Analysis of the impact of traffic density on training of reinforcement learning based conflict resolution methods for drones

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

DOI
10.1016/j.engappai.2024.108066

Publication date
2024

Document Version
Final published version

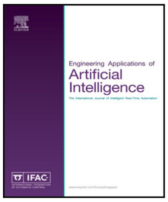
Published in
Engineering Applications of Artificial Intelligence

Citation (APA)
Groot, D. J., Ellerbroek, J., & Hoekstra, J. M. (2024). Analysis of the impact of traffic density on training of reinforcement learning based conflict resolution methods for drones. *Engineering Applications of Artificial Intelligence*, *133*, Article 108066. https://doi.org/10.1016/j.engappai.2024.108066

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai



Research paper

Analysis of the impact of traffic density on training of reinforcement learning based conflict resolution methods for drones

D.J. Groot [*], J. Ellerbroek, J.M. Hoekstra

Delft University of Technology, Kluyverweg 1, 2629 HS, Delft, Zuid-Holland, The Netherlands

ARTICLE INFO

Keywords:
Deep reinforcement learning
Soft actor critic
Conflict resolution
High-traffic density
Unmanned traffic management
Simulation
BlueSky simulator

ABSTRACT

Conventional Air Traffic Control is still predominantly being done by human Air Traffic Controllers, however, as the traffic density increases, the workload of the controllers increases as well. Especially for the area of unmanned aviation, driven by the rise in drones, having human controllers might become unfeasible. One of the methods that is currently being investigated for replacing the conflict resolution task of Air Traffic Control is Reinforcement Learning. As violation of the required separation margins, also called an intrusion, is an event of relatively low frequency, using Reinforcement Learning for this task comes with difficulties that can potentially be attributed to data imbalance. This paper artificially increased the traffic density during the training phase of the Reinforcement Learning method to investigate what the importance is of a balanced data set on the performance of the Reinforcement Learning method. It was found that as the traffic density increased, the Reinforcement Learning methods started to outperform the analytical methods. Beyond this it was found that methods trained at higher traffic densities, but tested at lower traffic densities, outperformed the methods trained at that specific density. This indicates that it might be better to always ensure that the training scenarios are more complex than anticipated during the execution phase, even if that results in unrealistic scenarios.

1. Introduction

The current Air Traffic Control (ATC) system makes use of human controllers who maintain an overview of the current situation, and provide commands to the different aircraft in the airspace if necessary. As traffic densities increase, the task of ATC become more complex, increasing the workload of Air Traffic Controllers (ATCos). It is likely that there exists a threshold traffic density at which human ATCos are no longer able to safely manage the air traffic, at which point either a maximum capacity for the airspace has to be set, or increasingly high levels of automation have to be introduced. Currently, research is already being done on how to alleviate the workload of the ATCOs, for example through conflict detection (CD) and conflict resolution (CR) advisory systems (Ribeiro et al., 2020). However, for unmanned aviation (e.g. drones) in urban environments, predicted traffic densities far exceed the current standards (Doole et al., 2018). It is likely that for these environments ATC has to be fully autonomous with a certain degree of decentralized CR implemented within the autopilots of the drones. The majority of existing CR methods take a geometrical approach to determine the resolution advisory, or to determine the complete set of velocities that is conflict-free (Ribeiro et al., 2020). A downside of these methods is that at higher traffic densities, as conflict geometries become more complex, the space of valid solutions becomes saturated and the effectiveness of these methods decreases.

An alternative to these analytical CR approaches is Deep Reinforcement Learning (DRL). DRL is an area of Machine Learning that tries to learn an optimal policy for sequential decision-making (linking desired actions to specific states or observations of the environment), in order to maximize the cumulative reward that is provided by a reward function (Sutton and Barto, 2018). As DRL methods always output an action to maximize the return of the entire trajectory, it can learn to minimize the impact of the more complex conflicts that occur at higher traffic densities where successful CR manoeuvres might require a sequence of actions and cannot be resolved in a single action. DRL might therefore suffer less at higher traffic densities than analytical methods that only look one action ahead. Using DRL for the task of CR in ATC is not new, and many different approaches have already been suggested, as is visible in the survey by Wang et al. (2022).

Most research that has been conducted on the topic of CR with RL either tried to improve the efficacy of the method by testing different RL algorithms and model architectures or by changing the underlying Markov Decision Process (MDP), changing the state, action and or reward formulations (Wang et al., 2022). However, it is still unclear how the generated scenarios and their associated complexity influence

[*] Corresponding author.
E-mail address: d.j.groot@tudelft.nl (D.J. Groot).

https://doi.org/10.1016/j.engappai.2024.108066
Received 31 March 2023; Received in revised form 22 December 2023; Accepted 6 February 2024
Available online 17 February 2024

the performance and training of the models. Because RL is a data-driven method, the information richness of the data used is very important, it is possible that performance improvements can be obtained by enhancing the quality of the data available for the training.

Conflicts and intrusions are events of low frequency with relatively high impact and high variability in geometry, as was shown in a previous study on conflicts and intrusions in unmanned aviation (Sunil et al., 2015). Because of the low occurrence rate of these events under standard operations, most state-transitions will not contain information relevant for learning how to decrease the number of intrusions. This essentially turns the task of CR with RL into a learning task with imbalanced data, which is known to skew the bias of the learning algorithm towards the majority group in the data (Krawczyk, 2016). This paper therefore aims to provide more insight into the importance of the training scenario complexity/difficulty and how it relates to the obtained performance of the final trained model. This can then hopefully be used as an additional tool for training high-quality models alongside reward engineering, hyperparameter optimization and algorithm selection among others.

In this paper it is investigated whether the efficacy of RL methods applied to the task of CR can be increased by increasing the traffic densities used in the training scenarios. Two hypotheses are established based on the aforementioned information, both of which are tested through a set of simulation experiments. The first hypothesis is that RL methods scale better than analytical CR methods when the traffic density increases, due to the increasingly high complexity of conflict geometries. Creating conflict cases that cannot be effectively encompassed within the analytical methods. The second hypothesis is that training at higher traffic densities reduces the data imbalance present in the data-set. This causes methods trained at high traffic densities to outperform methods trained at lower traffic densities, when tested in the same scenario.

These hypotheses are tested using a set of simulated traffic scenarios, using the BlueSky Open Air Traffic Simulator (Hoekstra and Ellerbroek, 2016).[1] In these traffic scenarios, the DRL method is tasked with resolving the conflicts that occur during vertical manoeuvres in a layered urban airspace (Sunil et al., 2015). Previous work has already shown that vertical manoeuvres in these types of airspace structures contribute to a large set of conflicts and intrusions, which potentially can be mitigated using DRL (Sunil et al., 2018; Groot et al., 2022). Additionally, these vertical conflicts are easily isolated from the entire set of conflicts. This allows horizontal conflicts to be ignored for the purpose of this research, lowering computational requirements. Because of this, much higher traffic densities can be simulated as all cruising aircraft follow a straight flight path and only the vertically manoeuvring aircraft have to be controlled by the DRL method.

The remainder of this paper is structured as follows. First the definitions of conflicts and intrusions are explained in more detail in Section 2. Section 3 introduces the experimental setup, scenario and variables of interest. The used methods are given in Section 4 and the results and discussion are shown in Section 5. Finally Section 6 concludes the paper and provides recommendations for future studies.

2. Definitions

2.1. Intrusions

Intrusions, also called Losses of Separation, are events that occur when two or more aircraft converge beyond some predefined separation minima in both the horizontal and vertical plane. These intrusions are considered serious safety breaches and can lead to collisions due to position/trajectory uncertainties. The number of intrusions should therefore always be kept at a minimum. Fig. 1 shows an example of such an intrusion, in this figure pz_h and pz_v are the horizontal and vertical separation minima respectively. Note the ‘no intrusion’ aircraft being labelled as such because of its vertical offset from the ‘ownship’ aircraft.

