Lateral and Vertical Air Traffic Control Under Uncertainty Using Reinforcement Learning

C.A. Badea & D.J. Groot & A. Morfin Veytia & M. Ribeiro,
J. Ellerbroek, and J. Hoekstra
Control and Simulation, Faculty of Aerospace Engineering
Delft University of Technology, The Netherlands

R. Dalmau
EUROCONTROL Innovation Hub (EIH)
Bretigny-Sur-Orge, France

***Abstract*—Air traffic demand has increased at an unprecedented rate in the last decade (albeit interrupted by the COVID pandemic), but capacity has not increased at the same rate. Higher levels of automation and the implementation of decision-support tools for air traffic controllers could help increase capacity and catch up with demand. The air traffic control problem can be effectively modelled as a Markov game, where a team of aircraft (the agents) interact in the airspace (the environment) and cooperatively take resolution actions to achieve a common goal: safe separation in the most efficient way. As in any Markov game, the optimal policy for the team could be learnt through trial and error in a simulated environment using reinforcement learning algorithms. In this paper, we use the soft actor-critic algorithm to unravel the optimal air traffic control policy. Unlike some previous works, we propose a global (i.e., shared) reward that encourages cooperative behaviour. Furthermore, we propose a versatile policy model capable of performing heading, speed, and/or altitude resolution actions. We also demonstrate that the policy is robust and can maintain safe separation even in the presence of uncertainty regarding aircraft position, delays in implementing resolution actions, and wind. The findings of this paper also suggest that there is still significant room for improvement when controlling three degrees of freedom at the same time.**

***Keywords*—Air Traffic Control, Reinforcement Learning, Automation, Artificial Intelligence**

I. Introduction

The increasing delays and congestion reported in many areas indicate that the current air traffic control (ATC) system is rapidly approaching saturation levels [1]. The capacity of the system is limited by the number of available controllers and the number of aircraft that each controller can manage. Consequently, research focus has been redirected to automated tools and alternative approaches that can automate part of the ATC process, especially in high traffic scenarios and in the presence of uncertainties, and thereby increase capacity.

Many deterministic conflict detection and resolution methods have been developed for aviation [2]. These have shown to be very successful in situations with a small number of aircraft. However, as the traffic density increases, these methods are hindered by the unpredictable behaviour resulting from multiple aircraft interacting. Knock-on effects of aircraft avoiding each other may result in unforeseen trajectory changes. In addition, these methods typically have limited rules that cannot guarantee safety in all different situations. However, manually increasing these rules is not trivial; these must be arduously defined by experts. This has redirected the focus to methods capable of adapting to the environment autonomously.

The ATC problem can be modelled as a Markov game in which multiple agents (the aircraft) interact in the airspace (the environment) with the same goal in mind: to ensure safe separation as efficiently as possible. Humans may find it difficult, if not impossible, to analyse and comprehend emergent behaviour in a multi-agent system. However, reinforcement learning (RL) techniques are often capable of identifying emerging patterns through repeated exposure and training in their environment. Furthermore, RL can adapt to high degrees of uncertainty regarding the position of other aircraft, delays in implementing resolution actions, and unpredictability of the weather, all of which are tested in this work.

Although RL techniques have previously been used in the ATC domain [3], few studies have focused on providing a unified approach for aircraft separation assurance and routing. This paper proposes a soft-actor-critic (SAC) algorithm for determining the optimal cooperative policy that leads all aircraft to the target in the most efficient manner, while maintaining safe separation. The SAC algorithm has been shown to produce good results when controlling aircraft's trajectory, and safekeeping a minimum distance from obstacles or other aircraft [4], [5]. Additionally, it has been proven that an RL agent can simultaneously control vertical and lateral actions [6]. Finally, previous work also suggests that RL agents can handle uncertainties regarding the position of other aircraft [7], [8]. However, to the best of the authors' knowledge, this is the first work to develop an RL agent responsible for controlling multiple en-route aircraft simultaneously with both lateral and vertical actions, under different degrees of uncertainty.

Section II describes the RL method: algorithm, the observation, action, and reward formulations. Different formulations are used depending on whether the RL method performs (1) heading, and speed variation, or (2) heading, speed, and altitude variation. The simulation scenario, built on top of the Gym open-source Python library [9], is detailed in Section III. The defined hypotheses are presented in Section IV. Section V describes the results of both training and testing of the RL method. All the code used to run the results is open-source and is available at [10]. Finally, Sections VI and VII present the discussions of the results and conclusion, respectively.

II. Method

The RL model developed in this work has the objective of guiding all aircraft to target, while preventing aircraft from getting closer to each other than the minimum separation distance. The latter is equal to 5 NM, as defined by ICAO [11]. When two aircraft are closer than this minimum separation distance, we consider that an intrusion has occurred.

Sections II-A and II-B describe the Markov game abstracting the air traffic control problem and the fundamental principles of the soft actor-critic (SAC) algorithm, respectively. Section II-C describes the steps taken in one round of simulation. Finally, Sections II-D, II-E, and II-F specify the information received, the actions performed, and the reward given to the RL model, respectively.

A. Markov Game Abstraction of the ATC Problem

A Markov game with homogeneous agents can be represented as a tuple (N, S, A, P, R, ρ_0) consisting of the number of agents, the state of the environment, the set of actions available to each agent, the transition probability function, the reward function of each agent, and the initial state distribution, respectively. Because the ATC is a cooperative Markov game, all agents share the same reward function. The partially observable Markov game is a special case in which agents do not have access to the entire state $s \in S$ of the environment before choosing an action, but only a portion $o \in \Omega$ of it, as determined by the observation function $O : S \times \Omega \to [0, 1]$.

