Conflict Resolution at High Traffic Densities with Reinforcement Learning

Ribeiro, M.J.

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

CONFLICT RESOLUTION AT HIGH TRAFFIC DENSITIES WITH REINFORCEMENT LEARNING

CONFLICT RESOLUTION
AT HIGH TRAFFIC DENSITIES
WITH REINFORCEMENT LEARNING

———

Dissertation

for the purpose of obtaining the degree of doctor
at Delft University of Technology
by the authority of the Rector Magnificus prof. dr. ir. T.H.J.J. van der Hagen
chair of the Board for Doctorates
to be defended publicly on
Friday , 17 February 2023 at 12:30 o'clock

by

Marta Joana RIBEIRO

Master of Science in Aerospace Engineering
Instituto Superior Técnico, Portugal

born in Lisbon, Portugal

This dissertation has been approved by

Promotor: Prof. dr. ir. J.M. Hoekstra
Copromotor: Dr. ir. J. Ellerbroek

Composition of the doctoral committee:

Rector Magnificus,	chairperson
Prof. dr. ir. J.M. Hoekstra	Delft University of Technology, *promotor*
Dr. ir. J. Ellerbroek	Delft University of Technology, *copromotor*

Independent members:

Prof. dr. ir. S. Hoogendoorn	Delft University of Technology
Prof. dr. D.G. Simons	Delft University of Technology
Prof. dr. D. Delahaye	Ecole Nationale de l'Aviation Civile
Dr. P. Wei	George Washington University
Dr. ir. E. Sunil	Nederlands Lucht- en Ruimtevaartcentrum

Reserve member:

Prof. dr. ir. M. Mulder	Delft University of Technology

Keywords:	Air Traffic Control, Conflict Detection & Resolution, Reinforcement Learning, Self Separation, U-Space
Printed by:	Ipskamp
Front & Back:	M. Ribeiro

An electronic version of this dissertation is available at
`http://repository.tudelft.nl/`.

SUMMARY

Increasing delays and congestion reported in many aviation sectors indicate that the current centralised operational model is rapidly approaching saturation levels. Air Traffic Control (ATC) system is not expected to keep pace with the ever-increasing demand for air transportation. Its capacity is still limited by the available controllers, and the number of aircraft that each controller can manage. This system cannot be stretched any further under its current conditions. However, it is expected that the number of aircraft operating simultaneously will continue to increase. On top of this, new unmanned aviation operations promise traffic densities never seen before.

The expected future increase in traffic demand has redirected focus into automated tools and alternative approaches. This research has been primarily characterised by a change in the degree of centralisation, more specifically by exploring distributed options, where control is transferred from ground-based Air Traffic Controllers (ATCOs) to each individual aircraft. As each aircraft only takes into account its neighbouring aircraft when resolving conflicts, each distributed resolution system is expected to have only a fraction of the computational strain that a centralised system would have. Nevertheless, a distributed approach has its own challenges. A crucial disadvantage is the lack of global coordination from surrounding traffic, which can affect safety. Without knowledge of the movement of intruders, decentralised solutions cannot guarantee globally optimal solutions when more than two aircraft are involved.

Conflict resolution (CR) methods based on geometric solutions have proven to be very successful in achieving a high level of safety for one-to-one conflicts, where a set of rules can be defined which leads to implicitly coordinated optimal behaviour. However, at higher traffic densities, when individual conflict situations can no longer be considered isolated events, successive CR manoeuvres can lead to traffic patterns with a negative effect on the global safety. Knock-on effects of intruders avoiding each other may result in unforeseen trajectory changes. The latter increases uncertainty regarding intruders' future movements, decreasing the efficacy of conflict resolution manoeuvres.

The goal of this research is to improve upon aircraft self-separation efficacy at higher traffic densities, with an emphasis on employing airspace designs and approaches applicable to future unmanned operations. To do so, we look at a scenario with multi aircraft interacting as a multi-agent problem. Analysis and understanding of emergent behaviour in a multi-agent environment is often almost impossible to the human eye. However, reinforcement learning (RL) techniques are often capable of identifying emerging patterns through training in the environment. We translate successful applications of RL techniques in other areas (e.g., car mobility, lane changing, freeways) to aircraft operational scenarios to mitigate the negative effect on safety of emerging patterns resulting from multiple successive resolution manoeuvres.

The first part of this study focusses on dynamic and static obstacle avoidance for unmanned aviation in an urban environment. The available airspace is divided according to the layered airspace concept, as researched by the Metropolis project. Aircraft are limited to speed and altitude variation for conflict resolution, so as to avoid crossing the barriers of the surrounding urban infrastructure. It is shown that employing RL techniques can help decrease conflict rate and severity. First, an RL method implements velocity speed limits, enabling a more homogeneous traffic situation during layer tran-

sition phase. These limits increase the distance between the aircraft, reducing the total number of violations of minimum separation. Second, to improve layer change decision, two RL modules are employed: a decision-making module, which outputs lane change commands, and a control-execution module which controls the aircraft longitudinally and vertically to ensure a safe merging manoeuvre. Both modules, working independently and combined, reduce the total number of conflicts and losses of minimum separation when compared to manually defined baseline rules.

Additionally, airspace structure plays a positive role in reducing conflict rate and severity by directly affecting the likelihood of aircraft meeting in conflict. Often, structures are set assuming a uniform traffic distribution. However, in a real-world this is often not the case. An RL method is used to set the headings allowed per layer, in a layered airspace environment, in accordance with the expected traffic scenario. The output structures optimise the usage of the airspace by segmenting aircraft efficiently throughout the available airspace by taking into account their flight plan. The adapted structures lead to fewer conflicts and losses of minimum separation, and faster flights when compared to an uniform, fixed structure which assumes a uniform traffic scenario.

The second part of this thesis focused on how RL can be used to directly improve conflict resolution. Experiment results show that RL cannot yet outperform current CR geometric methods. These calculate geometric resolution manoeuvres which guarantee implicit coordination with minimum path deviation. This is a level of precision impossible to be re-enacted by a machine learning method. However, conflict resolution algorithms work based on man made pre-defined rules (e.g., pre-defined look-ahead time, pre-defined manoeuvres). RL can instead create a much larger set of rules, adapted to a multitude of different conflict situations. Moreover, RL methods can be used to improve the behaviour in situations for which researchers do not have a clear guideline or an optimal set of rules (e.g., return to the nominal path after conflict resolution, prioritisation of intruders or deconflicting manoeuvres).

Lastly, it is necessary to consider the practical applications of this research. The final objective is for the methods herein explored to be employed in the design of new concepts enabling future operations. Due to the empirical nature of the results, the conclusions drawn in this thesis are, to some degree, sensitive to the parameter settings of the simulated airspace. However, the same methods can be adapted to different environments. First, the detection and resolution algorithms employed are independent of the environment; the only limitation is the number of degrees of freedom that aircraft are allowed to use to avoid conflicts. Second, the reinforcement learning methods employed can be trained in most environments and will adapt to its characteristics.

The main limitations before applying these methods relate to validation under fairer representations of real-world operational environments. For example, higher uncertainty regarding intruders' position and non-ideal weather conditions must be tested. Bad weather, and strong winds in particular, can severely reduce aircraft manoeuvrability, and decrease the set of possible manoeuvres for conflict resolution, affecting the safety of the airspace. Moreover, several issues associated with the RL methods must be addressed. Namely, a higher degree of interpretation and explanation of their actions. Additionally, safeguards must be implemented against unsafe actions that a method may output when faced with a new situation for which it does not possess sufficient knowledge.

SAMENVATTING

De toenemende vertragingen en congestie die in veel luchtvaartsectoren worden gemeld, wijzen erop dat het huidige gecentraliseerde operationele model snel zijn limieten bereikt. Naar verwachting zal het huidige gecentraliseerde ATC-systeem geen gelijke pas kunnen houden met de steeds toenemende vraag naar luchtvervoer. De capaciteit ervan wordt nog steeds beperkt door het aantal beschikbare verkeersleiders en de hoeveelheid vliegtuigen die elke verkeersleider kan beheren. Binnen de huidige omstandigheden kan dit systeem niet verder worden opgerekt. Echter wordt verwacht dat het aantal vliegtuigen dat gelijktijdig in de lucht is, zal blijven toenemen. Bovendien benodigen nieuwe onbemande luchtvaartoperaties nooit eerder vertoonde verkeersdichtheden.

Door de verwachte toekomstige toename van de verkeersvraag is de aandacht verlegd naar geautomatiseerde instrumenten en alternatieve aanpakken. Dit onderzoek werd vooral gekenmerkt door een verandering in de mate van centralisatie, specifieker nog het verkennen van gedistribueerde opties, waarbij de controle wordt overgedragen van luchtverkeersleiders op de grond (ATCO's) naar ieder individueel vliegtuig. Aangezien elk vliegtuig bij het vermijden van conflicten alleen rekening houdt met de naburige vliegtuigen, zal naar verwachting elk gedistribueerd ontwijkingssysteem slechts een fractie van de computerdruk benodigen van een gecentraliseerd systeem. Desalniettemin heeft een gedistribueerde aanpak zijn eigen uitdagingen. Een fundamenteel nadeel is het gebrek aan globale coördinatie van het omringende verkeer, wat de veiligheid in het gedrang kan brengen. Zonder kennis van de bewegingen van indringers, kunnen gedecentraliseerde oplossingen geen globaal optimale oplossingen garanderen wanneer er meer dan twee vliegtuigen bij betrokken zijn.

Conflictoplossingsmodellen op basis van geometrische oplossingen zijn zeer succesvol gebleken in het bereiken van een hoog veiligheidsniveau voor één-op-één conflicten. Hierbij wordt een reeks regels gedefinieerd die leiden tot impliciet gecoördineerd optimaal gedrag. Bij hogere verkeersdichtheden, wanneer individuele conflictsituaties niet langer als geïsoleerde gebeurtenissen kunnen worden beschouwd, kunnen opeenvolgende CR-manoeuvres echter leiden tot verkeerspatronen met een negatief effect op de algemene veiligheid. 'Knock-on' effecten van indringers die elkaar ontwijken kunnen leiden tot onvoorziene trajectwijzigingen. Dit laatste verhoogt de onzekerheid over de toekomstige bewegingen van indringers, waardoor de doeltreffendheid van conflictoplossende manoeuvres afneemt.

Het doel van dit onderzoek is het verhogen van de efficiëntie van zelfseparatie van vliegtuigen bij hogere verkeersdichtheden, met de nadruk op het gebruik van luchtruimontwerpen en aanpakken die toepasbaar zijn op toekomstige onbemande operaties. Daartoe bekijken we een scenario met meerdere vliegtuigen die met elkaar interageren als een 'multi-agent' probleem. Vaak is analyse en begrip van vertoond gedrag in een 'multi-agent' omgeving voor het menselijk oog vrijwel onmogelijk. Technieken van reinforcement learning (RL) zijn echter vaak in staat om door training, opkomende patronen te identificeren. Wij benutten succesvolle toepassingen van RL-technieken op andere gebieden (bv. automobiliteit, verandering van rijbaan, snelwegen) naar operationele scenario's om de veiligheid te waarborgen van vliegtuigen bij patronen die ontstaan uit meerdere opeenvolgende ontwijkingsmanoeuvres.