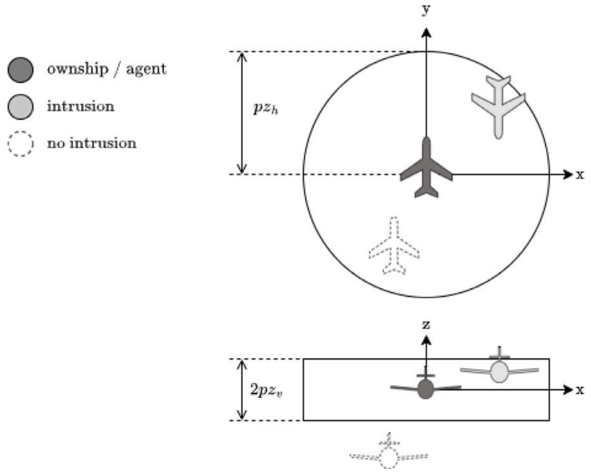


Fig. 1. Figurative illustration of an intrusion vs no intrusion.

2.2. Conflicts

A conflict is defined as a predicted intrusion. This prediction is either done based on extrapolation of the current aircraft state (state-based conflict detection) or by using the intentions of the different aircraft (intent-based conflict detection). In this research, state-based conflict detection is used, as illustrated in Fig. 2. In this figure, CPA is the so-called closest point of approach in the horizontal plane, d_{cpa} is the distance at CPA. The time until CPA, t_{cpa}, can be calculated using Eq. (1), here Δx, Δy and Δu, Δv are the relative positions and velocities in x and y direction respectively. This can then be used to calculate the d_{cpa} using Eq. (2). The time in and out of horizontal conflict follows from Eq. (3).

$$t_{cpa} = \frac{\Delta x \Delta u + \Delta y \Delta v}{(\Delta u^2 + \Delta v^2)} \tag{1}$$

$$d_{cpa} = \sqrt{(\Delta x - \Delta u \cdot t_{cpa}) + (\Delta y - \Delta v \cdot t_{cpa})} \tag{2}$$

$$t_{inhor}, t_{outhor} = t_{cpa} \pm \frac{\sqrt{pz_h^2 - d_{cpa}^2}}{v_{rel}} \tag{3}$$

To include vertical conflict detection, the time-window of vertical separation breach is calculated with Eq. (4), where Δh and Δv_z are the relative position and relative speed in the vertical plane respectively. In the case of $\Delta v_z = 0$ a very small number is used to avoid division by zero. These results are combined with the results from Eq. (3) to see if there is an overlap in horizontal and vertical conflict time-window, Eq. (5).

$$t_{inver}, t_{outver} = \frac{\Delta h \pm pz_v}{\Delta v_z} \tag{4}$$

$$\text{conflict} = \begin{cases} 1 & \max(t_{inhor}, t_{inver}) < \min(t_{outhor}, t_{outver}) \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

[1] As this simulator uses primarily the same units as used in the aviation industry, this paper will use the units used in the simulator for better reproducibility.

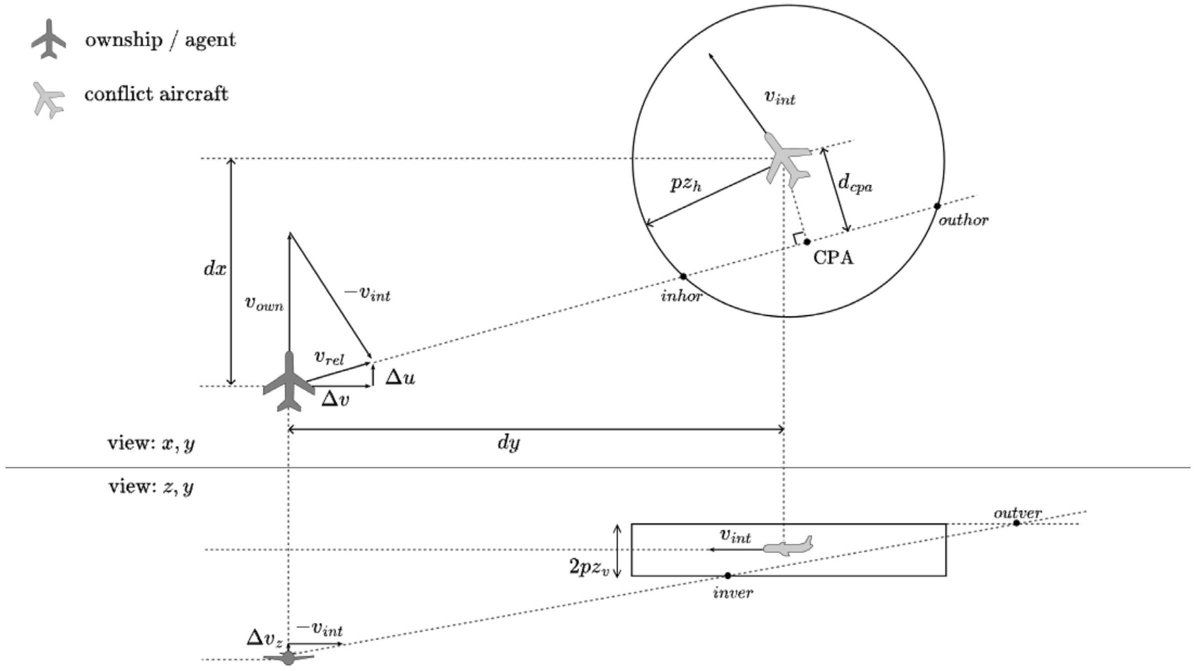


Fig. 2. Illustration of the horizontal and vertical component of a conflict. During a conflict the ownship can either change its heading and speed to ensure that the CPA is located outside of the area of minimum separation or change its vertical velocity to ensure that the horizontal and vertical components of the conflict do not overlap.

3. Experiments

In this research two different sets of experiments are conducted. The first experiment compares the effectiveness of an analytical conflict resolution method based on the Modified Voltage Potential algorithm with a method using Deep Reinforcement Learning over a variety of different traffic densities. The second experiment trains three different DRL methods at increasingly high traffic density and tests these same methods under different traffic densities than they were trained at.

For both of these experiments the experimental scenario and control variables are identical. In this section the simulation environment, scenario, different variables and the experimental hypotheses will be described.

3.1. Experimental setup

3.1.1. Airspace layout

The airspace layout is based on the layered airspace concept from the Metropolis project (Sunil et al., 2015). In this layered airspace aircraft are grouped by heading into different, vertically separated, layers in the airspace. This decreases the relative velocities between the different aircraft, but also requires vertical manoeuvres when changing heading, or traversing multiple layers of cruising aircraft when taking off and landing.

For this research a total of 16 vertically separated layers are used, each with an allowed heading range of 45° degrees, which means that the entire heading range from 0 – 360° degrees is covered twice. The top set of layers is occupied by long-distance flights and the bottom set of layers for short distance commute. A graphical representation of the airspace structure is given in Fig. 3. Each layer is set at 50ft in height.

The simulation area is a disc of radius R_{outer} = 1.62 NM (3 km) and height $h_{airspace}$ = 800 ft (243.8 m). Within this simulation area a smaller disc with radius R_{inner} = 1.35 NM (2.5 km) is used for the experimental area. Dependent measures are only recorded within this experimental area. The entire airspace is considered to be free of static obstacles by assuming that all aircraft are flying above buildings, which means that they are not obliged to follow pre-existing road networks.