B. Soft Actor-Critic Algorithm

For this research, the SAC algorithm is used in combination with the automatic tuning of entropy, as proposed by Haarnoja [12]. SAC is an off-policy, model-free algorithm acting in the continuous action domain. As an actor-critic algorithm, it employs the best of value and policy-based methods, learning both a policy and a value function. Additionally, SAC has been shown to outperform other state-of-the-art algorithms, including the well-known deep deterministic policy gradient (DDPG). It is beyond the scope of this paper to describe the mathematical implementation of SAC. A reader interested in this algorithm is directed to reference [13].

The general objective of a reinforcement learning agent is to maximise the discounted cumulative reward (also known as the return in the RL jargon). With SAC, this objective is augmented with an entropy term, which means that randomness/exploration is promoted in areas where the reward might still be uncertain. This allows the algorithm to better deal with local optima by better exploring the available solution space. The source code of the SAC implementation used in this paper (including the hyper-parameters) is available in reference [10] so that any researcher can reproduce the results shown herein.

C. Interaction of the RL Model with the Environment

At each time step, the RL model receives information from each aircraft (see Section II-D), and outputs an action for each in response. This action identifies the state that the aircraft will adopt in the current timestep. The elements of the aircraft's state modified by the RL model are specified in Section II-E. Each aircraft is analysed in parallel; the actions picked for each aircraft do not influence the decision of the model for the other aircraft. The value of each action is evaluated at the end of the time step. The reward given to the RL model is based on the position and state of each aircraft at the end of the time step (see Section II-F).

Note that, in this work, the observation and action formulations have different dimensions and content depending on whether the RL method is also responsible for altitude variation on top of heading and speed variation. Tables I, II, and III show the observation function, the action space, and the reward function, respectively.

D. Observation Function

The observation function per agent consists of two main parts: (1) the state of the ownship, and (2) its proximity and direction to the surrounding aircraft. A graphical representation of the latter is given in Fig. 1. Surrounding aircraft are represented through their distance to the ownship (both vertical and horizontally), and relative heading. Additionally, the RL method has information on the current speed of the ownship, its desired cruising speed, and the bearing to target. These allow the RL method to instruct the ownship to follow its desired speed as much as possible. Note that the desired cruising speed is a random speed value between the maximum and minimum speed values of the aircraft. The complete observation function, including an explanation of the elements shown in Fig. 1, can be found in Table I. The observation has a dimension of $4 + 5n$ when altitude variation is not employed, and $5 + 6n$ when it is, with n being the number of surrounding aircraft included in the observation. The latter must be defined beforehand.

The number of aircraft represented in the observation is not a trivial decision. On the one hand, the more information the RL method has, the better it can defend against intrusions. It may even be that the RL method cannot prevent a short-term intrusion, as it does not receive information on this aircraft. On the other hand, the larger the observation function is, the longer the method will have to train in order to learn the optimal

TABLE I. Observation function of the RL method. Note that extra information is added when the RL method can also perform altitude deviation.

Variable	Size
Current relative distance to aircraft $[d_t]$	#aircraft
Expected relative dist. future time step to aircraft $[d_{t+4}]$	#aircraft
Distance to aircraft in the x axis $[d_x]$	#aircraft
Distance to aircraft in the y axis $[d_y]$	#aircraft
Relative heading to aircraft $[d_{hdg}]$	#aircraft
Ownship airspeed	1
Ownship optimal airspeed	1
Bearing to target (*sin* component)	1
Bearing to target (*cos* component)	1
Only with altitude deviation:	
Ownship current altitude	1
Altitude difference to aircraft	#aircraft

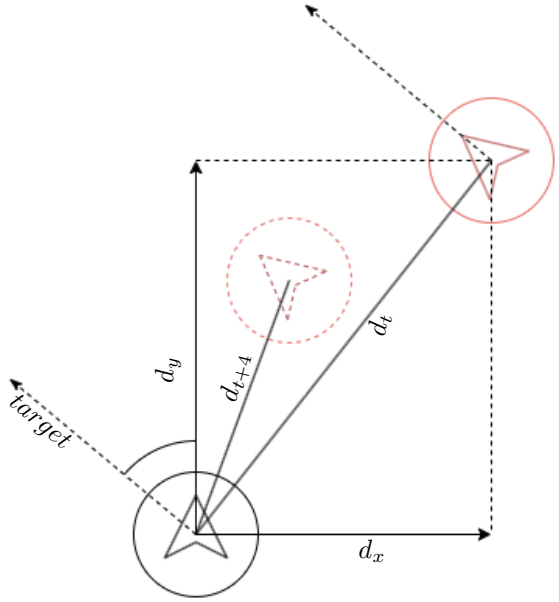

Figure 1. Graphical overview of the representation of surrounding aircraft in the function observation of the RL method. The variables are identified in Table I.

actions from a larger set of observation-action combinations. Moreover, it is not certain that depicting more aircraft directly correlates to a more efficient intrusion prevention. Often with geometric intrusion prevention algorithms, the look-ahead time for intrusions is limited to favour actions that prevent short-term intrusions as efficiently and as fast as possible [14]. Additionally, uncertainties regarding the relative future states of the aircraft increase with far-away aircraft.

In this work, we test different numbers of surrounding aircraft in the observation function to identify the ideal number of surrounding aircraft. However, this number is directly associated with the simulation environment tested. For different environments, a different number of aircraft, and, consequently, a different observation function may be preferable.