Het eerste deel van dit onderzoek richt zich op dynamische en statische obstakelver-

mijding voor onbemande luchtvaartuigen in een stedelijke omgeving. Het beschikbare luchtruim is onderverdeeld volgens het gelaagde luchtruimconcept, zoals onderzocht in het Metropolis-project. Vliegtuigen worden beperkt in snelheid en hoogtevariatie bij conflictoplossingen, om te voorkomen dat ze de barrières van de omringende stedelijke infrastructuur overschrijden. Er werd aangetoond dat het gebruik van RL-technieken kan helpen om het aantal conflicten en de ernst ervan te verminderen. Eerst werden in een RL-model snelheidsbeperkingen ingevoerd, waardoor een homogenere verkeerssituatie tijdens overgangsfases tussen lagen mogelijk werd. Deze limieten vergroten de afstand tussen de vliegtuigen, waardoor het totale aantal schendingen van de minimale separatie afneemt. Ten tweede werden, om de beslissing over verandering van laag te verbeteren, twee RL-modules gebruikt: een besluitvormingsmodule, die opdrachten voor verandering van rijbaan geeft, en een uitvoeringsmodule die het vliegtuig in horizontale en verticale richting controleert om een veilige samenvoegingsmanoeuvre te garanderen. Beide modules, die onafhankelijk en gecombineerd werken, verminderden het totale aantal conflicten en schendingen van minimale separatie in vergelijking met handmatig gedefinieerde basisregels.

Bovendien speelt de luchtruimstructuur een positieve rol bij het verminderen van de hoeveelheid en de ernst van conflicten, omdat dat rechtstreeks van invloed is op de waarschijnlijkheid dat luchtvaartuigen elkaar kruisen. Bij de vaststelling van de structuren wordt vaak uitgegaan van een uniforme verkeersverdeling. In de praktijk is dit echter niet vaak het geval. Een RL-model werd gebruikt om in een gelaagde luchtruimomgeving de toegestane richtingen per laag vast te stellen in overeenstemming met het verwachte verkeersscenario. De outputstructuren optimaliseren het gebruik van het luchtruim door vliegtuigen efficiënt te segmenteren in het beschikbare luchtruim, terwijl het rekening houdt met hun vliegplan. De aangepaste structuren leiden tot minder conflicten en schendingen van de minimale separatie, en snellere vluchten in vergelijking met een uniforme, vaste structuur die uitgaat van een uniform verkeersscenario.

In het tweede deel van dit proefschrift werd nagegaan hoe RL kan worden gebruikt om conflicten rechtstreeks op te lossen. Experimentresultaten tonen aan dat RL de huidige geometrische CR-methoden nog niet kan overtreffen. Deze berekenen geometrische oplossingsmanoeuvres die impliciete coördinatie met minimale padafwijking garanderen. Dit is een precisieniveau dat onmogelijk kan worden nagebootst door een machine-learning methode. Conflictoplossingsalgoritmen werken echter op basis van vooraf door de mens gedefinieerde regels (bv. vooraf gedefinieerde 'look-ahead'-tijd, vooraf gedefinieerde manoeuvres). Daarentegen kan RL een veel grotere reeks regels creëren, aangepast aan verschillende conflictsituaties. Bovendien kunnen RL-methoden worden gebruikt om het gedrag te verbeteren in situaties waarvoor onderzoekers geen duidelijke richtlijn of optimale reeks regels hebben (bijvoorbeeld terugkeer naar het nominale pad na conflictoplossing, prioritering van indringers of deconflicterende manoeuvres).

Tenslotte moeten de praktische toepassingen van dit onderzoek worden bekeken. Het uiteindelijke doel is dat de hier onderzochte methoden worden gebruikt bij het ontwerpen van nieuwe concepten die toekomstige operaties mogelijk maken. Door de empirische aard van de resultaten zijn de conclusies die in dit proefschrift worden getrokken, tot op zekere hoogte gevoelig voor de parameterinstellingen van het gesimuleerde luchtruim. Dezelfde methoden kunnen echter aan verschillende omgevingen worden aangepast. Ten

eerste zijn de gebruikte detectie- en oplossingsalgoritmen onafhankelijk van de omgeving; de enige beperking is het aantal vrijheidsgraden dat vliegtuigen mogen gebruiken om conflicten te vermijden. Ten tweede kunnen de gebruikte reinforcement learning-methoden worden getraind in de meeste omgevingen en zullen zij zich aanpassen aan de kenmerken ervan.

De belangrijkste beperkingen voor de toepassing van deze methoden hebben betrekking op validatie onder eerlijker representaties van reële operationele omgevingen. Zo moet bijvoorbeeld de grotere onzekerheid over de positie van indringers en niet-ideale weersomstandigheden worden getest. Slecht weer, met name sterke wind, kan de manoeuvreerbaarheid van vliegtuigen ernstig beperken en de reeks mogelijke manoeuvres voor conflictoplossing verkleinen, hetgeen de veiligheid van het luchtruim in gevaar brengt. Bovendien moeten verschillende problemen in verband met de RL-modellen worden aangepakt. Namelijk een hogere mate van interpretatie en uitleg van hun acties. Voorts moeten beveiligingen worden ingebouwd om onveilige acties te voorkomen die een model kan uitvoeren als het geconfronteerd wordt met een situatie waarover het onvoldoende kennis beschikt.

CONTENTS

1

INTRODUCTION

Safety in manned aviation still relies on manual intervention by ground-based air traffic controllers. The capacity of the air traffic control system is limited by the available controllers, and the number of aircraft that each controller can manage. This system cannot be stretched any further under its current conditions. However, it is expected that the number of aircraft that operate simultaneously will continue to increase. In addition, new unmanned aircraft operations promise traffic densities never seen before.

As a solution, new autonomous methods capable of balancing aircraft safety and efficiency are being developed. Reinforcement learning, in particular, has received special attention due to good results in multi-agent decision making problems where agents must remain separated. However, its application in aviation is still tentative. This thesis explores how this method can be used within self-separation assurance for future operations.

This chapter presents an overview of the previous literature in this domain. Several open questions regarding the management of future traffic densities are used as the foundation for the main research objectives of this thesis. The structure of this dissertation is presented at the end.

1

1.1. FUTURE OPERATIONS IN AVIATION

The increasing delays and congestion reported in many areas indicate that the current centralised operational model is rapidly approaching saturation levels [1]. The centralised Air Traffic Control (ATC) system is not expected to keep pace with the ever-increasing demand for air transportation [2, 3]. This has inspired research into automated tools and alternative approaches since the early 1990s. Several large research programs have been formed along this theme (e.g., FREER [4], PHARE [5], the Mediterranean Free Flight project in Europe [6], and DAG/TM in the US [7]; more recently, there is the American NextGen programme [8] and SESAR in Europe [3]). This research has been primarily characterised by a change in the degree of centralisation, more specifically, by exploring distributed options, where control is transferred from ground-based Air Traffic Controllers (ATCOs) to each individual aircraft. Figure 1.1 shows the main direction of information transmission for both cases.

(a) Centralised system.   (b) Distributed system.
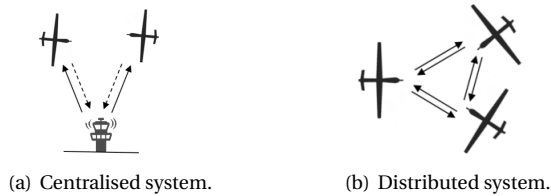
Figure 1.1: Difference between a centralised and a distributed system.

Previous research has shown that decentralised traffic separation can be expected to increase traffic capacity. In a centralised system, traffic flow is still limited to the maximum workload air traffic controllers (ATCOs) can handle within their sector [9]. Additionally, the number of sectors is limited by the number of ATCOs available. Delegating ATCOs to a monitoring position also allows for a re-evaluation of the airspace structure. Without the limitation of the number of ATCOs, sectors can now be replaced with a more complex structure, without being limited by the number of available personnel.

Moreover, although decentralisation was originally proposed to improve commercial manned aviation operations, the concept has become increasingly popular as a means to integrate Unmanned Aircraft Systems (UAS) into low altitude urban airspace. Many researchers and aviation authorities view the distribution of traffic separation tasks as a necessary step towards accommodating the high traffic volumes predicted for these new operations [3, 10]. Furthermore, both the Federal Aviation Administration (FAA) [11] and the International Civil Aviation Organisation (ICAO) [12] have ruled that an Unmanned Aerial System (UAS) must have Sense and Avoid capability to be allowed in civil airspace. Understandably, much of the current research into UAS has used or adapted methods previously found in manned aviation. These include the development of new distributed operational concepts [13], and new self-separation technologies [14].

Nevertheless, traffic separation in a decentralised manner entails its own new challenges, as each aircraft is only aware of its immediate surroundings, and only in control of its own movements. These limitations still require further research and understanding. The next subsection dwells on this topic further.

1.2. SELF-SEPARATION IN DECENTRALISED SYSTEMS

A distributed system reallocates the process of separation assurance from a centralised point to the individual aircraft. As each aircraft only takes into account its neighbouring aircraft when avoiding conflicts, each distributed resolution system is expected to have only a fraction of the computational strain that a centralised system would have. Nevertheless, a distributed approach has its own challenges. A crucial disadvantage is the lack of global coordination from surrounding traffic, which may affect safety. Without knowledge of the movement of intruders, decentralised solutions cannot guarantee globally optimal solutions when more than two aircraft are involved. Due to this, the efficacy of decentralisation in resolving multi-actor conflicts is often studied.

In particular, Free-Flight research [6] has focused on developing automated tactical algorithms for airborne conflict detection and resolution (CD&R). A conflict is a future prediction of a loss of minimum separation. CD&R consists of: (1) conflict detection (CD), the process of predicting future minimum separation violations by estimating how close the neighbouring aircraft will be to each other in the future; (2) conflict resolution (CR), which temporarily alters the current trajectory of the aircraft to avoid future detected separation violations.

Figure 1.2 represents a conflict; continuation of the current state of both aircraft will result in the aircraft getting closer than the pre-defined minimum separation distance. Figure 1.3 shows a loss of minimum separation (or intrusion). The desired minimum separation is defined as a circle around the aircraft. This area is designated as the aircraft's protected zone (PZ). It is considered that an aircraft is safely separated from other traffic or obstacles when these do not cross the PZ's barrier. The value of the PZ's radius may vary depending on the type of aircraft and operational environment.
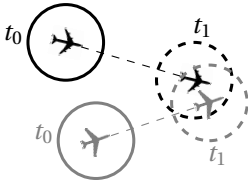
Figure 1.2: Conflict detected. Unless one or both aircraft alter their path, they will enter a minimum separation violation in the future.
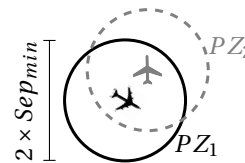
Figure 1.3: Loss of separation or intrusion. Sep_{min} is the minimum separation (i.e., the radius of the protected zone (PZ)).

In addition to CD&R, a limited number of studies have investigated the use of Conflict Prevention (CP) algorithms. These algorithms aim to improve safety by preventing aircraft from moving into new conflicts. Hoekstra [15], for example, focuses on mitigating conflict chain reactions, which greatly increase the number of conflicts. The effect of these chain reactions will be discussed in further chapters.

Guaranteeing self-separation in a high traffic density environment is the main objective of this thesis. With a high number of intruders, each aircraft will have to successively employ conflict resolution manoeuvres. These successive actions result in unpredictable traffic patterns that may lead to chain conflict reactions with neighbouring aircraft. As aircraft interact, an emergent behaviour surges, which cannot be easily predicted by the behaviour of a single aircraft. This problem is further defined in the following subsection.

1

1.3. REINFORCEMENT LEARNING APPROACH

Previous research has shown that, at high traffic densities, unwanted emergent behaviour from interactions between multiple aircraft working individually severely hinders conflict resolution. For one-to-one conflicts, a set of rules can be defined, which leads to implicitly coordinated optimal behaviour. However, as the number of aircraft increases, successive CR manoeuvres can lead to global patterns that cannot be predicted on the basis of these single rules or analytical methods. A high traffic operating scenario is essentially a multi-agent problem, with emergent behaviour and complexity arising as a result of agents interacting and co-evolving. Analysis and understanding of emergent behaviour in a multi-agent environment is often almost impossible to the human eye. However, reinforcement learning (RL) techniques are often capable of identifying emerging patterns through training in the environment. This thesis looks at the successful applications of these techniques in other areas (e.g., mobility of cars, lane change, highways [16–18]) and adapts them to aircraft operational scenarios. Since RL will be widely used in this thesis, it is important to briefly introduce this topic before applying it.