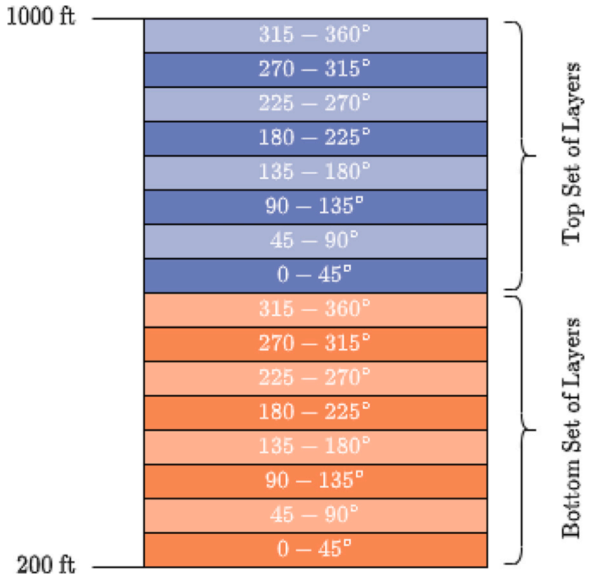


Fig. 3. Illustration of the layers used in the layered airspace. In each layer is indicated the allowed heading range of that layer.

3.1.2. Traffic generation

To ensure that the traffic is uniform-randomly distributed over the entire available airspace at the appropriate traffic density, scenarios are not scripted, but rather traffic is generated on-line. First the average duration of a flight in the airspace, t_{flight}, is calculated using Eq. (6), where v_{hor} is the mean horizontal velocity in m/s. This average flight time is then converted to an interval at which different aircraft should be created at the border of the environment using Eq. (7). In this equation 'density' is the required traffic density in AC/NM^2.

$$\bar{t}_{flight} = \frac{\pi \cdot R_{outer} \cdot 1852}{2\bar{v}_{hor}} \tag{6}$$

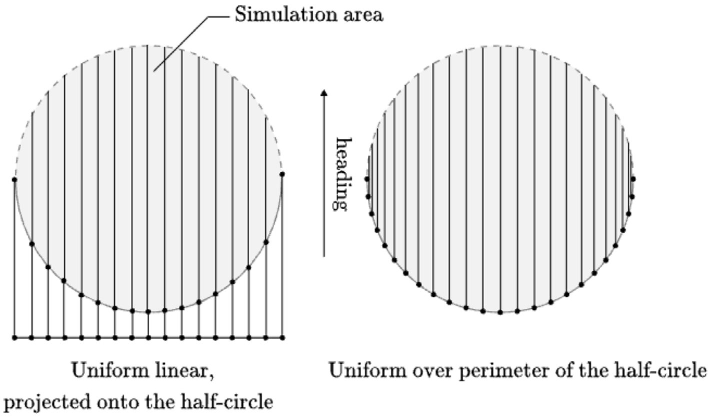

Simulation area
heading
Uniform linear,
projected onto the half-circle
Uniform over perimeter of the half-circle

Fig. 4. Different ways of determining entry bearing. Projected from a linear distribution (left), equal distribution over the perimeter of the half-circle (right).

$$I_{spawn} = \frac{\bar{t}_{flight}}{\text{density} \cdot \pi \cdot R_{outer}^2} \tag{7}$$

The spawn location of an aircraft on the border of the environment is based on the projection of a line orthogonal to the aircraft heading to the border of the environment. This ensures that the traffic is homogeneously distributed over the airspace (Fig. 4 left), instead of having a higher traffic density closer to the borders that would occur when an equal distribution over the perimeter of the half-circle would be used (Fig. 4 right). The calculation of entry bearing with respect to the centre of the environment is given in Eq. (8), where x is a random value between −1 and 1 corresponding with a point on the linear line projection of the half-circle, and heading is a randomly selected value between 0 and 360.

$$\text{entry bearing} = \text{heading} + 180 + \text{asin}(x); x \in \mathcal{R}[-1, 1] \tag{8}$$

3.1.3. Experimental scenario

In all experiments the goal of the aircraft is to safely conduct vertical manoeuvres in the layered airspace, traversing through the different cruising layers while avoiding intrusions with cruising aircraft. Aircraft entering the experimental area defined by R_{inner}, have a probability of 0.05 to be assigned to an altitude layer in the other layer set. This probability ensures that most vertical conflicts are with cruising aircraft, while still having enough simulated aircraft that generate relevant data for the research. During the vertical manoeuvres, the aircraft conducting the vertical manoeuvres are responsible for maintaining safe separation margins with the cruising aircraft by resolving the conflicts that occur. When aircraft reach their new target altitude layer they will be considered cruising aircraft for the remainder of the simulation.

Due to the setup of the experimental scenario, the resulting traffic density and distribution of vertically manoeuvring aircraft is relatively homogeneous. In reality, it is expected that hotspots of higher traffic densities will occur, something that is not captured in the scope of this research. Future research should investigate how and if the results of this research change if various traffic densities are present in a single environment.

3.2. Control variables

3.2.1. Aircraft performance model

All aircraft in the simulation use the same point mass performance model which is based on the Mavic DJI pro. The specifications of this model are obtained from the manufacturers website (Mavic, 2022). An exception is made for the downwards vertical velocity, which is set at −5 m/s to ensure symmetry with the reported positive vertical velocity of 5 m/s. This symmetry allows the same policy to be used for both the climbing and descending tasks, effectively doubling the training efficacy.

Table 1
Traffic densities used for the different experiments and the associated number of aircraft.

Density	Num. Instantaneous aircraft	Num. Vertically manoeuvring aircraft
25	206	10
50	412	20
100	824	41
150	1237	62
200	1649	82
300	2473	124

3.2.2. Separation minima

The separation requirements for all aircraft are 50 m of horizontal and 1 layer of vertical separation. The value for vertical separation is set such that aircraft cruising in one layer can only have an intrusion with other aircraft currently in that layer.

3.2.3. Cruising aircraft characteristics

In the experiments cruising aircraft will behave as dynamic obstacles flying at a speed of 10 m/s and a constant heading based on their initial conditions. All cruising aircraft will continue their trajectory without conflict resolution, which ensures that the observed performance is solely attributed to the actions taken by the vertically manoeuvring aircraft.

3.3. Independent variables

3.3.1. Traffic density

The main variable that is changed between the different experiments is traffic density, given in AC/NM^2. Table 1 shows the different traffic densities at which the first experiment is run in combination with the average number of instantaneous aircraft and concurrent vertical manoeuvres in the airspace. For the second experiments, an additional model trained at a traffic density of 2000AC/NM^2 is evaluated for the same traffic densities as used in experiment 1.

3.3.2. Aircraft control method

Each traffic density scenario is simulated with 3 different control methods for the vertically manoeuvring aircraft, 'No conflict resolution', 'SWO' and 'DRL'.

- **No conflict resolution** will fly the aircraft directly to the target altitude at a constant vertical velocity of 2.5 m/s and horizontal velocity of 10 m/s, without avoiding any of the aircraft.
- **SWO** uses a shortest way out algorithm to resolve conflicts encountered during the vertical manoeuvre. 'SWO' will by default follow the same flightpath as 'No conflict resolution' and only

change the trajectory when a conflict is detected. After the conflict is resolved the changed trajectory is maintained until $t_{cpa} < 0$, to ensure that the aircraft does not steer back into conflict. An exception to this is when a new conflict is detected, in that case the trajectory is changed to resolve the new conflict.

- **DRL** uses a DRL method to control the aircraft trajectory at each timestep. This is different from 'SWO' which is only activated around conflicts. Because of this the model is also able to manoeuvre to more favourable states when there is no conflict. For the first set of experiments the DRL method is trained and tested at the same traffic densities. For the second set of experiments the DRL method will be trained at one specific traffic density, indicated in the figures, and then tested at other traffic densities to observe how the method scales to different traffic densities.

3.4. Dependent measures

3.4.1. Intrusion rate

The main variable of interest is the intrusion rate, defined as the average number of intrusions per vertical manoeuvre. As the main goal of the aircraft is to minimize the number of intrusions, a lower value indicates a better performance.