E. Action Space

The RL method controls the state of all aircraft with the objective of leading them towards their target point without intrusions. Two different degrees of freedom are tested for the action space: (1) the RL method controls the heading and speed variation of each aircraft, (2) the RL method controls the heading, speed, and altitude variation of each aircraft.

All actions are computed using a *tanh* activation function; this maps the output values of the RL agent between -1 and +1. These values are then translated into a variation of the state of the ownship, as identified in Table II. For heading, negative and positive values correspond to counter-clockwise and clockwise heading variations, respectively. With respect to speed, with each action the ownship may reduce or increase its speed up to a third of the speed performance range. Note that the speed of an aircraft is always restricted to the performance limits. The altitude variation is the only element that is not linear. We use

TABLE II. ACTION FORMULATION OF THE RL METHOD. Δv REPRESENTS THE DIFFERENCE BETWEEN THE MAXIMUM AND MINIMUM CRUISING SPEED OF AN AIRCRAFT.

Property	Value by the RL method
Heading	[-1,+1] transforms to [-22.5°,+22.5°]
Speed	[-1,+1] transforms to [-$\Delta v/3$,+$\Delta v/3$]
Only with altitude deviation:	
Altitude	Secondary altitude $if > 0$, else primary altitude

two altitude levels to limit the number of altitude deviations performed by the RL method, which is often not preferred due to increased fuel consumption and passenger comfort.

F. Reward Function

The reward formulation of the RL method is represented in Table III. The most important value is the number of intrusions (i.e., agents must give priority to safety), and thus receives the biggest penalty (-10 per intrusion). After intrusions, minimising drift angle to ensure flying towards the target is prioritised. This is done by a drift penalty with respect to the ideal track angle. Thus, aircraft are motivated to travel in a straight line towards their target. In this case, the drift reward is equal to $+1$. Moving away from the target is penalised with negative values, up to a maximum of -0.5. Moving at a speed different from the preferred cruising is also penalised. However, this is a small penalisation as speed variation should be preferred over moving away from target to prevent intrusions. Finally, moving to the secondary altitude level (versus the main travel altitude) is heavily penalised. We consider that the RL method should only resort to this second flight level when heading and speed variation alone are not sufficient to prevent the intrusion.

This work uses global rewards, which has previously been used in research to prevent selfish policies [15], [16]. This is the summed value of the rewards observed by all aircraft currently acting within the environment. However, often it is not apparent which aircraft/action is to blame for the intrusion. A global reward ensures that actions by third parties, which are not directly involved in the intrusion but have a negative influence, can also be penalised. However, there are also disadvantages to a global reward. It often leads to a 'credit assignment' problem: it is difficult for an agent to correctly understand the effect of its own actions when the reward received is the aggregation of the actions of all agents. When it is not clear how its own actions contribute to the global reward, an agent may incorrectly change its policy to poorer actions, slowing down (or even hampering) its learning process.

TABLE III. REWARD FORMULATION OF THE RL METHOD. NOTE THAT THE ALTITUDE LEVEL IS ONLY ADDED WHEN THE RL METHOD CAN ALSO PERFORM ALTITUDE DEVIATION.

Variable	Weight
Number of intrusions	-10
Drift penalty	(0.5-abs(*drift*))*0.2
Speed difference from optimal	-0.001* Δv
Only with altitude deviation:	
If aircraft in secondary altitude level	-9

III. Experiment: En-route aircraft control with Reinforcement Learning

A. Simulation Scenario

Each traffic simulation scenario has an area shape and flight routes which are randomly generated. An example is given in Fig. 2; the airspace takes the shape of a polygon with an area between $125\,\mathrm{NM}^2$ to $250\,\mathrm{NM}^2$. The spawn and target points of each aircraft are generated randomly within this area. However, it should be noted that aircraft are allowed to travel outside of the area when necessary to avoid intrusions. Moreover, spawn points are generated at least at a minimum separation distance from each other, in order to avoid intrusions when aircraft spawn too close to each other.

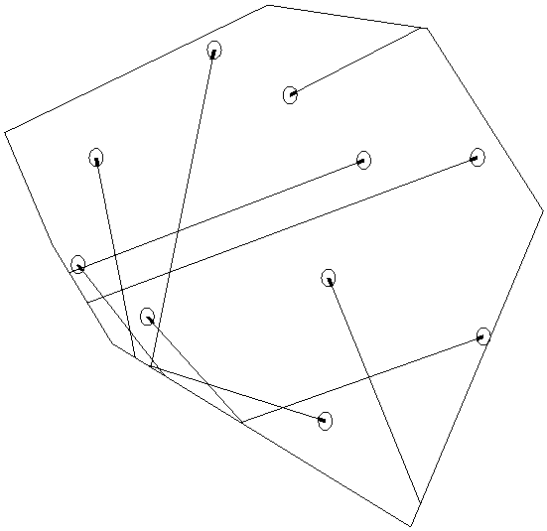

Figure 2. Example of a simulated scenario randomly generated.

B. Simulation Traffic

A fixed number of aircraft are created at the beginning of the simulation. Each scenario runs until each of these aircraft reaches its target or when the simulation reaches 300 time steps. A step represents 5 seconds of simulation time. Aircraft are homogeneous in performance, with a minimum and maximum speed of 400 and $500\,\mathrm{kt}$, respectively. However, each aircraft has a preferred cruising speed, which is randomly selected at the beginning of the simulation.