RL is, in a simplified way, the study of how an agent can interact with its environment to learn a policy which maximises the expected cumulative rewards for a task. The agent interacts with the environment in discrete timesteps. At each timestep, the agent receives the current state of the environment and performs an action based on which it receives a reward. The goal is for the agent to 'learn' which actions lead to receiving higher rewards. By using repetition, it can adapt to existing emergent behaviour and develop a large set of rules and weights from the knowledge of the environment captured during training. Unfortunately, RL also has its drawbacks. An RL model may not converge towards actions with 'optimal' rewards, or may take too long to learn to do so when it cannot properly connect the changes in the environment to the actions performed. This is particularly difficult in complex and rapidly changing environments. Thus, how RL is applied to help solve the problem of self-separation has to be carefully considered.

Recently, reinforcement learning has started to be used for conflict resolution purposes. In [19–21], RL was used to apply tactical speed and heading changes to the nominal trajectory to avoid aircraft getting too close to each other. However, these works show that RL techniques, although promising, do not achieve the efficacy of known conflict resolution methods. It is not certain that, at the current state of RL, it can successfully handle the uncertainty and complexity of a higher number of intruders in a multitude of different conflict situations. Translating the success of deep learning from single agent RL to a multi-agent environment continues to be a key challenge.

When applying RL to mitigate undesirable emergent patterns, several questions follow: which information should the RL model know?; which CR parameters should the RL model control? Additionally, two problems arise when using RL in cooperative multi-aircraft situations. First, with each action, the next state depends not only on the action performed by the ownship, but on the combination of that action with the actions simultaneously performed by the intruders. Second, from the point of view of each agent, the environment is non-stationary and, as training progresses, changes in a way that cannot be explained by the agent's behaviour alone. Current research shows that emergent behaviour and complexity arise mainly from agents interacting and co-evolving. This thesis explores these questions.

1.4. PROBLEMS WITH CD&R AT HIGH TRAFFIC DENSITIES

Operations with high traffic densities increase the likelihood of aircraft encountering multi-actor conflict situations. Aircraft must coordinate their own actions with the actions of the neighbouring aircraft to successfully avoid getting too close. In a pairwise conflict, conflict resolution methods, or rules, can be implemented so that aircraft work together to prevent a loss of minimum separation (i.e., implicit coordination). However, these rules alone cannot predict the traffic patterns that emerge from successive conflict resolution manoeuvres. These patterns lead to unpredictable trajectory changes, which not only result in aircraft having to cross the path of neighbouring aircraft to resolve pressing conflicts, but also make it harder to predict future losses of minimum separation and how to avoid them.

Moreover, with unmanned aviation in an urban environment, there is the additional challenge that any resolution manoeuvre must respect the boundaries of the surrounding urban infrastructure. This severely limits the number and magnitude of movements that an aircraft can adopt to resolve conflicts. To better approach this problem, we first divide it into four approachable sub-problems defined in the following subsections.

TRANSITIONING FROM MANNED TO UNMANNED AVIATION

The airspace is currently dominated by manned aviation, from aircraft types to traffic management approaches and regulations. With the introduction of new unmanned aviation operations, these tend to 'borrow' results of the self-separation research done for manned aviation. However, it is not yet clear if the same approach should be used for both cases, given their differences (e.g., performance disparity between the different types of aircraft, different self-separation margins).

Moreover CD&R methods can adopt different approaches. For example, many CD&R algorithms differ in how to propagate future trajectories, how to calculate the resolution manoeuvre, or even how far in advance the ownship defends from conflicts. More insight is needed into which CD&R characteristics better perform in tactical, high density traffic operations, especially in urban environments.

DYNAMIC AND STATIC OBSTACLE AVOIDANCE

To be used in unmanned operational environments, CD&R methods must either consider static obstacles, or at least limit conflict resolution manoeuvres in order for aircraft not to cross the barriers of the surrounding urban infrastructure. Not performing heading deviations to resolve conflicts guarantees that aircraft follow a pre-defined path, built around the urban infrastructure. However, such an approach strongly limits the number of possible ways to resolve conflicts. For example, (near-)head-on conflicts are (practically) impossible to resolve without heading deviations.

Furthermore, static obstacles often cause aircraft to have to make heading deviations to avoid direct collisions. Turns may lead to aircraft crossing traffic flows with other aircraft travelling in different directions, or even to speed heterogeneity caused by aircraft having to slow down to perform a turn with a small radius. The resulting large conflict angles or relative speeds are causal factors in the increase in complexity in air traffic operations.

1

AIRSPACE STRUCTURE

The structure of the airspace can play a positive role in reducing the severity of conflicts by directly affecting the likelihood of aircraft meeting in conflict. Conflict prevention may often be the best form of conflict resolution. The Metropolis project explored different types of distributed structures for a high-density urban airspace [13]. However, only static, relatively uniform traffic demand distributions were considered. In reality, traffic flows are often dynamic and can take up any distribution. Existing research on airspace design, towards optimising decentralised traffic flows, must be improved to adjust the airspace to the expected operational traffic scenario.

Furthermore, optimal airspace structuring is highly dependent on the (topological) characteristics of the traffic demand. Manned aviation employs fixed routes planned in advance, and thus has fixed sectors as a function of these expected routes. However, unmanned aviation is expected to include missions with unpredictable and variable routes, such as food and package delivery [22]. The latter entails a more dynamic, complex structuring based on the traffic needs at each instant.

MULTI-ACTOR CONFLICT RESOLUTION

Conflict resolution methods based on geometric solutions have proven to be very successful in achieving a high level of safety for one-to-one conflicts, where a set of rules can be defined, leading to implicitly coordinated optimal behaviour. However, at higher traffic densities, when individual conflict situations can no longer be considered isolated events, successive CR manoeuvres can lead to traffic patterns with a negative effect on the global safety. The knock-on effects of intruders avoiding each other may eventually result in unforeseen trajectory changes, deeming a resolution impossible within the available amount of time before a loss of separation. Thus, conflict resolution efficacy in multi-actor conflicts must be improved.

1.5. RESEARCH OBJECTIVES

This research aims to address the four open problems previously discussed for self-separation in high traffic density operational environments. More specifically, the primary objective of this thesis is to:

Primary Research Objective
Investigate whether reinforcement learning applications can improve aircraft self-separation efficacy at higher traffic densities, with an emphasis on employing airspace designs and approaches applicable to future unmanned operations.

First, to be able to improve upon the current performance of self-separation methods, these must be evaluated and understood. The next subsection provides more information on Chapter 2, which is intended as an analysis of the current state-of-the-art. Next, we define the research activities and associated research questions created to meet the primary research objective.

1

BACKGROUND CHAPTER

Chapter 2 of this thesis analyses the performance of current CD&R methods in manned and unmanned aviation, with the objective of better understanding which characteristics/approaches lead to a reduction of intrusions. As such, this chapter may be seen as an overview of the current state-of-the-art in conflict avoidance/resolution methods.

The last comprehensive review of conflict detection and resolution methods was published in 2000 by Kuchar and Yang [23]. Although it is more than 20 years old, this is still often cited as the main reference. However, since 2000, many new CD&R concepts have been proposed, including an entirely new branch of CD&R methods directed specifically at unmanned aviation. A possible taxonomy for the latter was first explored by Jenie [24]. However, a single, current overview of CD&R methods for both manned and unmanned applications is currently lacking.

In Chapter 2, a taxonomy is presented that characterises CD&R algorithms in terms of their approach to avoidance planning, surveillance, control, trajectory propagation, predictability assumption, resolution manoeuvre, multi-actor conflict resolution, considered obstacle types, optimisation, and method category. More than a hundred CR methods are evaluated based on this taxonomy. Such provides a base for the review of the current state of both manned and unmanned CD&R algorithms, and helps identify which characteristics lead to better efficacy at high traffic densities. The performance of four main CR algorithms is directly compared within the same simulation/traffic scenarios. Fast-time simulations are performed on an open-source airspace simulation platform.

RESEARCH ACTIVITY 1: DYNAMIC AND STATIC OBSTACLE AVOIDANCE

CD&R methods need to be adapted to unmanned aviation in an urban environment. Any resolution manoeuvre must respect the borders of the surrounding urban infrastructure. This limits the magnitude of heading changes that can be performed to resolve conflicts. Thus, the following research question is created:

Research Question on Dynamic and Static Obstacle Avoidance (Chapter 3)

RQ 1: How to reduce the conflict rate and severity in a constrained urban environment (especially when considering that aircraft cannot perform heading variations to aid in conflict resolution)?

Chapter 3 focuses on how conflict resolution can be performed in an urban environment. The available airspace is divided according to the layered airspace concept, as researched by the Metropolis project [13], where traffic is divided into different vertical layers according to their current heading. The emphasis is placed on speed variation with a velocity obstacle-based conflict resolution method. Intent information is added to trajectory propagation in order to mitigate crossing conflicts (i.e., conflicts resulting from changes in direction). Finally, a reinforcement learning agent is used to implement variable speed limits towards creating a more homogeneous traffic situation between cruising and climbing/descending aircraft.

The results of Chapter 3 show the need for additional focus on merging conflicts. We define merging conflicts as conflict situations resulting from an aircraft joining a traffic

1

flow in a different traffic layer (similarly to a road vehicle in a highway joining a different lane). Conflicts between cruising and climbing/descending aircraft are especially difficult to resolve. First, having simultaneous vertical and horizontal conflicts severely increases the level of complexity of conflict resolution. Second, when aircraft enter a traffic flow, they can potentially force a conflict chain reaction in which the follower aircraft have to adjust their speed to avoid a collision. Thus, the following research question arises:

> **Research Question on Dynamic and Static Obstacle Avoidance (Chapter 4)**
>
> **RQ 2:** How to reduce the conflict rate and severity during vertical merging manoeuvres?

Chapter 4 looks at reducing the impact of vertical transitions within an aviation environment. Two reinforcement learning methods are tested: a decision-making module and a control-execution module. The former issues a lane change command based on the planned route. The latter performs operational control to coordinate the longitude and vertical movement of the aircraft for a safe merging manoeuvre. The performance of these modules is compared to the use of manually defined navigation rules.

RESEARCH ACTIVITY 2: AIRSPACE STRUCTURE

This research aims to find the optimal structure for the expected traffic scenario. Previous airspace structures have assumed that traffic adopts a uniform heading range distribution. However, this is rarely the case in the real-world. When the structure of the airspace does not align with the current traffic scenario, aircraft will not be equally distributed across the available airspace. Thus, it raises the following research question:

> **Research Question on Airspace Structure (Chapter 5)**
>
> **RQ 3:** How to optimise the airspace structure based on the operational traffic scenarios?

First, it should be noted that the terms airspace *structure* and *design* are used interchangeably in this thesis. Both terms refer to procedural mechanisms for the separation and organisation of traffic. To address RQ 3, in Chapter 5, a reinforcement learning agent decides on the best airspace structure based on the traffic scenario. Multiple traffic demand scenarios are simulated for this activity. Subsequently, the effect of the airspace structure on the conflict rate and severity can be inferred.