3.4.2. Conflict rate

The conflict rate is similar to the intrusion rate, but looks at the average number of conflicts that occurred during vertical manoeuvres. An increase in the number of conflicts is expected for the methods that actively resolve the conflicts, as the resolution manoeuvres may result in secondary conflicts. If the conflict rate becomes too large compared to 'no conflict resolution' this might indicate destabilization of the traffic flow (Bilimoria et al., 2000).

3.4.3. Intrusion severity

The intrusion severity is a measure used to evaluate how bad the intrusion that occurred was. The intrusion severity is calculated using Eq. (9). From this equation follows that the highest possible intrusion severity of 1 corresponds to an intrusion that resulted in a collision and intrusion severity of 0 is given when no intrusion occurs.

$$\text{severity} = \begin{cases} 1 - dh/pz_h & \text{intrusion} \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

3.4.4. Return

The return is the total sum of rewards observed during the vertical manoeuvre. The value is used to assess the training of the DRL methods by observing the evolution of the return over time. The return is also used to evaluate the performance of the different methods on the actual task as it is defined mathematically.

3.4.5. Duration of vertical manoeuvre

The duration of the vertical manoeuvre is a different way to express the average vertical velocity. A lower duration of vertical manoeuvre indicates that the aircraft used smaller or fewer vertical velocity changes during conflict resolution.

3.4.6. Horizontal distance

The horizontally travelled distance is similar to the time of vertical manoeuvre but gives an indication for the usage of horizontal velocity changes for conflict resolution.

3.5. Experimental hypotheses

The hypothesis that is tested with the first set of experiments, which compares how analytical methods and DRL methods scale with more complex scenarios by increasing the traffic density, is: Deep Reinforcement Learning methods scale better than analytical methods when the traffic density increases due to the increasingly high complexity of the environment, creating scenarios that cannot be effectively encompassed within the analytical methods.

For the second set of experiments, evaluating the performance of DRL methods trained at different densities than they are tested in, the hypothesis is: Training at higher traffic densities results in information richer state transitions creating more relevant data to use for training, this in return causes models trained under high traffic densities to outperform models trained for the same scenario but at lower traffic densities.

4. Materials and methods

4.1. Simulator: BlueSky

All of the experiments are simulated using the BlueSky open-source Air Traffic Simulator (Hoekstra and Ellerbroek, 2016). BlueSky is a fast-time air traffic simulator that allows easy implementation of custom scenarios and plugins whilst being fully open-source, which aids in the repeatability of the conducted research. The most recent version of BlueSky and documentation in the form of a wiki, can be found at TUDelft-CNS-ATM (2022), and the version including the plugins used for this research is provided at Groot (2022).

4.2. Shortest way out conflict resolution

The baseline conflict resolution model used for this research is a shortest way out method based on the Modified Voltage Potential (MVP) Algorithm (Hoekstra et al., 2002). If a conflict is detected, the MVP algorithm determines the smallest required change of the velocity vector to escape the velocity obstacle from the aircraft's own perspective. Both conflicting aircraft determine their own shortest way out of conflict making the method implicitly coordinated in the case that both aircraft actively resolve the conflict. Fig. 5 shows a graphical representation of the change in horizontal speed and heading due to the MVP algorithm. Determining the new velocity vector is done using Eqs. (10) and (11).

For the change in vertical speed, the values for Δv_z in Eq. (4) are calculated, such that the resulting t_{inver} and t_{outver} do not lead to a conflict according to Eq. (5). The smallest required vertical speed change is used for the change in vertical speed. For this research, only the aircraft that are conducting vertical manoeuvres will be using this conflict resolution method, which does change the method compared to MVP. This makes the method consistent with the DRL method, which also only controls the vertically manoeuvring aircraft through speed, heading and vertical speed commands.

$$V_{avoid} = \frac{\vec{CT}}{t_{cpa}} \tag{10}$$

$$V_{new} = V_{old} + V_{avoid} \tag{11}$$

4.3. Deep reinforcement learning

Reinforcement Learning (RL) is a machine learning used for learning optimal policies in environments modelled as Markov Decision Processes. Through interactions with this environment the RL method continuously updates the policy, which specifies actions to take, given an observation of the environment, to maximize the cumulative sum of rewards (the return R). Traditional reinforcement learning methods

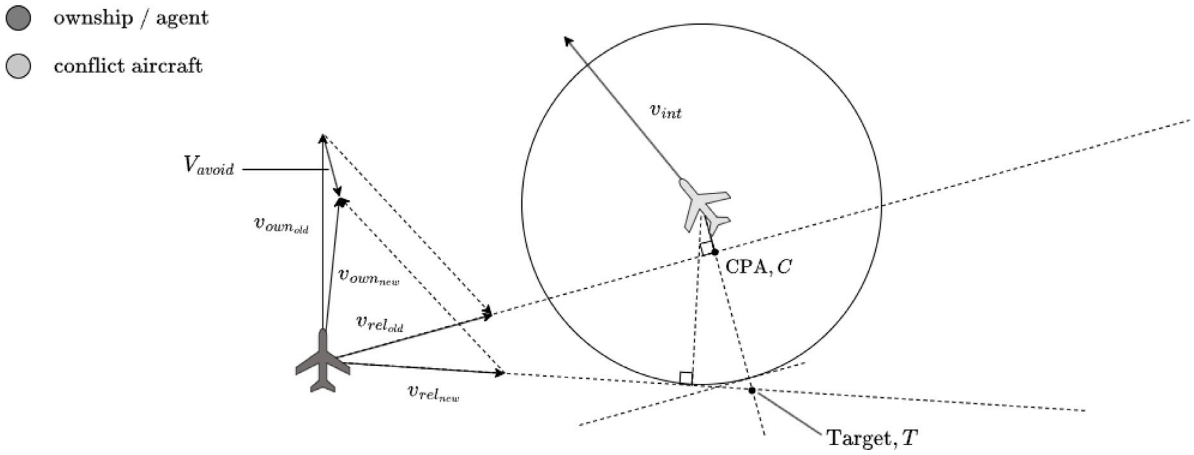


Fig. 5. Construction of the horizontal speed and heading change for MVP in a single conflict scenario.

use tables to store all the possible state action pairs and their value (estimated return) which limits the scalability to large state and action spaces and requires discretization of all continuous variables (Sutton and Barto, 2018). Deep Reinforcement Learning (DRL) solves this problem by replacing the tables with function approximators in the form of Deep Neural Networks (DNNs). This allows the field of Reinforcement Learning to be applied to larger state spaces with continuous actions or to, for example, learn from raw pixel inputs (Mnih et al., 2015) and generalize the learned policy to never before encountered states.

4.3.1. Soft Actor Critic

For this research the Soft Actor Critic (SAC) algorithm is used as it allows usage of continuous actions, is robust and implicitly incorporates exploration of the available solution space by including an augmentation of the reward with an entropy term. This rewards higher randomness in the policy to aid with the exploration of more state action pairs (Haarnoja et al., 2018).

SAC is an Actor–Critic method. Actor–Critic methods consist of 2 elements, the actor, and the critic. The Actor (denoted by π) takes as input the current state of the environment through an observation vector, and returns the action to be taken by the agent, Eq. (12). The Critic (denoted by Q) takes as an input the observation vector and the selected action and provides an estimate of the (discounted) return, Eq. (13), here $\gamma \leq 1$ is the discount factor, r is the reward at timestep n and T is the termination timestep. SAC includes also a Value function (V), which takes as an input only the observation vector, and outputs another estimate of the (discounted) return, Eq. (14). This allows the Value function to encompass the randomness of the Actor policy (present due to the reward for higher entropy) which gets lost in the Critic function (Q takes a deterministic action as input). More details of the SAC algorithm including the algorithm in pseudo-code are given in the original paper by Haarnoja et al. (2018). A full list of the hyperparameters used for this research is given in Table 2 and are based on the hyperparameters used by Haarnoja et al., as they show that these hyperparameters result in stable behaviour over a wide range of tasks.

$$a_t = \pi(s_t) \tag{12}$$

$$Q(s_t, a_t) = \mathbb{E}[\sum_{n=t}^{T} \gamma^{n-t} r_n \mid s_t, a_t] \tag{13}$$

$$V(s_t) = \mathbb{E}[\sum_{n=t}^{T} \gamma^{n-t} r_n \mid s_t] \tag{14}$$

Table 2
Hyperparameters for the soft actor critic algorithm.