Additionally, we assume instantaneous changes in heading, speed, and altitude by each aircraft as indicated by the RL method. Although such a portrait of aircraft dynamics is not realistic, it maximises the performance of the RL method. The final reward is thus a direct result of the new state of the aircraft, and not of an intermediate state between the previous and the new state due to acceleration limitations. As this work mainly aims at identifying the maximum potential of RL for separation assurance, this approach was preferred. However, future efforts should focus on repeating this work with realistic aircraft dynamics, e.g., using the base of aircraft data (BADA) or open-source simulators such as Bluesky [17].

C. Intrusions and Minimum Separation

A horizontal separation of $5\,\mathrm{NM}$ is used, as defined by ICAO [18]. An intrusion occurs every time the distance between two aircraft is less than this separation distance. When the RL method also varies altitude, a vertical separation is implemented. However, no vertical distance is defined, as aircraft perform an immediate climb. Thus, we consider that aircraft in different vertical layers are safely separated.

D. Independent Variables

Different independent variables were tested over the course of 3 sub-experiments. These are described below:

1) No Uncertainties

The RL method is trained without uncertainties. We analyse whether the method can understand reaching the target, and prevent intrusions through heading and speed variation while doing so. Only one altitude level is considered, as at this phase the RL method does not control the flight level of aircraft. Moreover, the RL method is trained with a different number of surrounding aircraft, two and four, represented in the observation function. The objective is to find which function leads to stronger prevention of intrusions.

2) With Uncertainties

The RL method is trained with 50% and 100% probability per time step of position uncertainties and action delays. Wind is also added to the testing environment in an attempt to make the operational environment more realistic. These elements are implemented as follows:

- Position uncertainty: in every simulation step, there is a probability that the position of a surrounding aircraft, as received by the ownship, suffers a random offset between $0\,\mathrm{m}$ to $500\,\mathrm{m}$ in any direction.
- Wind: for each episode, a uniform wind field is initialised with a random direction and magnitude between $0\,^\circ$ to $360\,^\circ$ and $0\,\mathrm{m/s}$ to $30\,\mathrm{m/s}$, respectively.
- Action execution delay: for every action (once per simulation step), there is a chance that a delay is experienced. The delay is set between $0\,\mathrm{s}$ to $3\,\mathrm{s}$. In practise, this means that an aircraft does not immediately adopt the new state output by the RL method but, instead, continues with the previous state for the duration of the delay. The aircraft adopts the new state after this delay.

3) Effect of Altitude

The method is trained and tested with two altitude levels. A second level is added to the environment, meant to be used as a last resort to prevent intrusions when heading and speed variation are not sufficient. Here, the RL method can vary the altitude on top of heading and speed variation. We consider an aircraft to have reached its target point when it is in the main altitude layer. Moreover, the RL is trained twice, one time with 10 and a second time with 20 aircraft. The traffic density is doubled with the objective of creating multi-actor situations where intrusions may only be prevented with vertical deviation.

E. Dependent Measures

The performance of the RL method is evaluated in terms of both safety and flight efficiency. Safety-wise, the total number of intrusions is considered. The method is also evaluated in terms of how efficiently it prevents intrusions, namely the total number of steps necessary for all aircraft to reach the target, as well as the speed variation necessary to prevent intrusions.

IV. Experiment: Hypotheses

It is hypothesised that the RL method can reduce the number of intrusions compared to a baseline situation in which aircraft move in a straight line towards their target. Furthermore, it is hypothesised that the efficacy of the method in preventing intrusions decreases as uncertainty increases. We also anticipate a greater increase in the number of intrusions when the action delay is implemented, as this will limit the efficacy of the method in preventing short-term intrusions.

Regarding the number of surrounding aircraft in the observation function, it is hypothesised that knowing about more aircraft will help the method to be more effective in its intrusion prevention. A larger observation function is also expected to result in longer training times, so the RL method will need to run more episodes to determine the best policy.

The efficacy of the RL method in preventing intrusions depends on the degrees of freedom controlled by the method. Although more degrees of freedom lead to a higher number of actions that the method may resort to in order to prevent an intrusion, it is hypothesised that a larger action space (e.g., with vertical actions) will make it harder for the method to identify optimal actions for each observation function.

Finally, it is hypothesised that the self-separation concept learnt by the method is independent of the traffic density. However, higher traffic densities lead to more complex multi-actor situations. Therefore, it is likely that the RL method requires more information on the surrounding aircraft in the observation function than at a lower traffic density. Thus, it is also hypothesised that the method will not be as efficient in preventing intrusions at higher traffic densities.

V. Experiment: Results

1) No Uncertainties

Figs. 3-5 show the evolution in performance during training without uncertainties with just 2 degrees of freedom: heading and speed. The progression over time is shown for both 2 and 4 surrounding aircraft included in the observation function. In the figures, 'MA100' represents the moving average of the last 100 episodes. Fig. 3 shows the evolution of the number of intrusions during training. Here, it is shown that the number of intrusions is effectively minimised after 200 and 350 episodes with 2 and 4 aircraft in the observation function, respectively. Throughout the training, only including 2 surrounding aircraft in the observation function was more stable, obtaining fewer than 1 intrusion per episode on average.

Fig. 4 shows the evolution of the number of total time steps per episode. Naturally, moving aircraft out of their straight path towards the target to prevent intrusions will result in longer episodes. However, the increase in the number of steps is relatively small. For example, with 2 surrounding aircraft, it takes, on average, 10 more time steps to lead all aircraft towards their target point. This indicates that small heading and speed variations are executed to prevent intrusions.