RESEARCH ACTIVITY 3: CONFLICT RESOLUTION WITH REINF. LEARNING

This research activity will focus on the potential of using reinforcement learning directly for conflict resolution. At the extreme densities envisioned for such drone applications, performance is hindered by the unpredictable emergent behaviour of interacting traffic. Reinforcement learning can potentially be used to mitigate the negative effect of these emergent interactions. The following research question is then created:

1

> **Research Question on Conflict Resolution with Reinforcement Learning (Chapters 6 & 7)**
>
> **RQ 4:** Can reinforcement learning surpass or improve the efficacy of current analytical conflict resolution methods, specifically in multi-actor conflict situations?

Before application into the real-world, the potential of reinforcement learning as a tool needs to be further researched and its limitations understood. On the one hand, reinforcement learning can potentially identify trends and patterns for multi-conflict resolutions where human observation cannot. On the other hand, RL also has its drawbacks, such as non-convergence, high dependence on initial conditions, and long training times. Chapters 6 and 7 explore how to better approach reinforcement learning as a conflict resolution tool, and how its 'learnt' behaviour can be used to complement CD&R methods, thus improving conflict detection and resolution at high traffic densities. Multiple experiments are performed with different degrees of control over the aircraft's movements. The results help identify in which use cases reinforcement learning is optimal.

Finally, Chapter 8 is intended as an overview of the limitations of reinforcement learning, exploring non-successful applications which also contribute to a complete comprehension of this tool.

1.6. RESEARCH TERMINOLOGY AND SCOPE

Throughout the following chapters, it is assumed that several common-used concepts are understood by the reader and do not require further explanation. For reference, these concepts and their scope are formulated in the following paragraphs:

- **Aircraft**: the term aircraft is used interchangeably to refer to both manned and unmanned aviation. Unless one or the other is specifically mentioned, the reader may assume that the current topic applies to both.

- **Conflict Detection**: a conflict occurs when the horizontal and vertical distances between two aircraft are expected to be less than the minimum separation distance within a predetermined 'look-ahead' time (see Figure 1.2). Conflicts are thus indications of expected future violations of minimum separation.

- **Conflict Resolution**: once a conflict is detected, a conflict resolution algorithm is used to modify the aircraft's route to avoid the future loss(es) of separation. In each chapter, the conflict resolution algorithm used in the experiments is defined.

- **Losses of (minimum) separation (LoSs) or intrusion**: occurs when two aircraft are closer to each other than the pre-defined minimum separation distance, as displayed in Figure 1.3. This is the paramount safety factor and should be avoided.

- **Protected zone (PZ)**: the protected zone is a flat, three-dimensional disc around each aircraft, that should remain clear of other traffic. The value of the minimum safe separation may depend on the density of air traffic and the region of the airspace. However, for manned aviation, most CD&R studies use ICAO's definition of 5NM horizontal separation and 1000 ft vertical separation. For unmanned avia-

1

tion, there are no established separation distance standards yet. The values used in each experiment will be specified throughout this thesis.

- **Reinforcement Learning (RL)**: a machine learning based training method where the agent performs actions in an environment and, by trial and error, learns which actions result in maximising a cumulative reward.

- **Simulation Platform**: this thesis employs the open-source, multi-agent ATC simulation tool Bluesky [25]. All implementation code used and built throughout this work is available online.

- **Unmanned Aircraft System (UAS)**: the term Unmanned Aircraft System, abbreviated as UAS, refers to the definition of an aircraft and associated elements which are operated without a pilot on board. Well-known UAS methods are simulated in the experiments performed throughout this thesis.

1.7. THESIS OUTLINE

This thesis presents an answer to the previously introduced research questions (RQ). An overview of the relationships between the chapters and the research questions is shown in Figure 1.4. Chapter 2 is meant as a background chapter introducing the current state-of-the-art of conflict detection and resolution methods. Then, the thesis is divided into two main parts. The first includes Chapters 3 to 5, and is directed at conflict detection and resolution for unmanned aviation in a constrained environment. The second part, Chapters 6 to 8, focuses mainly on how reinforcement learning can be used to improve conflict resolution, and its limitations as a tool.

The following paragraphs present an outline of all chapters of the thesis:

Chapter 1: **Introduction**: introduces the research questions and topics to be discussed throughout this thesis.

Chapter 2: **Review of Conflict Resolution Methods for Manned and Unmanned Aviation**: an overview of current CD&R methods for both manned and unmanned applications. A single taxonomy is presented, thus creating a means of comparison between all methods.

Chapter 3: **Velocity Obstacle Based Conflict Resolution in Urban Environment with Variable Speed Limit**: travelling rules benefitting tactical conflict resolution are implemented in an urban airspace. These include vertical segmentation of all traffic, and the usage of reinforcement learning techniques to implement variable speed limits towards creating a more homogeneous traffic situation.

Chapter 4: **Using Reinforcement Learning in a Layered Airspace to Improve Layer Change Decision**: two reinforcement learning modules are used to improve lane change behaviour. The first is responsible for outputting lane change commands. The second receives these commands and controls the longitudinal and vertical movements of the aircraft towards a safe merging manoeuvre.

Chapter 5: **Using Reinforcement Learning to Improve Airspace Structuring in an Urban Environment**: a reinforcement learning model is used to determine the optimal heading range distribution per layer, within a layered airspace scenario, according to the operational traffic scenario.

Chapter 6: **Distributed Conflict Resolution With Reinforcement Learning**: a reinforcement learning model is directly responsible for distributed resolving multi-actor conflict situations. The performance levels obtained are directly compared with current geometric CD&R methods.

Chapter 7: **Improving Algorithm Conflict Resolution Manoeuvres With Reinforcement Learning**: a reinforcement learning model decides on the value used for the calculation of conflict resolution manoeuvres by a geometric conflict resolution algorithm. The performance levels obtained are directly compared with calculating resolution manoeuvres with predefined, commonly used values.

Chapter 8: **On the limitations of Reinforcement Learning in Aviation**: an overview of the limitations found with the training and testing of reinforcement learning modules in the resolution of conflicts with aircraft. Recommendations are made on how reinforcement learning should be applied towards more effective results.

Chapter 9: **Discussion and Recommendations**: summarises all chapters into an overview of the results. This chapter also provides some recommendations for further research, especially towards enabling high traffic densities into the airspace.

Chapter 10: **Conclusions**: provides a concise summary of the main conclusions of this thesis.

1.8. GUIDE TO THE READER

Chapters 2 to 7 of this dissertation are based on publications in journals that were written independently and, therefore, can be read separately. Each chapter is preceded by an introductory paragraph explaining how the chapter is related to the overall research. These preamble paragraphs also provide the publication history of each chapter, and mention sections contained within that are repeated from previous chapters. A list of publications of the research in this dissertation, both conference and journal articles, can be found after the last chapter.

1
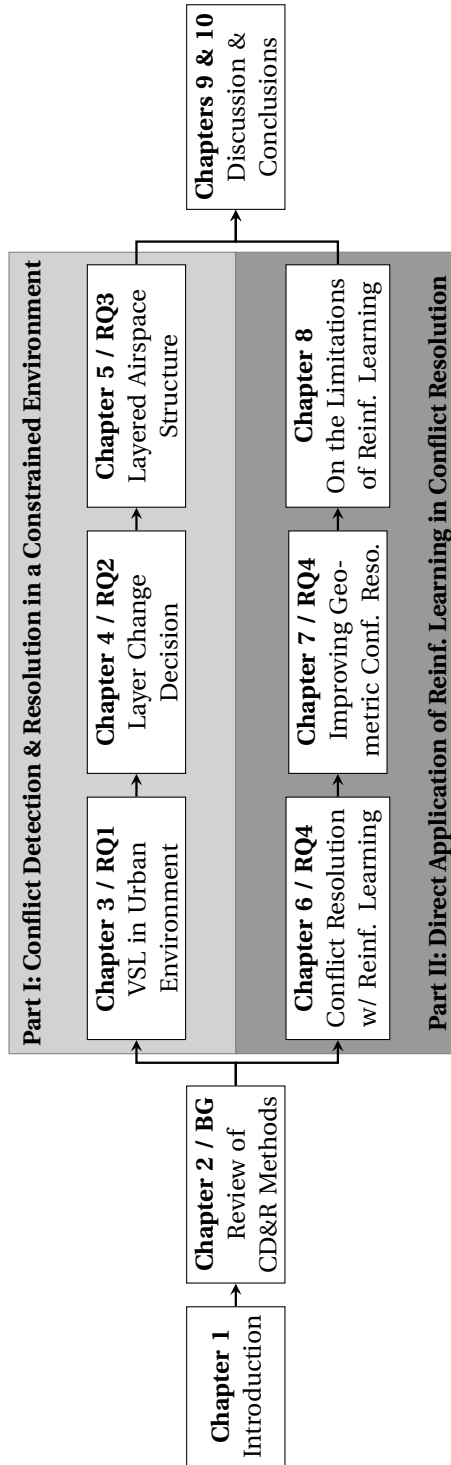
Figure 1.4: Structure of the chapters in this dissertation and their correlation with the research questions. Chapter 2 offers a state-of-the-art review of current conflict detection and resolution methods. Then, the thesis is divided into two main parts. The first, which spans from Chapter 3 to 5, includes work investigating conflict resolution and detection of static and dynamic obstacles in a constrained environment. The second, which includes chapters 6 to 8, focuses on the use of reinforcement learning techniques directly for conflict resolution in aviation.

2

REVIEW OF CONFLICT RESOLUTION METHODS FOR MANNED AND UNMANNED AVIATION

The last overview of conflict detection and resolution methods was published in 2000. Although more than 20 years old, this is still often cited as the main reference. However, since 2000, many new concepts have been proposed, including an entirely new branch of methods directed specifically at unmanned aviation. A single, current overview of conflict detection and resolution methods is missing for both manned and unmanned applications. The present chapter covers this gap.

Section 2.2 introduces a taxonomy for both manned and unmanned systems that categorises algorithms in terms of their approach to avoidance planning, surveillance, control, trajectory propagation, predictability assumption, resolution manoeuvre, multi-actor conflict resolution, obstacle types, optimisation, and method category. This will serve as a knowledge base for the rest of the thesis.

Finally, Section 2.3 provides a direct comparison of the performance of the main identified conflict resolution methods within the same traffic scenarios. The chapter ends with conclusions and suggestions for future improvement of conflict detection and resolution methods.

ABSTRACT

Research into urban aerial mobility, as well as the continued growth of global air transportation, has renewed interest in conflict detection and resolution methods. With the new applications of drones, and the implications of a profoundly different urban airspace, new demands are placed on such algorithms, further spurring new research. This chapter presents a review of current CR methods for manned and unmanned aviation. It presents a taxonomy that categorises algorithms in terms of their approach to avoidance planning, surveillance, control, trajectory propagation, predictability assumption, resolution manoeuvre, multi-actor conflict resolution, obstacle types, optimisation, and method category. More than a hundred CR methods are considered, showing how most work in a distributed, tactical framework. To enable a reliable comparison between methods, this chapter argues that an open and ideally common simulation platform, common test scenarios, and common metrics are required. This chapter presents an overview of four CR algorithms, each representing a commonly used CR algorithm category. Both manned and unmanned scenarios are tested through fast-time simulations on an open-source airspace simulation platform.

2.1. INTRODUCTION

The continued growth of aviation has been considered a threat to the current approach to air traffic control for decades, inspiring research into automated tools and alternative approaches since the early 1990s. As a result, several large research programs have been formed along this theme, such as FREER [4], PHARE [5], and the Mediterranean Free Flight [6] project in Europe, and DAG/TM [7] in the US. More recently, there are the American NextGen programme [8] and SESAR [3] in Europe. This research has been primarily characterised by the proposed degree of centralisation (delegation to the flight deck or maintaining centralisation), and along the dimension from tactical separation to strategic (re)planning. An extensive review of methods by Kuchar and Yang [23], published in 2000, is still often cited as an overview of Conflict Detection and Resolution (CD&R) methods.