Parameter	Value
Optimizer	Adam
Learning rate	3e−4
Discount factor (γ)	0.995
Memory buffer size	10e6
Sample size	256
Smoothing coefficient (τ)	5e−3
Number of layers	2
Neurons per layer	256
Network update frequency	1

4.3.2. Observation vector

The implementation of SAC for this research requires the input vector to be consistent in size. The state of the environment, which is variable in size with the number of aircraft, is therefore approximated by an observation vector of constant length, which will be used as the input for both the policy/Actor and the Critic. The observation vector consists of 2 different elements, the features related to the states of the agent and the features related to potential intruders surrounding the agent. The agent's state features that are included in the observation vector are given in 'ownship' section of Table 3. For the intruder features three aircraft are selected from the environment based on t_{cpa}, d_{cpa} and relative position. The intruder selection is shown in algorithm 1, with $d_{lookahead}$ = 0.3 NM (555.6 m), and ensures that the elements in the observation vector are sorted based on urgency, which relates to the conflict boolean, t_{cpa} and spatial relationship. For each of these aircraft the features in the 'intruder' section of Table 3 are included in the observation vector. Finally all entries in the observation vector are normalized to have a mean of 0 and variance of 1 under a fully random actor to stabilize the initial training phase.

Algorithm 1 Find neighbouring aircraft for observation vector

```
Ensure: preselection list → empty
  for $int$ in list of aircraft do
    if $(int_{alt} - own_{alt})(int_{alt} - target_{alt}) < 0$ and $d_{hor_{int,own}} < d_{lookahead}$ then
      add $int$ to preselection list
      get $t_{cpa}$, conflict (boolean), intrusion (boolean)
    end if
  end for
  sort preselection list on intrusion, conflict and $t_{cpa}[t_{cpa} > 0]$ in ascending order
  return first 3 elements of preselection list
```

Table 3
Features included in the observation vector.

	Variable	Description
ownship	Δalt	Altitude difference with target altitude
	v_z	Vertical speed
	v_h	Horizontal speed
	Δhdg	Heading difference with the nominal heading of the current layer
intruder	Conflict	Boolean conflict variable
	t_{cpa}	Time till closest point of approach
	d_{cpa}	Distance at closest point of approach
	Δh	Height difference with the intruder
	Δu	Speed difference in the x direction
	Δv	Speed difference in the y direction
	dx	x position of the intruder
	dy	y position of the intruder
	dh	Horizontal distance of the intruder

Table 4
Action space limits.

	Δ per timestep	Bounds
v_z (m/s)	[−2,86, +2.86]	[0, 5]
v_h (m/s)	[−2.5, +2.5]	[5, 15]
Heading (°)	[−45, 45]	[0, 360]

4.3.3. Action space

The control methods control the aircraft by selecting the desired vertical speed, change in horizontal speed and change in heading as a float value. The control module is set to always use the maximum acceleration and angular velocity as specified by the aircraft performance model to attain these desired changes. Because of this the action space is bounded by the aircraft performance characteristics. An additional constraint is used for the vertical velocity, which can only be between 0 and 5 m/s, where the positive direction is defined in the direction of the target altitude. This ensures that even a random actor will eventually reach the target state, further enhancing the stability of the DRL method. The bounds and maximum change per timestep of the different actions are given in Table 4. The Actor has 3 output neurons with a sigmoid activation layer, one for each available action. This results in outputs in the range of [0,1], which are afterwards mapped to the corresponding ranges shown in Table 4.

4.3.4. Reward function

To minimize the introduction of human biases in the learned policy of the DRL method, the reward function is kept as simple as possible. Only penalties are given for intrusions and the reward is zero otherwise. The reward for each timestep is given in Eq. (15). This reward function ensures that the only incentive of the DRL method is to learn how to minimize the number of timesteps containing intrusion states.

$$r_t = -0.25 \cdot N_{intrusions} \tag{15}$$

5. Results and discussion

5.1. Training results

Fig. 6 shows the evolution of the return against the number of conducted vertical manoeuvres for the DRL methods trained at 100, 200, and 300 AC/NM^2. From this figure it is visible that all methods have a steep initial learning curve before levelling off at approximately 20.000 vertical manoeuvres. The lower return for the methods trained at higher traffic densities correlates with the higher expected number of intrusions that occur at these densities.

5.2. Comparing learned methods against a shortest way out method

The results in this section show the outcome of the first experiment, which compares the performance of the ‘No conflict resolution’, ‘SWO’ and ‘DRL’ methods at various traffic densities.

Table 5
Number of intrusions per vertical manoeuvre for the SWO and DRL methods and the relative change in number of intrusions for the DRL method with respect to the SWO method.

Traffic density	Intrusions SWO	Intrusions DRL	Relative difference
25	2.64e−3	4.97e−3	+87.5%
50	2.77e−3	5.17e−3	+86.3%
100	7.73e−3	6.40e−3	−17.3%
150	1.23e−2	1.20e−2	−2.5%
200	1.82e−2	2.04e−2	+12.3%
300	3.50e−2	2.45e−2	−29.9%

5.2.1. Intrusion rate

The intrusion rate for the ‘No conflict resolution’, ‘SWO’, and ‘DRL’ methods are given in Fig. 7. This figure shows that both the ‘SWO’ and ‘DRL’ methods effectively reduce the total number of intrusions when compared with not doing any conflict resolution, which is expected. Table 5 shows the mean intrusion rate for the ‘SWO’ and ‘DRL’ methods and their relative difference. From this table, it can be observed that the performance of the ‘DRL’ method improves with respect to the ‘SWO’ method as the traffic density increases.

This observation is in line with the hypothesis that the ‘DRL’ method scales better with increasing traffic densities than analytical methods, and can be attributed to two effects that occur at higher traffic densities. First, the higher traffic density ensures that there are more conflict and intrusion states in the memory buffer, which makes it more likely that these states are used for policy updates, resulting in better sample efficiency. Secondly, the higher traffic density leads to more complex conflict geometries as the solution space shrinks with more aircraft occupying the available space. It is more difficult to encapsulate all of these complex conflicts effectively in an analytical method such as a shortest way out algorithm. A data-driven method such as DRL on the other hand can learn how to resolve these conflicts the more it is exposed to them, a feature that is not present in analytical methods.

5.2.2. Conflict rate

As expected, the conflict rate increases when conflict resolution is performed, as is shown in Fig. 8. What is interesting however is the relatively higher conflict rate of the ‘DRL’ method when compared with the ‘SWO’ method. Because the intrusion rate of the methods is relatively similar, especially at higher traffic densities, it is notable that the resolution strategy of the ‘DRL’ method results in a larger number of secondary conflicts. This can likely be attributed to the fact that the ‘SWO’ method uses the minimum required change in velocity, which exposes the aircraft to a smaller portion of the state space, decreasing the probability of secondary conflicts occurring. The ‘DRL’ method on the other hand has no explicit incentive to use the minimum required velocity change to resolve the conflict, as this is not included in the reward function. Including the efficiency of the operations in the reward function might decrease the number of secondary conflicts. However, this should be balanced accordingly to ensure that safety does not decrease at the cost of better efficiency.