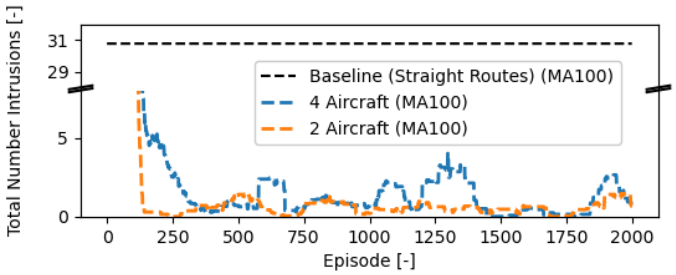

Figure 3. Evolution of the number of intrusions during training.

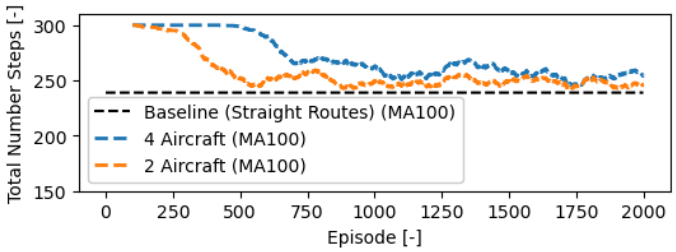

Figure 4. Evolution of the number of steps during an episode during training.

Fig. 5 shows the absolute difference between the preferred cruising speed of the aircraft and the speed output by the RL method. An average $15\,\mathrm{m/s}$ speed difference from the preferred cruising speed (for 2 surrounding aircraft), is roughly $35\,\%$ of the total speed range. This difference is expected, as speed variation was given a smaller penalty than heading variation.

Fig. 6 shows the actual routes controlled by the RL method for the traffic scenario in which the planned routes were shown in Fig. 2. Regarding the planned scenario (Fig. 2), most of the routes in Fig. 6 take the aircraft on a direct path to the target. The only significant heading changes occur when two aircraft approach each other. Even so, once the ownship is no longer at risk, it quickly redirects to the target, as shown in the red circle. Both aircraft in the orange circle deviated from their intended path to avoid a collision. As a result, it appears that the method can determine the required heading deviation.

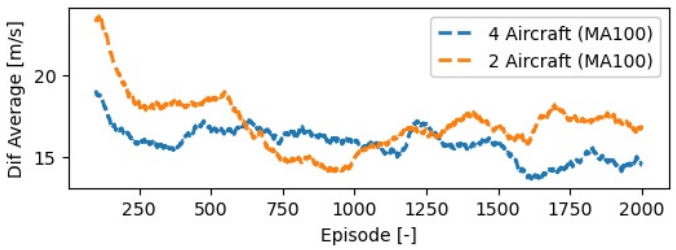

Figure 5. Evolution of the difference between the current and preferred cruising speed during training.

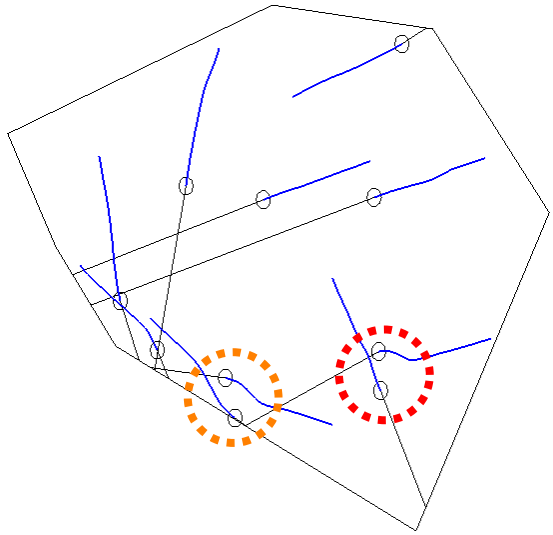

Figure 6. Final routes defined by the RL method in the example scenario.

Fig. 7 shows the distance between aircraft through 500 testing episodes. The boxplot on the left shows the distance between aircraft when an intrusion has occurred. These values can also be used to quantify the intrusion severity. Given the results, most intrusions have a low severity: the minimum distance between aircraft is between 4 and 5 NM.

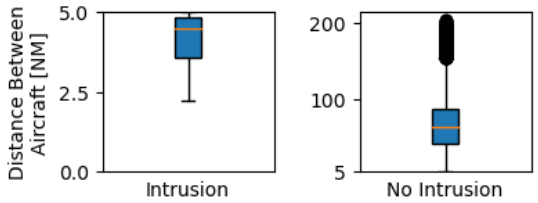

Figure 7. Distance between aircraft throughout an episode.

Table IV compares the testing results obtained when the method trained with 10 aircraft is tested with 10 and 20 aircraft. In this case, we used the observation function including the 2 closest surrounding aircraft, as its performance proved to be more stable. With 20 aircraft, separation assurance is required at double the training traffic density. Still, all aircraft are successfully guided to their final target. However, preventing intrusions proves to be more difficult with higher traffic densities. This indicates that training probably has to be done at least at the same traffic densities expected during testing. Additionally, the current observation function might not contain enough information to prevent intrusions with a higher number of aircraft, as the observation only contains information on the closest 2 aircraft.

TABLE IV. TOTAL INTRUSIONS AND STEPS WHEN THE RL METHOD IS TESTED WITH THE SAME (10 AIRCRAFT), AND WITH HIGHER TRAFFIC DENSITIES (20 AIRCRAFT).