In recent years, the prospect of a wide range of drone operations and the application of different aerial vehicles in an urban setting has renewed interest in CD&R research. However, there are several aspects that set these applications apart from the concepts considered in previous research. The capabilities of new platforms such as drones are different, and operating in an urban environment introduces new constraints (such as obstacles and hyperlocal weather) that did not need to be considered before. In addition, should the most ambitious concepts, such as drone-based package delivery and personal aerial mobility, become a reality, these applications will face traffic densities that are well beyond anything considered for manned aviation. Already, the Federal Aviation Administration (FAA) has ruled that an Unmanned Aerial Vehicle (UAV) must have Sense and Avoid capability in order to be allowed in the civil airspace [11]. Furthermore, the International Civil Aviation Organisation (ICAO) requires UAV CD&R methods to be capable of detection and resolution in both static and non-static environments. Only after meeting this requirement, will civil-UAVs be allowed to fly beyond the operator's visual line-of-sight [26].

Following these developments, many new CD&R concepts have been proposed since Kuchar and Yang's review study [23], including an entirely new branch of CD&R methods directed specifically at unmanned aviation. A possible taxonomy for the latter was first explored by Jenie [24]. To include these new methods, and to incorporate the demands that are placed on CD&R algorithms by the new application areas, this chapter aims to present a current overview of CD&R methods for both manned and unmanned applications. It will evaluate both manned and unmanned CD&R methods jointly in one single taxonomy, where the methods are categorised in terms of their approach to avoidance planning, surveillance, control, trajectory propagation, predictability assumption, resolution smanoeuvre, multi-actor conflicts, obstacle types considered, optimisation, and method category. The goal is for this framework to be used when developing new methods, or when identifying the most suitable method for a specific situation. As a result, this study can be considered an extension of the work performed by Kuchar and Yang [23] and Jenie [24], by providing a more complete analysis of CR methods that combines both manned and unmanned aviation.

In addition, this chapter provides a direct overview of the performance of the main identified CR method categories. Many publications related to new CR methods include an evaluation of the proposed method. However, comparison between such studies based on their individual results is often impossible, due to the differences in approach taken in the evaluations. Studies that present a comparison of multiple CR methods under the same conditions do not yet exist. Such evaluations are essential for a fair comparison between methods, as performance is highly dependent on factors such as the simulation platform, scenarios, and metrics used. To foster repeatable evaluations and fair comparisons, publicly available simulation tools, open data, and common scenarios and metrics should be used. Therefore, this study uses the open-source, multi-agent Air Traffic Control (ATC) simulation tool BlueSky [25]. The obtained experimental results are used to identify the differences in performance between manned and unmanned environments, as well as which CR methods are more efficient in the uprising unmanned aviation world.

2.2. TAXONOMY FOR DETECTION & RESOLUTION METHODS

Conflict resolution methods can be evaluated by a combination of several factors that define the airspace environment. In this review, we evaluate methods according to the following ten characteristics: the timescale on which avoidance planning takes place, the type of surveillance, whether control is centralised or distributed, trajectory propagation, predictability assumption, the manoeuvre employed for resolution, approach to multi-actor (>2) conflicts, obstacle types, optimisation objective, and method category. These categories are divided between detection and resolution as per Tables 2.1 and 2.2, respectively. For each category, the possible variations are presented below. More detail is provided in the following subsections.

2

Table 2.1: Taxonomy of conflict detection categories.

Conflict Detection Categories		
Surveillance	Trajectory Propagation	Predictability Assumption
Centralised Dependent		Nominal
Distributed Dependent	State-Based	Probabilistic
Independent	Intent-Based	Worst-Case

Table 2.2: Taxonomy of conflict resolution categories.

Conflict Resolution Categories		
Control	Method Categories	Multi-Actor Conflict Resolution
Centralised	Exact	Sequential
	Heuristic	Concurrent
Distributed	Prescribed	Pairwise Sequential
	Reactive	Pairwise Summed
	Explicitly Negotiated	Joint Solution

Table 2.2: *Cont.*

Applicable For All Conflict Resolution Categories			
Avoidance Planning	Resolution Manoeuvre	Obstacle Types	Optimisation
Strategic	Heading	Static	Flight Path
Tactical	Speed	Dynamic	Flight Time
Escape	Vertical	All	Fuel/Energy Consumption
	Flight Plan		

2.2.1. SURVEILLANCE

Aircraft surveillance can be defined in terms of whether the aircraft is dependent on external systems, or on its own on-board systems (i.e., independent). Within the former, an additional distinction can be made based on the origin of the data: a centralised system receives data from a common station, whereas a distributed system processes information from the surrounding traffic.

For centralised dependent surveillance (Figure 2.1(a)), aircraft are equipped with transponders capable of responding to ground interrogation. Ground sensors determine the 2D position of the aircraft, and altitude is provided by the aircraft. In manned aviation, this is done by ATC, and aircraft are expected to cooperate by broadcasting their altitude and identity. Distributed dependent surveillance (Figure 2.1(b)) uses the Automatic Dependent Surveillance-Broadcast (ADS-B) system; aircraft broadcast their position, altitude, identity, and other parameters by means of a data link, without any intervention from the ground systems.

Independent surveillance (Figure 2.1(c)) is commonly referred to as Sense and Avoid and uses on-board, non-cooperative systems/sensors. As unmanned aviation does not have a standard broadcast system, it often resorts to this type of surveillance with on-board sensors that detect both static and dynamic obstacles. This system is not employed in manned aviation, as aircraft are expected to cooperate through the ADS-B system.
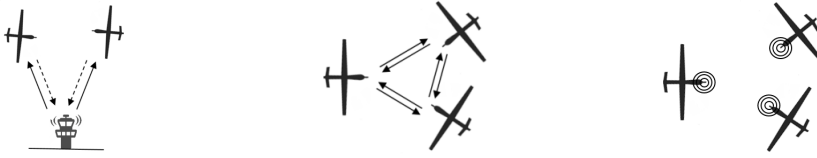
(a) Centralised dependent surveillance.

(b) Distributed dependent surveillance.

(c) Independent surveillance.

Figure 2.1: Different types of surveillance.

2.2.2. TRAJECTORY PROPAGATION

Future trajectories of aircraft can be considered based on their current state (i.e., state-based) or their future intent (i.e., intent-based). The former assumes a straight line as a continuation from the current state. The latter assumes turns and changes in heading and speed, based on the future waypoints of the aircraft.

State-based methods assume a straight-line projection of the aircraft's current position and velocity vector. This projection is simpler and faster computationally, since intent requires data transmission and heavier computational processing. Yang [27] has shown that reducing a non-linear trajectory to a series of straight lines trajectories allows for accurate computation of conflict states at speeds feasible in real-time complex scenarios. However, when future trajectory changes of all involved aircraft are not taken into account false alarms may occur, and future losses of separation resulting from changes in trajectory may be overlooked.

Intent informed can be simulated as a series of straight leg segments. Research conducted in the past for singular cases identified the potential of using intent. Multiple works [27–30] have used waypoint information to improve the prediction of a single intruder's trajectory. However, when conflict resolution is implemented, as aircraft diverge from their originally intended trajectory to avoid intrusions, new false alarms are also introduced. In manned aviation, distributed sharing of future trajectory change points (TCPs) can be done through ADS-B. For unmanned aviation, there is still no research on how this could be performed.

The use of both state and intent information in high traffic densities was investigated for civil aviation [31, 32], improving overall safety. The previous works also showed that adopting rules disallowing pilots from turning into a conflict, prevents intrusions resulting from sudden aircraft manoeuvres nearby. This can help mitigate the need for intent information.

2.2.3. PREDICTABILITY ASSUMPTION

A conflict is found once it is identified that two aircraft will be closer than the minimum required separation at a future point in time. This process requires an estimate of the future positions of all aircraft, and differs on whether uncertainties are added to the trajectory propagation. Uncertainties often arise in the form of uncoordinated behaviour of other traffic, and unknown wind or state variation. A nominal assumption (Figure 2.2(a)) does not consider uncertainties. A worst-case assumption (Figure 2.2(b)) considers all possible trajectory changes resulting from uncertainties. However, this is impractical in

a real environment, as its complexity results in heavy computation. Instead, a middle term, a probabilistic assumption (Figure 2.2(c)) is more often employed. In this case, the likelihood of each possible trajectory change is taken into account based on the current position, maximum turn, and climb rates. Whether to act, and how to act, is decided on the basis of the most likely trajectories.

The nominal assumption is often used in favour of simplicity and good computational performance. It is mostly used with shorter look ahead times (i.e., a few minutes), and can be quite accurate in an environment where aircraft have a steady behaviour. However, accuracy is expected to decrease as the model looks further into the future, as multiple small unexpected changes could have accumulated into a significant change in the trajectory. Therefore, alarms predicted far into the future are more likely to be unreliable.

Incorporating uncertainties may improve accuracy. The more potential trajectories considered, the more likely it is that one will resemble the real observed position into the future. However, this is at the cost of more false positive conflicts which are detected in the other trajectories that the aircraft could have taken. Adding more future states of neighbouring aircraft also reduces manoeuvring space. The further you look ahead, the larger the uncertainty space is and the smaller the manoeuvring is expected to be, which reduces traffic mobility. It may even reach a situation where no conflict resolution manoeuvre is found, as there is no manoeuvre which avoids all conflicts. A probabilistic assumption provides a solution; fewer trajectories are accounted for depending on their likelihood. This likelihood threshold may be decided based on the number of alarms the model can process within a limited amount of time.

(a) Nominal assumption.   (b) Worst-case assumption.   (c) Probabilistic assumption.
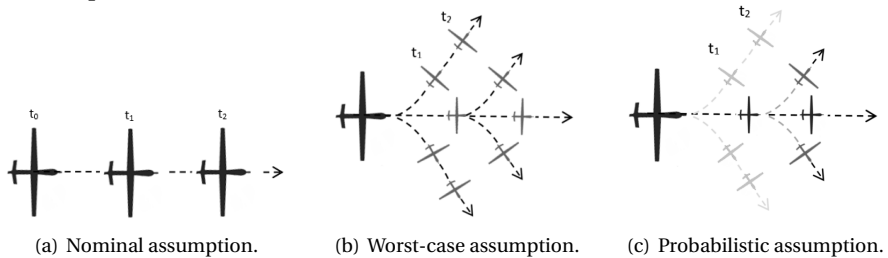
Figure 2.2: Different types of predictability assumption methods.

2.2.4. Control

Separation management, or control, may be centralised when decisions regarding future trajectory and conflict resolution are computed in a centralised location for multiple aircraft, or distributed when each aircraft is responsible for its own conflict resolution. Both approaches rely on a communication network to broadcast information such as intent, trajectories, and priorities.

A centralised system is capable of providing a global solution to complex multiple-actor problems. Uncertainty is reduced as each aircraft follows the solution defined by the centralised agent. Centralised methods typically work towards optimising trajectories; finding non-intersecting trajectories will guarantee separation. These centralised approaches are often computationally heavy, as a result of having to consider several possible manoeuvres for a number of aircraft, and therefore may not be suitable for real-time

implementation when this number increases considerably [33]. Hypothetically, when all information is known (e.g., the traffic situation, flight-specific optimisation preferences) and there is sufficient processing power, a centralised approach will lead to the most optimal solution. As all trajectories are known, these can be optimised for all involved aircraft. However, in practise, demands on the availability of information and the speed of information transfer must be taken into account. The availability of optimisation-related information is often limited by the willingness of airlines to share it. The prediction horizon tends to be large due to the time it takes to generate and communicate a global solution. Computational intensity increases with the traffic density; thus, there is a limit to the number of aircraft a centralised approach can operate simultaneously. Furthermore, a single processing point is also a single point of failure, resulting in a central failure mode with global consequences, which is absent in distributed systems.