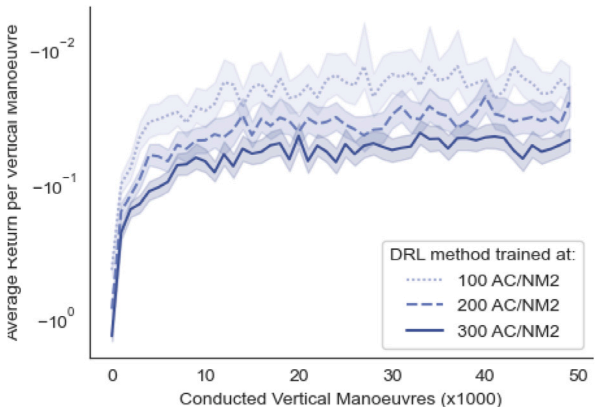


Fig. 6. Evolution of the return during training for the DRL methods trained at 100, 200, and 300 AC/NM^2.

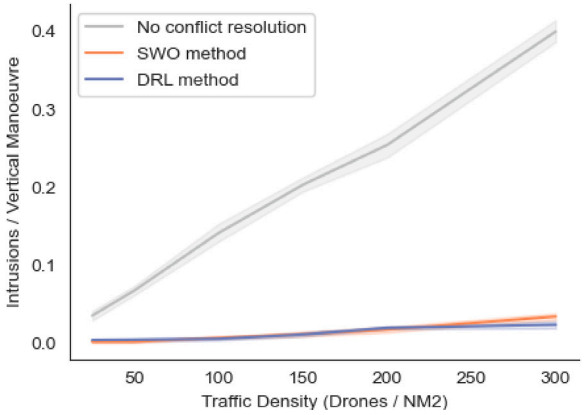


Fig. 7. Number of intrusions per vertical manoeuvre at different traffic densities for the 'No conflict resolution', 'SWO', and 'DRL' methods.

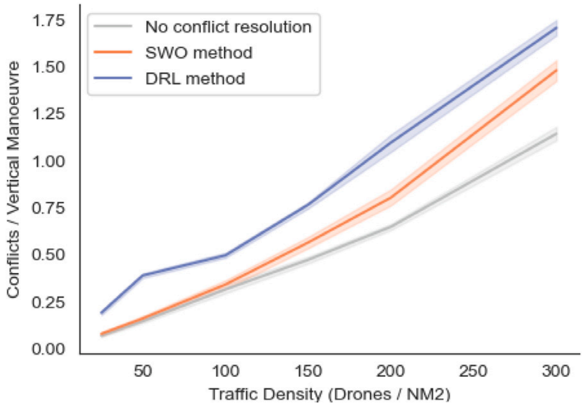


Fig. 8. Number of conflicts per vertical manoeuvre at different traffic densities for the 'No conflict resolution', 'SWO', and 'DRL' methods.

5.2.3. *Intrusion severity*

Using conflict resolution methods during the vertical manoeuvres not only reduces the number of intrusions that occur during the vertical manoeuvres, but also reduces the severity of these intrusions when they occur, as is shown in Fig. 9. It also becomes apparent that there is a small increase in the intrusion severity for both resolution methods when increasing the traffic density. Again, at lower traffic densities the 'SWO' method outperforms the 'DRL' method. At higher traffic densities, however, the lower intrusion severity of the 'SWO' method compared to the 'DRL' method diminishes. This observation further strengthens the first hypothesis that the 'DRL' method scales better with increasing traffic densities than analytical methods.

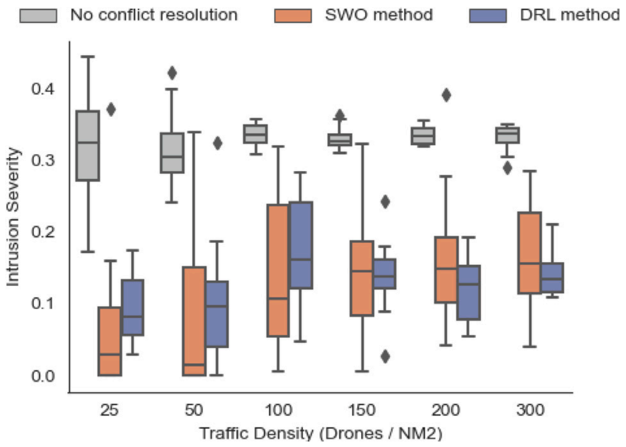


Fig. 9. Box and whisker plot for the severity of the intrusions for the 'No conflict resolution', 'SWO' and 'DRL' methods.

5.3. *Effect of training at higher densities than used for testing*

This section shows the outcome of the second experiment, where these three 'DRL' methods, trained at different traffic densities, are compared under the same conditions as used for the first experiment. The traffic densities used for training are 25, 300, and 2000 AC/NM^2

5.3.1. *Intrusion rate*

The intrusion rate for the different 'DRL' methods is given in Fig. 10. This figure clearly shows that training at lower traffic densities does not scale properly to higher traffic densities. Interestingly, the performance of the methods trained at a traffic density of 300 and 2000 AC/NM^2 are relatively similar for traffic densities up to 150 AC/NM^2, but the lines diverge when going to higher traffic densities, with the method trained a traffic density of 2000 AC/NM^2 having a lower intrusion rate. At higher traffic densities the probability of being in conflict with more than one aircraft at a time increases. If the efficacy for resolving single- and multi-aircraft conflicts would be the same for both methods, the observed trend of both methods having a similar intrusion rate should be continued beyond the 150 AC/NM^2 mark. As the method trained at 2000 AC/NM^2 has a higher exposure rate to these multi-aircraft conflicts during training than the method trained at 300 AC/NM^2, it is concluded that not only the frequency of relevant state transitions (intrusions and near intrusions) is important for the performance of data-driven methods such as DRL, but that also the diversity of samples within this set of relevant state transitions contributes to the overall performance.

To obtain a clearer picture of the impact of training at a much higher traffic density than the one used for testing, Fig. 11 shows the intrusion rate of the 'DRL' method trained at a traffic density of 2000 AC/NM^2 together with the intrusion rates of the 'SWO' and 'DRL' methods also shown in Fig. 7. This figure shows that the model trained at the much higher traffic density has a lower intrusion rate than both the 'SWO' and 'DRL' methods for virtually all traffic densities except for the lower densities, where 'SWO' is relatively equal.

5.3.2. *Return*

To assess how well the different methods perform when compared against the reward function Fig. 12 shows the average return of the methods. The difference between the 'DRL' methods and the 'SWO' method is more clearly visible in this figure, indicating that the models

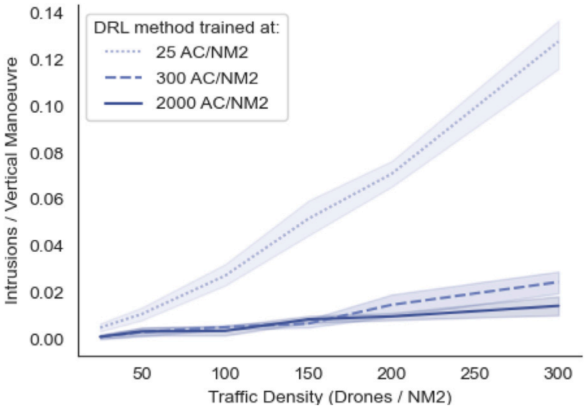


Fig. 10. Number of intrusions per vertical manoeuvre at different traffic densities for the 'DRL' methods trained at a traffic density of 25, 300, and 2000 AC/NM^2.