	Number of Intrusions (MA100)	**Number of Steps (MA100)**
10 aircraft	0.52	248.07
20 aircraft	7.22	272.96

To better understand the actions selected, Figs. 8 and 9 show the average selected heading and speed change with respect to the position of the surrounding aircraft, respectively. It is important to note the strong preference to rotate counterclockwise to resolve imminent intrusions, as indicated by the red area in Fig. 8. The same policy is shared between all agents in the environment, ensuring implicit coordination in pairwise intrusions.

Fig. 9 shows a tendency to accelerate or decelerate depending on the position of the nearby aircraft. At very small absolute distance values, the RL method produces the strongest speed variation actions. However, Fig. 9 shows a very strong acceleration value directly next to a strong deceleration value (see points near (0, 0)). Strong acceleration values can be understood as the method attempting to move the ownship away from the surrounding aircraft as fast as possible. However, the strong deceleration is not as clear. Future work will increase the training complexity of the RL method to further clarify this behaviour.

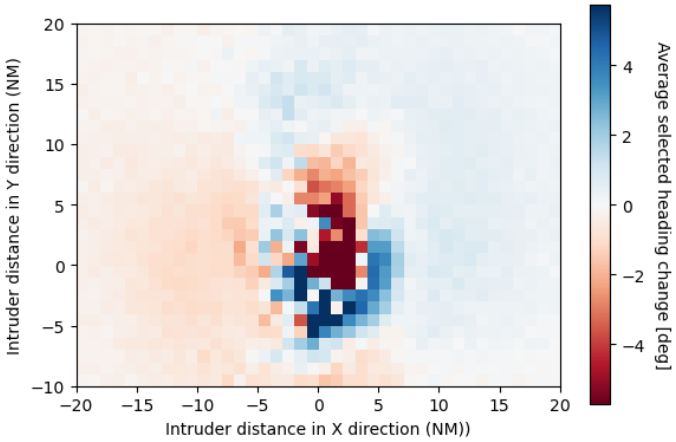

Figure 8. Heading deviation in relation to the average distance to surrounding aircraft.

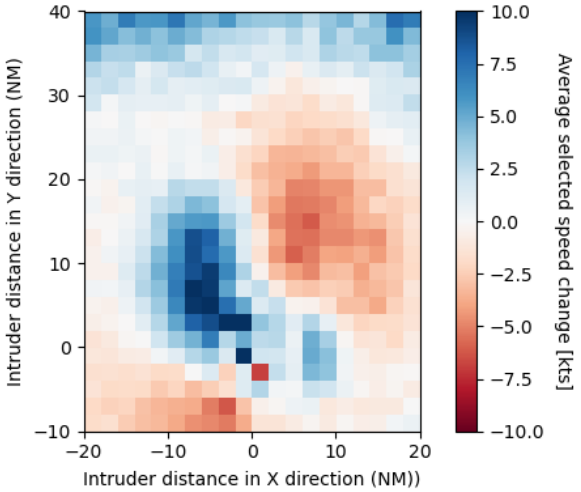

Figure 9. Speed deviation in relation to the average distance to surrounding aircraft.

Figs. 10 and 11 show the same policy for filtered head-on intrusions (defined by a relative heading between 160 ° to 200 °). The agent realises the danger, indicated by the higher magnitude of the selected actions. Furthermore, there is a clear acceleration present just before intrusion (see Fig. 11) that, combined with the strong heading change shown in Fig. 10, results in a final attempt to prevent an intrusion.

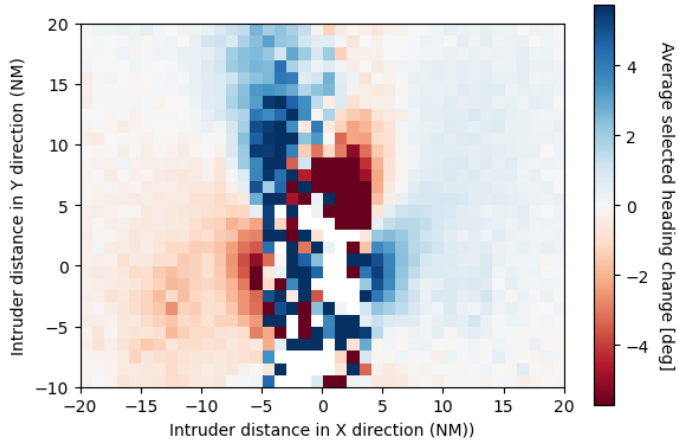

Figure 10. Heading deviation in relation to the average distance to surrounding aircraft in the case of (near-)head-on intrusions.

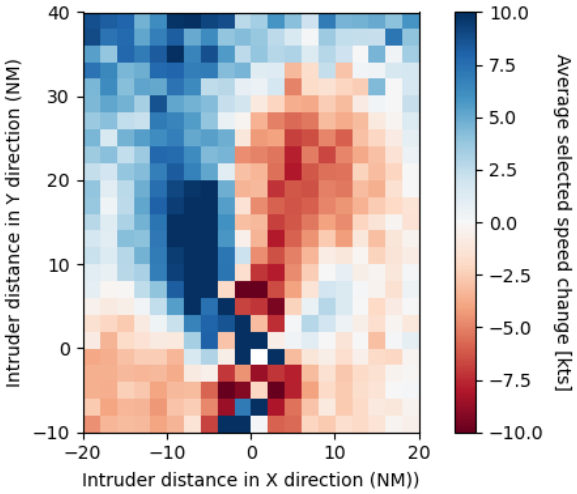

Figure 11. Speed deviation in relation to the average distance to surrounding aircraft in the case of (near-)head-on intrusions.