In manned aviation, ATC is the centralised point responsible for guaranteeing the safety of all traffic. Air traffic controllers maintain a minimum separation between all aircraft in their airspace sector. Naturally, the traffic density allowed in the sector is thus limited by the maximum number of aircraft that controllers are capable of operating simultaneously. One objective of CD&R research is to reduce the constraint on ATC, whether by creating another centralised point capable of computing optimal trajectories for all involved aircraft without human aid, or distributed systems to be introduced into the on-board systems of each aircraft. In particular, the rise of unmanned aviation applications, where the number of aircraft involved is expected to greatly exceed the number currently operated by ATC [22], has led to the exploration of distributed approaches.

A distributed system reallocates the process of separation assurance from a centralised point to the individual aircraft. As each aircraft only takes into account its neighbouring aircraft when resolving conflicts, each distributed resolution system is expected to have only a fraction of the computational strain a centralised system would have. Nonetheless, the speed at which an aircraft can make a decision is still limited by the speed at which information from surrounding traffic is received and processed. A crucial disadvantage of a distributed system is the lack of global coordination from surrounding traffic which may impair safety. Without knowledge of the movement of intruders, decentralised solutions cannot guarantee globally optimal solutions when more than two aircraft are involved. Because of this, the efficacy of decentralisation in resolving conflicts is often studied and compared to that of centralised systems. Bilimoria [33] showed that a distributed resolution strategy can successfully solve complex multiple aircraft problems in real time. Durand [34] tested this with a no-speed variation scenario where only a centralised system was able to find a solution. Finally, the Free Flight concept [2, 6, 15] also illustrates that, when aircraft are fully responsible for their own separation from other traffic, they are free to decide upon their optimal route ('direct routing'), versus following the route received from a centralised point for safety. Studies for this project concluded that, once ADS-B technology is developed to a higher reliability and performance, a distributed conflict resolution system can safely guarantee airborne separation.

2.2.5. METHOD CATEGORIES

This review defines five main categories that can be used as the main classification for almost all currently existing methods. Two main categories are identified within research

for centralised approaches: the exact, and heuristic categories. Regarding distributed approaches, we identify three main categories: prescribed, reactive, and explicitly negotiated. These categories classify methods according to how resolution manoeuvres/trajectories are identified in environments with multiple aircraft, where all involved aircraft are expected to perform conflict resolution and modify their path in accordance.

In a centralised approach, a single agent is responsible for deciding the resolution path of all involved aircraft, thus it is known how aircraft will move in the future. During optimisation of an aircraft's trajectory towards separation, it is assumed that intruders will follow the path set by this agent. The selection of trajectories/resolution manoeuvres can be optimised towards a preference policy, a certain cost, or in other words to minimise a penalty function. The trajectory with the lowest cost is chosen from a set of limited possibilities. A preference can be made for either performance (e.g., lower fuel/energy consumption, flight path, time optimisation) or safety. It may even be considered that crossing the protected zone of another aircraft, over a short period of time, is better than increasing the flight path or adopting a significant change in speed. Methods may be classified on whether they are guaranteed to find the global optimum, i.e., exact algorithms, or heuristic algorithms which attempt to yield a good, but not necessarily global optimum solution. A Mixed Integer Linear Programming (MILP) approach is commonly used to find the global optimum [35]. However, an exact algorithm requires a long computing time, making it usually impractical for real-life applications [36]. Thus, heuristic algorithms, although not ensuring optimality, are often employed to shorten execution times. Commonly used heuristic approaches are Variable Neighbourhood Search (VNS) [37], Ant Colony Optimisation [38], and Evolutionary Algorithms [39, 40].

In both prescribed and reactive categories, coordination between aircraft is implicit. Traffic either reacts in accordance with a pre-defined set of rules (i.e., prescribed) or a common manoeuvre strategy in response to the conflict geometry (i.e., reactive). Prescribed is mainly achieved by application of the Right-of-Way (RoW) [41] rules. In short, these define that traffic from the left must give way, overtaking aircraft manoeuvre to the right, and head-on conflicts are resolved with both aircraft turning to the right. However, Balasooriyan [42] demonstrated that applying these rules results in a higher number of losses of separation and conflicts than employing other rule sets where both aircraft are expected to initiate a trajectory change to avoid conflicts. When both aircraft adopt a deconflicting route, the time in conflict decreases as both aircraft are moving away from each other. Reactive methods 'react' to the position of the intruders; resolution manoeuvres are a direct result of the conflict geometry. A common example is to use the 'shortest-way-out' principle, which ensures implicit coordination in one-to-one conflicts, as single conflicts are always geometrically symmetrical [2, 43]. It should be noted that the 'shortest-way-out' and the RoW coordination define rules for conflict pairs. As the minimum separation distance represents the distance between two aircraft, multi-actor conflicts are simultaneous occurrences of two-aircraft conflicts. When implementing a coordination rule per pairwise conflict, it may be that, given the geometry of the conflict, an aircraft receives contradictory solutions to solve its multiple pairwise conflicts. For example, when resolving pairwise conflicts sequentially, the resolution manoeuvre to the closest conflict can aggravate the next pairwise conflict or even create secondary conflicts with other aircraft. Such prompts the study and verification of implicit rules

among different multi-actor conflict geometries. Research aims to resolve this problem by developing better ways of implicit coordination, combination of resolution manoeuvres, and/or prioritisation [44].

Resolution methods in the explicitly negotiated category resolve conflicts based on explicit communication between aircraft. There is no uncertainty about intruder movements, as they are clearly defined in the shared information. This data sharing towards deconflicting can be done by setting a negotiation mechanism, where aircraft communicate towards an agreement [45], and/or prioritisation in which a lower priority aircraft follows a resolution manoeuvre based on communication from aircraft with more priority. There are advantages for both cases; negotiation allows aircraft to share/act according to their preferred policy. The objective is for the final solution to be the best globally possible for all. However, in any negotiation there is the risk of a deadlock, where aircraft communicate indefinitely without reaching an agreement. Some sort of prioritisation, respected by all involved aircraft, can limit the number of interactions. Priority can be based on factors such as aircraft's current speed, proximity to destination, rules of the air (RoTA) [12], conflict geometry, or even type of operation. In any case, the rate of communication is a crucial factor. The communication frequency of the network is limited in bandwidth, and aircraft may be unable to exchange data at high frequency. Thus, the number of interactions in any case must be limited compared to a real-life scenario. The number of data transmissions necessary to reach an agreement, to establish a priority (when not implicit), or of sequential messages to the next aircraft in a priority sequence, must be optimised according to this limit. Additionally, a break condition must be added to the communication cycle to prevent the aircraft from negotiating or waiting for data from other aircraft indefinitely.

Approximately one third of the researched CR methods do not follow either of the previously mentioned categories. For unmanned aviation, this is mainly in cases where only static obstacles are expected. Therefore, there is no uncertainty regarding future behaviour, or in cases where other aircraft do not have a conflict resolution mechanism and their path is thus not expected to suffer alterations (e.g., Klaus [46], Teo [47]). For manned aviation, different approaches include mostly research works focused on airspace structure to guarantee minimum separation. Works such as Mao [48], Treleaven [49], and Christodoulou [50], resort to traffic flows that limit aircraft movement. These flows are separated by a safe margin, and the lateral displacement when aircraft switch to a different flow is coordinated. Finally, other studies, such as Bilimoria [33], Christodoulou [50], and Lupu [51], focus predominantly on the effects of different manoeuvres in similar conflict situations.

2.2.6. Multi-Actor Conflict Resolution

Centralised and distributed systems have different approaches to multi-actor conflicts. The former works towards a joint optimisation of all involved trajectories, until a safe distance between all traffic is achieved. In such centralised systems, the number of conflicts and the degree of connection between trajectories will affect the speed with which the system will converge to its solution. It may also occur in complex situations that no solution is found. Centralised approaches may be divided into two main categories: (1) sequential algorithms that optimise trajectories one by one according to the prioritisation

of aircraft [52], and (2) concurrent resolution, where all trajectories are computed simultaneously [53]. The first of these two approaches is less computationally demanding; for each interaction, the system iterates over possible trajectories for a specific aircraft. Once a safe trajectory is found, it moves on to the next aircraft. When a safe trajectory is identified for each aircraft, a solution is found. This approach requires an adequate prioritisation order, to be able to guarantee the identification of safe trajectories for all involved aircraft [54, 55]. Concurrent resolution methods do not require prioritisation. However, application of such methods is often only possible under the assumption of limited uncertainty, which is required to reduce the complexity of the calculations. Durand [34] mentions, for example, an assumption of constant speeds and perfect trajectory prediction, or having the manoeuvres start at the same known optimisation time step.

For distributed systems, resolution manoeuvres adopt the point of view of each aircraft and local optimisation is the objective. At higher traffic densities, where conflicting aircraft pairs can no longer be considered disconnected from other traffic, this local optimisation does not guarantee a globally optimal solution, and there is a risk of unwanted emergent behaviour from interactions between multiple aircraft working individually. The resolution capacity of distributed systems is limited to the intruders that the aircraft is capable of detecting. The solution to a subset of aircraft can unknowingly lead to future secondary conflicts with other aircraft, creating a chain reaction of conflicts, or in ultimate, very high traffic density cases, infinitely perpetuating chain conflicts, or Brownian motion [56, 57]. How distributed methods deal with multi-actor conflicts is therefore a key characteristic of these methods. In this chapter, we distinguish between three distributed approaches to multi-actor conflicts: joint solution, pairwise sequential, and pairwise summed. In a joint solution, multiple intruders are considered simultaneously, and a single solution is found that simultaneously resolves all conflicts in which the ownship is involved. To limit the complexity of a solution, CR models normally detect and resolve in a limited look-ahead time. Other distributed approaches generate pairwise resolutions, focusing only on individual conflict pairs. In pairwise sequential resolution, each manoeuvre resolves a conflict with an intruder, starting with the conflict of highest priority. Other methods, such as Hoekstra [15], sum the resolution vectors resulting from each pairwise resolution (i.e., pairwise summed). A single manoeuvre is then computed and performed resulting from this sum. The choice of whether to employ a pairwise or joint resolution also has consequences on the method's ability for implicit coordination. As previously mentioned, for example, the 'shortest-way-out' principle in pairwise conflicts ensures implicit coordination. However, when summing or in a joint solution implicit coordination is not guaranteed. However, as shown by Hoekstra [15], the summing of the resolution vectors has a beneficial emergent, global effect of distributing the available airspace between the different vehicles.

2.2.7. Avoidance Planning

Planning of a manoeuvre can be defined per the look-ahead time and the state of the aircraft after the resolution manoeuvre is performed. Strategic is a long-range action that changes the flight path significantly; tactical is a mid-range action that changes a small part of the flight path; escape is a short-term manoeuvre that brings aircraft to safety without considerations regarding the flight path. Figure 2.3 illustrates the differences.
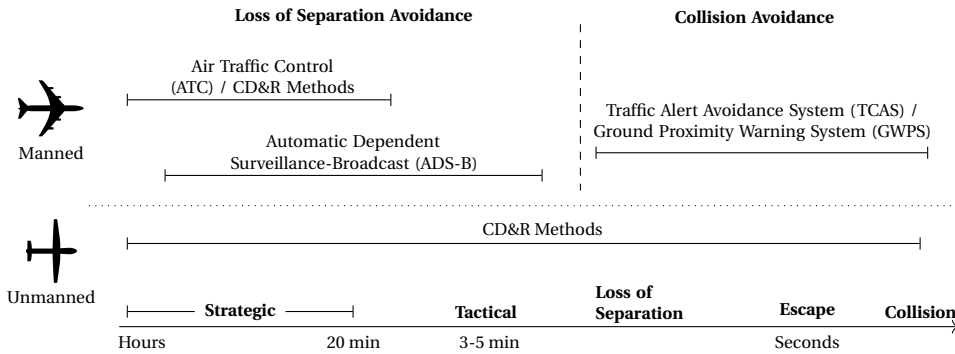
Figure 2.3: Conflict detection and resolution methods for manned and unmanned aviation per look-ahead time.