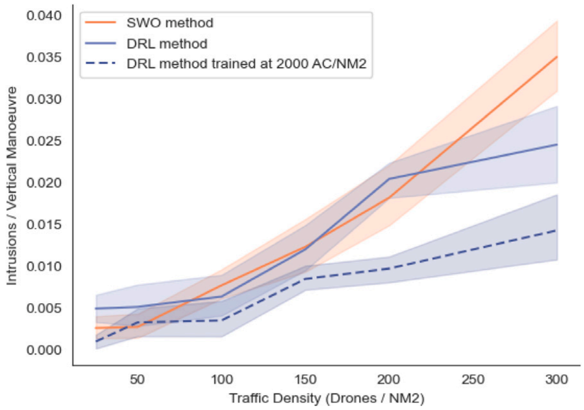


Fig. 11. Comparison of the intrusion rate between the 'SWO', 'DRL', and 'DRL trained at 2000 AC/NM^2' methods.

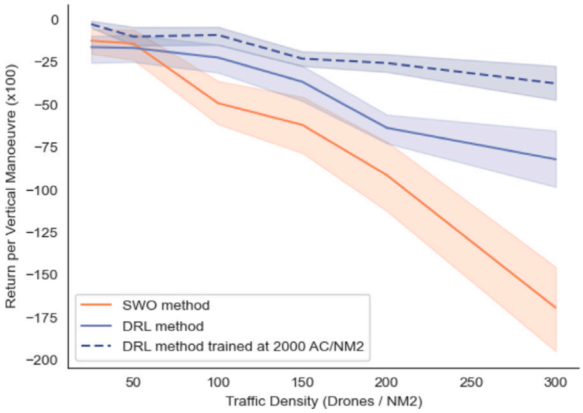


Fig. 12. Return per vertical manoeuvre for the 'SWO', 'DRL' and 'DRL trained at 2000 AC/NM^2' methods.

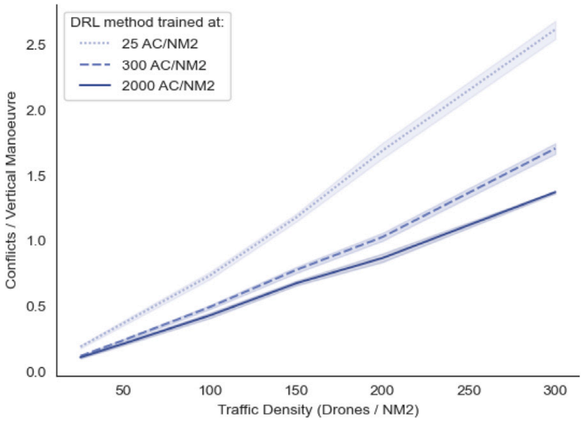


Fig. 13. Number of conflicts per vertical manoeuvre at different traffic densities for the 'DRL' methods trained at a traffic density of 25, 300, and 2000 AC/NM^2.

score better when assessed against the reward function than the intrusion rate. As the reward function is the only type of feedback the methods get during the training phase this difference is expected, it however also shows that the reward function does not fully encompass all necessary information to minimize the intrusion rate. This is because in the reward function different intrusions are not distinguished from each other. Instead, every timestep with an intrusion, regardless of it being a new or old intrusion will be penalized equally. The intrusion rate on the other hand looks at the total number of unique intrusions, instead of the number of intrusion timesteps in the manoeuvre. If minimizing the number of unique intrusions, as is used for the intrusion rate, would be the main objective, altering the reward function such that unique intrusions are only counted once might result in an even higher reduction in the intrusion rate than is currently observed. For this research, however, the main goal is to lower the total time in intrusion instead, which is reflected in Fig. 12.

5.3.3. Conflict rate

The conflict rate for the methods trained at different traffic densities, shown in Fig. 13, shows an interesting trend. Fig. 10 already showed that the intrusion rate decreases when the method trains at higher traffic densities. However, from Fig. 13 it becomes apparent that the conflict rate also decreases when training the method at a higher traffic density, which is not necessarily in line with the common conception that the number of conflicts increases when conflict resolution is used due to secondary conflicts. This trend can be explained as reducing the total number of conflicts is more important at higher traffic densities. At higher traffic densities the probability of secondary conflicts is higher, which when not addressed can lead to instabilities. Therefore the methods trained at higher traffic densities learned that besides resolving the conflicts, minimizing the total number of (secondary) conflicts should also be considered. This behaviour can be considered emergent, as the reward function does not contain any explicit incentive to reduce the number of conflicts.

5.3.4. Intrusion severity

The trend seen in Figs. 10 and 13 also continues when looking at the intrusion severity, shown in Fig. 14. For the higher traffic densities, it can be observed that the methods trained at the higher traffic densities have a consistently lower intrusion severity. Something interesting to observe is the break of the trend that can be observed at lower traffic densities (25 and 50 AC/NM^2). Here the method trained at a traffic density of 2000 AC/NM^2 shows an increase in intrusion severity compared to the higher traffic densities. The same pattern is also observable in Fig. 11 where 50 AC/NM^2 is the cut-off where the 'DRL' method trained at a traffic density of 2000 AC/NM^2 no longer has a lower intrusion rate than the 'SWO' method. This potentially indicates that there is a lower limit to the traffic density in which methods trained at higher traffic densities can be deployed.

5.4. Duration of, and horizontal distance covered during, vertical manoeuvres

To put the performance of the different methods into perspective, this section will analyse the duration of the vertical manoeuvres and the

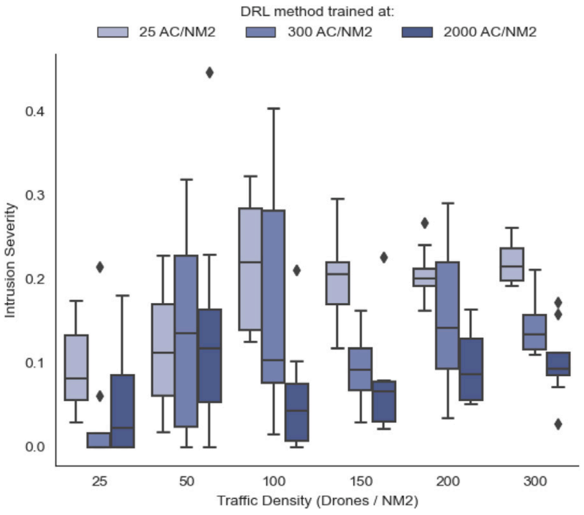


Fig. 14. Box and whisker plot for the severity of the intrusions for the 'DRL' methods trained at a traffic density of 25, 300 and 2000 AC/NM^2.

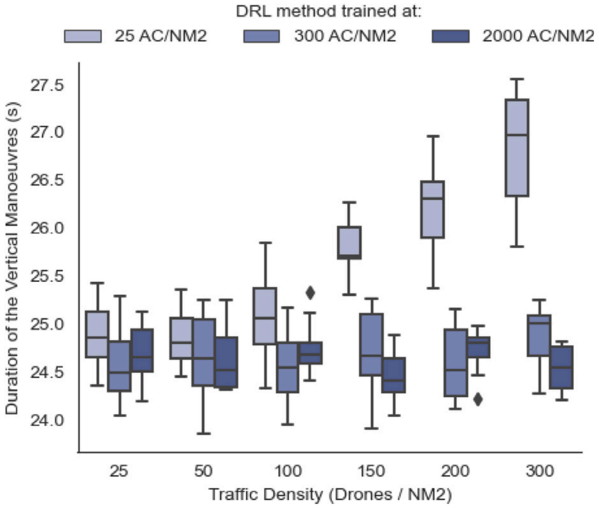


Fig. 15. Duration of the vertical manoeuvres at different traffic densities for the 'DRL' methods trained at a traffic density of 25, 300, and 2000 AC/NM^2.

horizontal distance covered during the manoeuvres. This data is provided in Figs. 15 and 16 respectively. Fig. 15 shows that the duration of the vertical manoeuvres for all of the methods is relatively similar at the lower traffic densities. At higher traffic densities, however, the duration goes up for the method trained at a traffic density of 25 AC/NM^2, indicating a decrease in the mean vertical velocity when the number of aircraft increases. Because this behaviour is not present in the methods trained at higher traffic densities, this strategy likely works at lower traffic densities, where the different cruising layers are relatively empty and loitering is possible to enable safe flights. However, this strategy becomes invalid at higher traffic densities, leading to the higher intrusion rate observed in Fig. 10.