2) With Uncertainties

Table V shows the average number of intrusions and steps for different RL methods trained in the operational environment with the level of uncertainty defined in the first column. Here, training is done with 2 surrounding aircraft in the observation function. As hypothesised, adding any kind of position uncertainty, wind conditions, or action delays with a probability of 100 %, results in an increase in intrusions.

Surprisingly, the average number of intrusions decreases slightly when position uncertainty and wind are added to the environment with a probability of 50 %. Naturally, 50% probability may not be sufficient to influence the performance. However, it could also be that position uncertainty results in the method defending in advance due to the incorrect perception that other aircraft are closer than they really are. Uncertainties could also lead to 'better' geometries. For example, head-on pairwise intrusions are practically impossible to prevent without a coordinated manoeuvre from both aircraft. In this case, uncertainties may lead to each aircraft thinking that the other is not directly head-on but more to one side, which facilitates the decision to which side the ownship should turn to in order to prevent the intrusion.

TABLE V. RESULTS AFTER TRAINING CONVERGENCE WHEN UNCERTAINTIES ARE ADDED TO THE ENVIRONMENT.

	#Intrusions (MA100)	**#Steps (MA100)**
No uncertainties	0.52	248.07
50% prob. position uncertainty + wind	0.48	248.85
100% prob. position uncertainty + wind	0.79	250.76
50% prob. action delay	0.78	248.39
100% prob. action delay	0.83	243.86

Finally, the average number of intrusions increases when there is a delay in aircraft adopting the provided action. This is expected as: (1) aircraft may not adopt the new state fast enough to prevent short-term intrusions, (2) the longer aircraft take to adopt the new state, the more likely it is that this state will no longer be effective against the intrusion. The previous action was calculated by the RL method assuming that the ownship would act immediately. With less time to prevent intrusions, stronger heading or speed variation is needed to prevent the more imminent intrusion. The number of steps decreases when the probability of a delay is at 100 %. As the ownship takes longer to adopt the new intrusion prevention heading, it spends more time on its 'default' trajectory, which is directed towards the target. However, aircraft reach the target faster at the expense of a higher number of intrusions.

3) With Vertical Deviation

Table VI shows the evolution of the RL method trained with altitude variation, on top of heading and speed variations. With 10 aircraft, the method achieved 0.88 intrusions per episode, compared to 0.52 intrusions with only heading and speed variation (see Table V). This increase was not expected. The RL method did not learn to use the altitude level for intrusion prevention as desired. A possible reason is that, with only 10 aircraft, most intrusion situations can be solved with heading and speed variation alone. Thus, there are not enough situations where the method is motivated to learn to use the additional altitude layer.

TABLE VI. RESULTS AFTER TRAINING CONVERGENCE WHEN VERTICAL MANOEUVRES ARE ADDED TO THE ACTION SPACE.

	#Intrusions (MA100)	**#Steps (MA100)**
Vertical deviation (with 10 aircraft)	0.88	249.97
Vertical deviation (with 20 aircraft)	1.48	281.31

The method was also trained with twice the traffic density, 20 aircraft, to force more danger situations involving multiple aircraft. With deterministic intrusion prevention methods, having twice the traffic density more than doubles the number of intrusions [19]. However, this is not observed in this case. With 10 and 20 aircraft, the average number of intrusions is 0.88 and 1.48, respectively. This indicates that the RL method can adapt to higher traffic densities when trained in it. However, it should be noted that the higher the traffic density, the longer it takes the method to identify optimal solutions.

With 20 aircraft, the RL method still predominantly chose to use heading and deviation alone to prevent intrusions. With 1.48 average intrusions, there are still not enough intrusions for the RL method to learn to use altitude deviation correctly. We consider that, with three degrees of freedom combined, the method requires a larger set of training intrusions directed at understanding the advantage of varying each degree.

VI. Discussion

This work explored whether RL methods can successfully control the trajectory of all aircraft in a multi-agent, en-route sector. The results show that an RL method is capable of guiding aircraft to the target point, while preventing each aircraft from getting closer than the minimum separation distance. Moreover, this behaviour was tested under uncertainties regarding the position of the surrounding aircraft, as well as delays in adoption of the method's produced state. Although these hindered the ability of the method to defend against intrusions, the RL method was still able to guide aircraft to target with limited intrusions.

The amount of information to add to the observation function is an important factor. On the one hand, a larger observation array provides more information about the environment in which it operates. On the other hand, more information could be redundant and slow down the learning process. This can be seen in Fig. 3, where considering 4 surrounding aircraft was not as stable as considering only 2. For these experiments, the traffic density may have been low enough so that information about more surrounding aircraft was unnecessary. Aircraft, other than the 2 closest ones, did not pose an imminent threat to the safety of the ownship. However, for larger densities, it could be that information about 2 surrounding aircraft is not enough, as other nearby aircraft may cause future intrusions. This can be seen in Table IV: doubling the number of aircraft to 20 created, on average, more than 7 intrusions per episode.

In addition, this work explored the effect of the number of degrees of freedom on the ability to prevent intrusions. With heading and speed variation, the method was able to choose the necessary variation to prevent intrusions in a continuous space. Table VI shows that, without uncertainties, the method for the 10 aircraft scenario averaged fewer than 1 intrusion per episode. A larger action space, with altitude variations on top of heading and speed, decreased the efficacy. The method never learnt to use altitude variation as an intrusion prevention tool, possibly as a consequence of the implementation characteristics of this work. First, altitude was given as a discrete action. This inconsistency with the two other degrees of freedom may have contributed to the decline in performance. Second, the number of multi-actor situations might not have been sufficient to successfully learn to use altitude deviation as the last resource to prevent intrusions.