A strategic manoeuvre (Figure 2.4(a)) is normally employed with more than 20 minutes to loss of separation (LoS), and may even extend to a pre-departure action. It considerably affects the planned flight, as future waypoints are modified to avoid conflict. In manned aviation, ATC is responsible for strategic and tactical avoidance planning. One of the ways to aid air traffic controllers would be to delegate part (or all) of the separation responsibility to the aircraft crew. In manned aircraft, this is made possible by resorting to on-board systems which receive broadcast information from nearby traffic; such a system is called ADS-B. In comparison, unmanned aviation often employs (independent) sensors to detect other traffic. Given the physical limitations of such means of surveillance, these are tactical systems. A deviation manoeuvre is carried out to avoid obstacles (Figure 2.4(b)). Of all possible manoeuvres that prevent loss of separation, CR methods attempt to identify one that minimises distance from the desired path, flight time, or even fuel consumption or energy. Recovery to the initial flight plan is often not included in the tactical plan; normally, aircraft will just redirect to the next waypoint after the conflict situation has been resolved.

In manned aviation, CD&R methods are used to avoid minimum separation losses. Escape manoeuvres are not usually employed. Given the large minimum separation distance in manned aviation, i.e., ICAO's [58] definition of 5 NM horizontal separation and 1000 ft vertical separation, a loss of separation does not necessarily represent a collision (see Figure 2.5). In cases where a collision is imminent, the Traffic Alert and Collision Avoidance System (TCAS) and the Ground Proximity Warning System (GPWS) are used instead of CD&R. For these systems, pairwise collision avoidance is the only objective. No similar mechanism is currently available for unmanned aviation, and therefore, CD&R must atone for this gap. Furthermore, there is no predefined standard separation distance, and considerably small values may be used (e.g., 50 m [59]). Thus, there is a higher chance that the drone is close to a collision once it has lost minimum separation. As a result, contrary to maned aviation, unmanned aviation research employs escape manoeuvres (Figure 2.4(c)). This, the last resource within seconds prior to collision, solely attempts to escape the obstacle with no additional considerations. Contrary to a tactical manoeuvre, typically no coordination or optimisation is employed in these cases due to the lack of time.

(a) Strategic planning.   (b) Tactical planning.   (c) Escape planning.
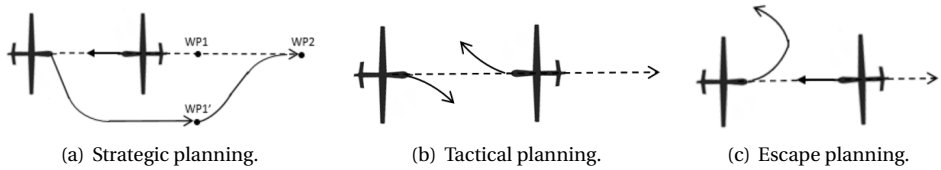
Figure 2.4: Different types of avoidance planning.
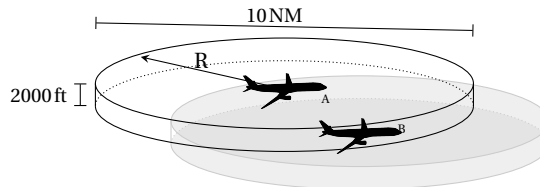
Figure 2.5: The International Civil Aviation Organization's (ICAO) self separation for manned aviation: 5 NM horizontal separation and 1000 ft vertical separation.

2.2.8. Resolution Manoeuvre

To avoid a future loss of minimum separation, several resolution manoeuvres can be used which will change the initially intended trajectory. These can be based on changing the current state: heading variation (Figure 2.6(a)), aircraft change their current heading; speed variation (Figure 2.6(b)), which will change the position of the aircraft for a given point in time; vertical variation (Figure 2.6(c)), where aircraft increase or decrease altitude; or an aircraft can change its future intent by changing its flight-plan. One or multiple of these manoeuvres are performed so as to follow a conflict-free path. Most CR methods are set on decreasing the number of manoeuvres performed, resulting in a minimum deviation from the original path.

(a) Heading manoeuvre.   (b) Speed manoeuvre.   (c) Vertical manoeuvre.
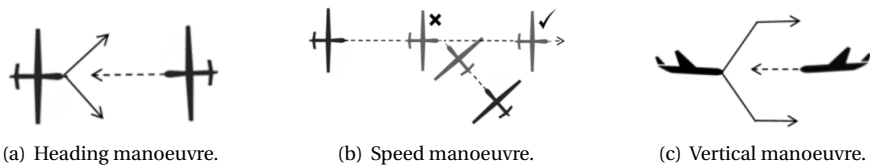
Figure 2.6: Different types of resolution manoeuvres.

Methods are often restricted to manoeuvres in the horizontal plane. Only a small percentage also consider vertical resolution manoeuvres. There are advantages for both. Adding a degree of freedom allows for a variety of conflict resolution movements. However, the extra degree of freedom results in a more complex optimal route calculation. This could be vital, given that a solution must be found before loss of minimum separation. TCAS is singular in applying only vertical manoeuvres. For resolving short-term conflicts, climb/descend is a fast and efficient action since the required vertical separation is smaller than the horizontal one. Sunil [60] showed that for a stratified airspace, having only horizontal resolutions improves stability; fewer conflicts are considered and accounted for with only an horizontal conflict layer. Not including vertical changes is also acceptable

from a performance point of view, as the latter is highly affected by the flight level the aircraft is operating in. Additionally, travelling at high altitudes is not the best scenario for speed manoeuvring. When the stall speed increases, the manoeuvring space decreases.

Initially, most CD&R methods used heading changes as preferred by air traffic controllers, as they often segment the airspace into layers. Lately, speed variation has received new attention with 'subliminal' speed control, which consists of modifying the aircraft speeds within a small range around their original speeds without informing air traffic controllers. As a result, some of the work of air traffic controllers can be automated, thus reducing their workload. Research such as the ERASMUS project [61] and Chaloulos [62] show that, although for simple two-aircraft situations subliminal control can reduce the workload of air traffic controllers, its efficiency depends on the nominal minimal separation between the aircraft and on the time available to loss of separation. Conflict resolution based on speed change alone is only possible with non-(near-)head-on conflicts. The likelihood of these kind of conflicts is dependent on the airspace structure and the heading difference between aircraft flying at similar flight levels. In other methods, such as Hoekstra [15], Rey [63], and Balasooriyan [42], the combination of heading and speed deviations showed potential results.

Flight plan modifications change the waypoints the aircraft is intended to follow. This is similar to real-life operations, with flight paths being defined through successive waypoints. This way of avoiding conflicts has gained new attention with the development of the concept of four-dimensional trajectory-based operations (4DTBO) [64]. This refers to 3D waypoints associated with timestamps that define when the aircraft is expected to reach each waypoint. With 4DTBO, the complete path and duration of the flight can be defined by specifying arrival times for a sequence of waypoints. Whenever it is detected that the initially defined 4D waypoints for all involved aircraft will result in one or more losses of minimum separation, new flight plans are constructed by selecting either different waypoints, different arrival timestamps, or both.

Performance limitations naturally have an impact on the manoeuvrability of aircraft. When defining a conflict resolution, maximum turn rates, and maximum speed and acceleration ranges must be taken into account. Defining a heading and/or speed change which the aircraft cannot successfully complete, will jeopardise the success of the manoeuvre on achieving minimum separation from other traffic. Moreover, different look-ahead requirements may be considered based on speed ranges. For unmanned aviation, taking into account performance differences is an especially important factor given the wide range of possible missions, which can involve many different types of UAVs (e.g., rotorcraft, fixed-wing). To avoid the calculation of resolution manoeuvres outside of the performance limits, methods, such as Van Dam [43], define a solution space bounded by the possible range of speeds; it is not possible to define a resolution manoeuvre outside of these boundaries. However, the speed range is often defined per aircraft type without taking the environment into account. As mentioned above, the manoeuvring space depends on the altitude of the aircraft. Studies such as Lambregts [65], attempt to develop a conflict resolution method with envelope protection functionality that can identify the maximum manoeuvring space in order to take advantage of the full performance capabilities of the UAV.

Finally, resolution manoeuvres may also be distinguished on whether they are discrete

2

or continuous; on whether a resolution is calculated given a discrete state and assumes no modification of this state until the manoeuvre is terminated, or if the environment is observed periodically during the manoeuvre, which is adapted incrementally in response. In theory, most resolution algorithms have a discrete implementation since they calculate resolution manoeuvres that should resolve the conflict without further intervention. However, in practise, these algorithms can still be used to reevaluate conflicts in each update cycle of the implementation. In this case, in each update cycle, where the ownship is detected to be in conflict, the conflict resolution algorithm outputs a resolution manoeuvre given the current state of the environment. As a result, the ownship may change a previously defined resolution manoeuvre at any update step, based on the changed nature of the traffic situation.

2.2.9. Obstacle Types

A CD&R method may prevent collision only with static obstacles, with dynamic obstacles, or with all (i.e., both static and dynamic obstacles). When a model avoids solely static objects, it may be inferred that it has strategic planning, with the trajectory being set before the beginning of the flight in a known environment.

Manned aviation CD&R models will naturally be directed at detecting other dynamic traffic, as these models are used mostly when aircraft are flying at cruise altitude. Note that it is not guaranteed that a model directed at dynamic obstacles can also avoid static obstacles. First, while most of these CD&R models assume obstacles as a circle with radius equal to the minimum separation distance, a static object can have different sizes. Second, most models also assume some sort of coordination and non-zero speed. Most dynamic obstacle oriented CD&R models would have to be enhanced when transposed to, for example, an urban environment where deviation also from static objects, such as buildings, must be guaranteed.

For unmanned aviation, a considerable number of CD&R models still focus solely on static obstacles. However, these can only be used for operations where the environment is well known in advance. This is the case, for example, of an area where a drone must carry an object from a start to an end point, and no other traffic is expected.

2.2.10. Optimisation

For CD&R methods, safety is paramount. However, there is a preference for methods that do not significantly alter the initially planned trajectory or significantly increase the costs of an operation. The efficiency of a CR method can be evaluated with respect to its effect on the time and/or path of the flight or even fuel/energy consumption. Note that a CR method may contain weights of costs which vary based on the mission/situation, thus its efficiency being dependent not only on the intrinsic method but on the weights employed.

A simple way to minimise the path length is to be partial to small heading changes when avoiding obstacles [48]. Minimising flight time can be a direct consequence of minimising flight path when the speed is assumed constant. In other cases, minimising flight time results in a preference for resolution manoeuvres which do not include lowering the aircraft's speed.

Computing fuel expenditure is not direct, as it depends on several physical factors

of the aircraft such as model, speed, and weight at the moment of the operation. A simplification is to opt for the manoeuvre which minimises speed variation [35] as the latter is a major coefficient on fuel waste. From the examined research, the Base of Aircraft Data (BADA) performance model [66] is preferred for fuel consumption calculations [63]. For unmanned aviation, energy efficiency based CD&R is currently being investigated as more drones are developed and more information on these systems is made available. Research, such as Dietrich [67] and Stolaroff [68], offer a first look at estimating drones' energy consumption.