The covered horizontal distance shown in Fig. 16 shows that the mean horizontal velocity of the method trained at 25 AC/NM^2 is consistently higher, indicating a completely different strategy than that observed in the methods trained at a higher traffic density. This shows that the methods trained at a higher traffic density have potentially become more 'cautious'. Nevertheless, all methods favoured a reduction of horizontal velocity for conflict resolution, indicated by the drop in horizontal travel distance for increasing traffic densities.

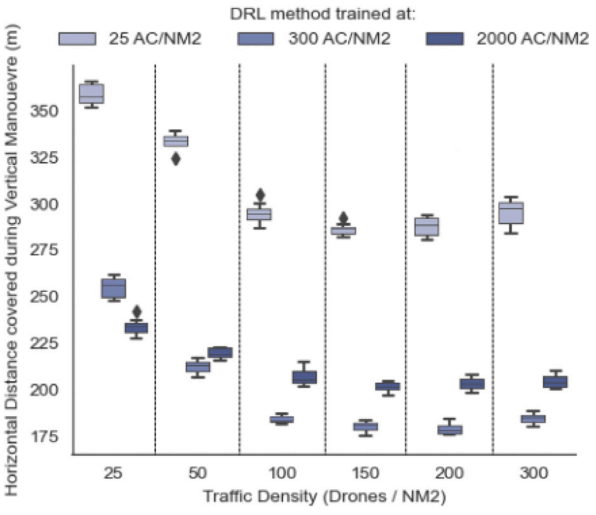


Fig. 16. Horizontal distance covered during the vertical manoeuvres at different traffic densities for the 'DRL' methods trained at a traffic density of 25, 300, and 2000 AC/NM^2.

6. Conclusions

This research investigated the impact of traffic density on the performance of an RL method on the task of conflict resolution. It was found that at lower traffic densities a shortest way out algorithm is more effective than the RL method, this difference however decreases at higher traffic densities, with the RL method eventually surpassing the shortest way out algorithm. Furthermore, the results show that training at higher traffic densities than used for testing can result in substantial improvements over the methods that have been trained at lower traffic densities. From this it is concluded that in order to obtain the most effective policy for the task at hand, training of the models should be done with scenarios of (much) higher complexity than those expected to be encountered during execution. This conclusion is based on two benefits associated with training at higher traffic densities, which can be generalized to environments in other domains. First, training at higher traffic densities leads to more diverse state transitions, creating a training data-set that contains a more diverse set of problems, some of which might have a very low probability of occurring at low traffic densities. This ensures that the method is still capable of dealing with these low frequency events. Secondly, higher traffic densities ensure that the reward signal associated with intrusions becomes less sparse, decreasing the imbalance present in the data at lower traffic densities. This decreases the bias of the learning algorithm towards the majority group by effectively decreasing the size of the majority group.

Still, there are some limitations to this research. First, even though the simulations were multi-agent, the specific episodes can be considered predominantly single-agent. As the vertically manoeuvring aircraft were solely responsible for maintaining safe separation with the cruising aircraft, the only observed multi-agent interactions were those between two or more vertically manoeuvring aircraft. This limits the results of this research to the single-agent domain. Future work should investigate whether the same observations are seen in the multi-agent domain. Secondly, some trend reversal was observed for the methods trained at high traffic densities while being tested at (much) lower traffic densities, specifically for the intrusion rate and intrusion severity. More research must be done on what is contributing to these observations in order to ensure that unexpected performance drops are limited during testing. Finally, as the results of this research are limited to the domain of conflict resolution, it is important that similar experiments are done in other domains, before a generalized conclusion about the importance of environment complexity for RL methods can be drawn.

CRediT authorship contribution statement

D.J. Groot: Methodology, Writing – original draft, Writing – review & editing. **J. Ellerbroek:** Conceptualization, Supervision. **J.M. Hoekstra:** Conceptualization, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data and Code is freely available on the 4TU repository and referenced in the manuscript.

References

Bilimoria, Karl, Sheth, Kapil, Lee, Hilda, Grabbe, Shon, 2000. Performance evaluation of airborne separation assurance for free flight. In: 18th Applied Aerodynamics Conference. p. 4269.

Doole, M.M., Ellerbroek, Joost, Hoekstra, J.M., 2018. Drone delivery: Urban airspace traffic density estimation. In: 8th SESAR Innovation Days.

Groot, Jan, 2022. Vertical conflict resolution in layered airspace with reinforcement learning using the BlueSky open air traffic simulator. URL https://data.4tu.nl/articles/software/Vertical_Conflict_Resolution_in_Layered_Airspace_with_Reinforcement_Learning_using_the_BlueSky_Open_Air_Traffic_Simulator/21572364/0.

Groot, Jan, Ribeiro, Marta, Ellerbroek, Joost, Hoekstra, Jacco, 2022. Improving safety of vertical manoeuvres in a layered airspace with deep reinforcement learning. In: International Conference on Research in Air Transportation (ICRAT) 2022.

Haarnoja, Tuomas, Zhou, Aurick, Hartikainen, Kristian, Tucker, George, Ha, Sehoon, Tan, Jie, Kumar, Vikash, Zhu, Henry, Gupta, Abhishek, Abbeel, Pieter, et al., 2018. Soft actor-critic algorithms and applications. arXiv preprint arXiv:1812.05905.

Hoekstra, Jacco M., Ellerbroek, Joost, 2016. 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, p. 132.

Hoekstra, Jacco M., van Gent, Ronald N.H.W., Ruigrok, Rob C.J., 2002. Designing for safety: the 'free flight'air traffic management concept. Reliab. Eng. Syst. Saf. 75 (2), 215–232.

Krawczyk, Bartosz, 2016. Learning from imbalanced data: open challenges and future directions. Progr. Artif. Intell. 5 (4), 221–232.

Mavic, DJI, 2022. Mavic pro - product information - DJI. URL https://www.dji.com/nl/mavic/info.

Mnih, Volodymyr, Kavukcuoglu, Koray, Silver, David, Rusu, Andrei A, Veness, Joel, Bellemare, Marc G, Graves, Alex, Riedmiller, Martin, Fidjeland, Andreas K, Ostrovski, Georg, et al., 2015. Human-level control through deep reinforcement learning. Nature 518 (7540), 529–533.

Ribeiro, Marta, Ellerbroek, Joost, Hoekstra, Jacco, 2020. Review of conflict resolution methods for manned and unmanned aviation. Aerospace 7 (6), 79.

Sunil, Emmanuel, Ellerbroek, Joost, Hoekstra, Jacco M, Maas, Jerom, 2018. Three-dimensional conflict count models for unstructured and layered airspace designs. Transp. Res. C 95, 295–319.

Sunil, Emmanuel, Hoekstra, JM, Ellerbroek, Joost, Bussink, Frank, Nieuwenhuisen, Dennis, Vidosavljevic, Andrija, Kern, Stefan, 2015. 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.

Sutton, Richard S., Barto, Andrew G., 2018. Reinforcement Learning: An Introduction. MIT Press.

TUDelft-CNS-ATM, 2022. BlueSky github. URL https://github.com/TUDelft-CNS-ATM/bluesky.

Wang, Zhuang, Pan, Weijun, Li, Hui, Wang, Xuan, Zuo, Qinghai, 2022. Review of deep reinforcement learning approaches for conflict resolution in air traffic control. Aerospace 9 (6), 294.