A. Future Work

There is still ample work to be done on the development and testing of reinforcement learning methods for ATC before these can be implemented in a real-world scenario. First, the effect of different observation functions must be further investigated. Future studies should focus on analysing the essential information that is required in order to be able to prevent multi-actor intrusions. Furthermore, it should be taken into account that not all surrounding aircraft pose the same threat. For example, an aircraft travelling near the ownship, but in the same direction, is not as risky as one further away but heading towards the ownship. Prioritisation of certain aircraft can help limit the size of the observation function while still guaranteeing that sufficient information is provided in order to prevent intrusions.

Additionally, this work assumes the immediate adoption of the provided actions. This is far from a real-world scenario, where performance limits dictate how fast an aircraft can modify its state. In a worst-case scenario, an aircraft may not be able to adopt an intrusion prevention state fast enough to prevent a short term intrusion. Furthermore, this affects the predictability of the state transition function. The change to the environment may not be a direct result of the action output, but instead of the maximum state change that the ownship managed to achieve in a limited amount of time.

Finally, uncertainties regarding the positions of the surrounding aircraft affect the ability of the RL method to defend against intrusions. However, teaching the method to defend in advance against these uncertainties may result in the agent adopting a more defensive state, considering larger distances of minimum separation. These lead to greater state deviations, which are not desirable as they increase flight time, and consequently, fuel consumption. An alternative is to consider other machine learning methods that analyse environmental conditions and provide the RL method with an accurate estimate of the position of other aircraft [20].

VII. Conclusion

This paper is exploratory work to identify whether reinforcement learning (RL) can be used as a complete ATC tool. This work introduced an RL method responsible for leading the aircraft to the target as fast as possible, while keeping all the aircraft at a safe separation distance. The method receives only the information available to each aircraft, namely their current state and the relative position of the surrounding aircraft. Under nominal conditions, the RL method is capable of safely guiding 10 aircraft to their target point with minimum heading and speed deviation. Even in the presence of uncertainties regarding the position of other aircraft, action delay, and wind, the average number of intrusions is fewer than 1.

Future work should focus on extending these results to environments with more realistic aircraft dynamics, such as horizontal and vertical accelerations. Additionally, specific training scenarios should be developed towards reducing the effect of uncertainties on the efficacy of the method. A possible solution may include the creation of additional machine learning methods that provide an accurate estimate of the position of nearby aircraft.

ACKNOWLEDGEMENTS

The authors would like to thank the EUROCONTROL Innovation Hub for creating and coordinating the Master Class challenge 'Conflict Resolution with Reinforcement Learning', which led to the creation of this work. Furthermore, we would like to congratulate the other teams that participated in this challenge, especially Cranfield University, with whom we had interesting and fruitful conversations regarding different approaches to this problem.

REFERENCES

[1] "Performance review report covering the calendar year 2018," EUROCONTROL, Tech. Rep., 2018.

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

[3] 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, 2022.

[4] M. H. Lee and J. Moon, "Deep reinforcement learning-based uav navigation and control: A soft actor-critic with hindsight experience replay approach," 2021.

[5] J. Groot, M. Ribeiro, J. Ellerbroek, and J. Hoekstra, "Improving Safety of Vertical Manoeuvres in a Layered Airspace with Deep Reinforcement Learning," *10th International Conference for Research in Air Transportation (ICRAT)*, 2022.

[16] M. W. Brittain, X. Yang, and P. Wei, "Autonomous separation assurance with deep multi-agent reinforcement learning," *Journal of Aerospace Information Systems*, vol. 18, no. 12, pp. 890–905, 2021.

[6] J. Mollinga and H. van Hoof, "An autonomous free airspace en-route controller using deep reinforcement learning techniques," 2020.

[7] D.-T. Pham, P. N. Tran, S. Alam, V. Duong, and D. Delahaye, "Deep reinforcement learning based path stretch vector resolution in dense traffic with uncertainties," *Transportation Research Part C: Emerging Technologies*, vol. 135, p. 103463, 2022.

[8] M. Brittain, X. Yang, and P. Wei, "A deep multi-agent reinforcement learning approach to autonomous separation assurance," 2020.

[9] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," 2016.

[10] A. Badea, R. Dalmau, D. Groot, A. M. Veytia, and M. Ribeiro, "Air Traffic Control RL Environment," DOI: 10.4121/20455296, 2022.

[11] ICAO, "Doc 4444 - pans-atm, procedures for navigation services – air traffic management," 2016.

[12] 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.

[13] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel, and S. Levine, "Soft actor-critic algorithms and applications," 2018.

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

[15] R. Isufaj, D. Aranega Sebastia, and M. Angel Piera, "Towards Conflict Resolution with Deep Multi-Agent Reinforcement Learning," in *ATM seminar 2021, 14th USA/EUROPE Air Traffic Management R&D Seminar*, 2021.

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

[18] I. C. A. Organization, *Doc 4444: Procedures for air navigation. Air Traffic Management*, sixteenth ed., 2016.

[19] E. Sunil, J. Ellerbroek, J. 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, 10 2018.

[20] Z. Wang, M. Liang, and D. Delahaye, "Data-driven Conflict Detection Enhancement in 3D Airspace with Machine Learning," in *2020 International Conference on Artificial Intelligence and Data Analytics for Air Transportation (AIDA-AT)*, 2020, pp. 1–9.