[1] M. Doole, J. Ellerbroek, and J. Hoekstra, "Drone delivery: Urban airspace traffic density estimation," *8th SESAR Innovation Days*, 2018.

[2] "Organization, i.c.a. icao circular 328 - unmanned aircraft systems (UAS). technical report, icao, 2011."

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

[4] E. Sunil, J. Ellerbroek, and J. M. Hoekstra, "Camda: Capacity assessment method for decentralized air traffic control," in *Proceedings of the 2018 International Conference on Air Transportation (ICRAT), Barcelona, Spain*, 2018, pp. 26–29.

[5] Z. Wang, W. Pan, H. Li, X. Wang, and Q. Zuo, "Review of deep reinforcement learning approaches for conflict resolution in air traffic control," *Aerospace*, vol. 9, no. 6, p. 294, 2022.

[6] J. Groot, M. Ribeiro, J. Ellerbroek, and J. Hoekstra, "Improving safety of vertical manoeuvres in a layered airspace with deep reinforcement learning," in *Proceedings of the 10th International Conference for Research in Air Transportation (ICRAT), Tampa, FL, USA*, 2022, pp. 19–23.

[7] 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 *Proceedings of the 11th USA/Europe Air Traffic Management Research and Development Seminar, Lisbon, 23-26 June, 2015*. FAA/Eurocontrol, 2015.

[8] J. M. Hoekstra, J. Ellerbroek, E. Sunil, and J. Maas, "Geovectoring: reducing traffic complexity to increase the capacity of UAV airspace," in *International conference for research in air transportation (ICRAT), Barcelona, Spain*, 2018.

[9] E. Sunil, J. Ellerbroek, J. M. Hoekstra, and J. Maas, "Three-dimensional conflict count models for unstructured and layered airspace designs," *Transportation Research Part C: Emerging Technologies*, vol. 95, pp. 295–319, 2018.

[10] M. Tra, E. Sunil, J. Ellerbroek, and J. Hoekstra, "Modeling the intrinsic safety of unstructured and layered airspace designs," in *ATM R&D Seminar*, 2017.

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

[12] J. M. Hoekstra and J. Ellerbroek, "Bluesky atc simulator project: an open data and open source approach," in *Proceedings of the 7th International Conference on Research in Air Transportation*, vol. 131. FAA/Eurocontrol USA/Europe, 2016, p. 132.

[13] R. Bellman, "A markovian decision process," *Journal of mathematics and mechanics*, vol. 6, no. 5, pp. 679–684, 1957.

[14] J. Sun, J. Ellerbroek, and J. M. Hoekstra, "Wrap: An open-source kinematic aircraft performance model," *Transportation Research Part C: Emerging Technologies*, vol. 98, pp. 118–138, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X18306089

[15] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[16] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International conference on machine learning*. PMLR, 2018, pp. 1861–1870.

[17] J. M. Hoekstra and J. Ellerbroek, "Aerial robotics: State-based conflict detection and resolution (detect and avoid) in high traffic densities and complexities," *Current Robotics Reports*, vol. 2, no. 3, pp. 297–307, 2021.

[18] M. Ribeiro, J. Ellerbroek, and J. Hoekstra, "Velocity obstacle based conflict avoidance in urban environment with variable speed limit," *Aerospace*, vol. 8, no. 4, p. 93, 2021.

[19] C.-J. Hoel, K. Wolff, and L. Laine, "Automated speed and lane change decision making using deep reinforcement learning," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 2148–2155.

[20] K. Bilimoria, K. Sheth, H. Lee, and S. Grabbe, "Performance evaluation of airborne separation assurance for free flight," in *18th Applied Aerodynamics Conference*, 2000, p. 4269.

[21] M. Ribeiro, J. Ellerbroek, and J. Hoekstra, "Determining optimal conflict avoidance manoeuvres at high densities with reinforcement learning," *10th SESAR Innovation Days*, 2020.