2.2.11. Reviewed Conflict Detection & Resolution Models

The reviewed manned and unmanned CR methods are presented in Tables 2.3 and 2.4, respectively. Table 2.5 serves as an indication of the abbreviations used for each category.

2.3. Experiment: Direct Comparison of CR Methods

This section describes the design of the fast-time simulation experiments conducted in order to compare four conflict resolution methods in terms of safety and efficiency. The implementation code can be accessed online at [140]; scenarios and result files are available at [141].

2.3.1. Apparatus and Aircraft Models

The evaluation is performed using the open-source Air Traffic Simulator BlueSky [25]. This section gives a description of the most relevant aspects of this simulator, and of the scenarios that are used to compare concepts. The exact implementation of the simulator set-up, the scenarios, and the resolution algorithms are available online [140, 141]. The BlueSky simulator tool can be used to easily implement and evaluate different CD&R methods, allowing for all CD&R to be tested under the same scenarios and conditions. The simulation scenarios are based on the work of Sunil [142]. These scenarios were chosen as they represent a homogeneous traffic picture, uniform in terms of altitude, spatial, and speed distribution. The results thus reflect the ideal behaviour of the CR method, and not its response to agglomerates of aircraft that are unaccounted for.

Bluesky uses a kinematic aircraft performance model based entirely on open data [143]. Different aircraft types can be introduced into the Bluesky simulation when performance limits are known. The aircraft in the simulation are all Boeing 747-400's and DJI Mavic Pro quadcopters, for manned and unmanned aviation, respectively. These types of aircraft were selected for their significant speed range. In this way, the limitations of the aircraft flight envelope affect the resolution choices as little as possible. The characteristics of these aircraft are presented in Table 2.6. The data for the B747-400 aircraft comes from BADA [66]. For the DJI Mavic Pro model, speed and mass were retrieved from the manufacturer's data. Although exact turn rate and acceleration/braking values are not available, generic values were assumed.

As mentioned above, performance limitations have an impact on the manoeuvrability of the ownship aircraft, which in turn limits the range of actions that can be performed to avoid a conflict. For unmanned aviation, this work employs the DJI Mavic Pro, a well-known model used in a wide range of applications [144–146]. However, a mission

2

Table 2.3: Conflict detection and resolution methods for manned aviation. Table 2.5 defines abbreviations.

	Surv	Traj	PAsm	Control	MultiActor	Plan	AvMan	Obst	Examples
	C	S		C		S + T	H + V	A	ATC
	D	S		D		T		D	ADS-B
	D	S	N	D	PSE		V	D	TCAS
	D	S	N	D	PSE		H/V	D	TCAS II [69]
	D	S	P	D	PSE		V	D	TCAS X [70]
	D	S	N	D	PSE		V	D	GPWS
	C	I	P	-	-	-	-	D	Vink [71]
Exact	C	S	N	C	C	S	H/S	D	Cafieri [72] [1]
	C	S	N	C	C	T	H/S	D	Pallottino [53]
	C	S	N	C	C	S	H + S	D	Vela [73]
	C	S	N	C	C	S	H + V	D	Hu [52]
	C	S	P	C	C	S	S	D	Rey [63]
	C	S	P	C	C	S	FP	D	Chen [74]
	C	I	N	C	C	T	FP	D	Le Ny [75]
	C	I	N	C	C	S	FP	D	Hu [52]
	C	I	P	C	C	S	FP	D	Niedringhaus [76] [2]
Heuristic	C	S	N	C	S	T	H	D	Ayuso [37]
	C	S	N	C	S	T	H	D	Liu [38]
	C	S	N	C	S	T	H/S/V	D	Ayuso [77]
	C	S	P	C	S	S	H	D	Durand [78]
	C	S	P	C	S	T	H	D	Sathyan [39]
	C	S	P	C	S	T	H	D	Yang [79, 80]
	C	S	P	C	S	T	H	D	Allignol [81]
	C	S	P	C	S	T	H + S	D	Tomlin [82]
	C	I	P	C	S	S	FP	D	Visintini [83]
	C	I	P	C	S	S	FP	D	Prandini [84]
	C	I	P	C	S	S	FP	D	Hao [85] [1,3]
Explicitly Negotiated	D	S	N	D	PSE	T	H	D	Chipalkatty [86] [2]
	D	S	N	D	PSE	T	FP	D	Pritchett [87]
	D	I	N	D	J	T	FP	D	Sislak [88] [1]
	D	I	N	D	PSE	T	H + S	D	Harper [89]
	D	I	N	D	PSE	T	H	D	Blin [90]
	D	I	P	D	PSE	T	FP	D	Bicchi [91]
	D	I	P	D	PSE	T	H	D	Granger [92]
	D	S	N	D	J	T	H + S	D	Balasooriyan [42] [1]
Reactive	D	S	N	D	PSU	T	H + S + V	D	Hoekstra [15] [1]
	D	S	P	D	PSE	T	H/S	D	Paielli [93]
	D	I	N	D	J	T	H + S	D	Van Dam [43] [1]
	D	I	N	D	J	T	H + S	D	Velasco [94]
Prescribed	D	-	-	D	-	T	H	D	RoW [41], RoTA [12]
	C	S	N	C	C	T	H	D	Mao [48]
	C	S	N	C	S	T	H	D	Treleaven [49]
	C	S	N	C	S	T	H	D	Huang [95]
	C	S	P	D	S	T	H/V	A	Viebahn [96]
	D	S	N	D	J	S	H	D	Devasia [97]
	D	S	N	D	PSE	T	H	D	Zhao [98]
	D	S	N	D	PSE	T	H	D	Mao [99]
	D	S	N	D	J	T	S	D	Christodoulou [50]
Other	D	S	N	D	PSE	T	H/S/V	D	Bilimoria [100]
	D	S	N	D	PSE	T	H/S/V	D	Krozel [101]
	D	S	N	D	PSE	T	H + S	D	Lupu [51]
	D	S	P	D	PSE	T	H	D	Zhang [102]
	D	S	N	D	PSE	T	H/S	D	Peng [103]
	D	I	P	D	PSE	T	-	D	Yang [27]
	D	I	N	D	J	T	FP	D	Menon [104]
	D	I	N	D	PSE	T	FP	D	Burdun [105]
	D	-	N	D	J	T	FP	S	Patel [106]

[1] Minimises path length. [2] Minimises time length. [3] Increases distance to threats.

Table 2.4: Conflict detection and resolution methods for unmanned aviation. Table 2.5 defines abbreviations.

	Surv	Traj	PAsm	Control	MultiActor	Plan	AvMan	Obst	Examples
Exact	C	S	N	C	C	T	H + S	D	Alonso-Mora [107]
	C	I	N	C	C	S	FP	D	Borrelli [35]
	I	-	-	C	C	S	H + V	S	Kelly [108] [1,3]
Heuristic	C	S	P	C	S	T	H	A	Yi Ong [109]
	C	I	N	C	S	S	FP	D	Borrelli [35]
	C	I	N	C	S	S	FP	D	Alejo [59]
	C	I	N	C	S	S	FP	D	Beard [110] [1,3]
	C	-	-	C	-	S	FP	S	Nikolos [111]
	C	S	N	C	S	T	H	D	Ho [112]
	C	I	N	C	S	T	FP	A	Liao [113]
	C	S	N	C	S	T	H + V	A	Richards [114] [2]
	C	S	N	C	S	T	FP	D	Fasano [115]
	C	S	P	C	S	T	FP	D	Rathbun [40]
	C	S	N	C	S	T	H + S	D	Alonso-Mora [107]
	I	-	-	C	-	S	FP	S	Langelaan [116] [1,3]
	I	-	-	C	S	S	H	S	Obermeyer [117]
Explicitly Negotiated	D	S	N	D	PSE	T	H	D	Park [118]
	D	S	N	D	J	T	H	D	Duan [119] [1,3]
	D	S	N	D	PSE	T	V	D	Manathara [120]
	D	S	P	D	PSE	T	H	D	Yang [45]
	D	S	P	D	J	T	FP	D	Prevost [121]
	D	S	N	D	PSE	E	V	D	Zeitlin [122]
Reactive	D	S	P	D	J	T	H	A	Yang [123]
	D	S	N	D	J	T	H + S	D	Alonso-Mora [107]
	D	S	N	D	J	T	H	D	Balachandran [44]
	D	S	N	D	PSE	T	S	D	Mujumdar [124]
	D	S	N	D	J	T	H + S	D	Alonso-Mora [107]
	D	S	N	D	J	T	H + S	D	Jenie [14]
	D	S	N	D	PSE	T	H + V	D	Leonard [125]
Prescribed	D	-	-	D	-	T	H	D	RoW [41], RoTA [12]
Other	C	-	N	D	J	T	FP	S	Yang [126] [1,2]
	D	S	N	D	PSE	T	H	D	Zhu [127]
	D	S	N	D	PSE	T	H	D	Hwang [128]
	D	S	N	D	PSE	T	H/V	D	Jilkov [129]
	D	I	-	-	J	T	FP	S	Hurley [130]
	I	S	N	D	J	T	H + V	A	Kitamura [131]
	I	-	N	D	J	T	FP	S	Hrabar [132]
	I	-	N	D	J	T	H	S	Jung [133]
	I	-	-	D	PSE	T	H	S	Schmitt [134] [1]
	I	-	-	D	J	T	FP	S	Chowdhary [135]
	I	-	-	D	J	T	FP	S	Nikolos [111]
	I	S	P	D	PSE	T	H	D	Klaus [46]
	I	S	N	D	PSE	E	H + S + V	D	Teo [47] [3]
	I	-	-	D	J	E	H + V	S	Beyeler [136]
	I	-	-	D	J	E	H + V	S	deCroon [137, 138]
	I	-	-	D	J	E	H + V	S	Muller [139]

[1] Minimises path length. [2] Minimises time length. [3] Increases distance to threats.

2

Table 2.5: Abbreviations for the Conflict Detection and Resolution (CD&R) categories in Tables 2.3 and 2.4.

Category	Abbreviation	Meaning
Surveillance (Surv)	C	Centralised Dependent
	D	Distributed Dependent
	I	Independent
Trajectory Propagation (Traj)	S	State-based
	I	Intent-based
Predictability Assumption (PAsm)	N	Nominal
	P	Probabilistic
	WC	Worst-case
Control	C	Centralised
	D	Distributed
Multi-Actor Conflict Resolution (MultiActor)	S	Sequential
	C	Concurrent
	PSE	Pairwise Sequential
	PSU	Pairwise Summed
	J	Joint Solution
Avoidance Planning (Plan)	S	Strategic
	T	Tactical
	E	Escape
Resolution Manoeuvre (AvMan)	H	Heading
	S	Speed
	V	Vertical
	H + V	Horizontal AND vertical simultaneously
	H/V	Can choose either horizontal or vertical
	FP	Flight-Plan
Obstacle Types (Obst)	S	Static
	D	Dynamic
	A	All

Table 2.6: Performance data for Boeing 747-400 and DJI Mavic Pro used with BlueSky simulations.

	Boeing 747-400	DJI Mavic Pro
Speed [kts]	450–500	-35–35
Mach [-]	0.784–0.871	–
Mass [kg]	285.700	0.734
Turn Rate [°/s]	1.53–1.70	max: 15
Load Factor in Turns	1.22	–
Acceleration/Breaking [kts/s]	1.0	1.0

employing an UAS model with significant differences in performance (e.g., a fixed-wing model), should not directly extrapolate from the results herein obtained.

2.3.2. INDEPENDENT VARIABLES

Two independent variable are considered in this experiment: traffic density, and conflict resolution methods.

2

The experimental scenarios build the volume of traffic from zero to a desired value, after which traffic density is maintained at this desired value. Traffic density varies from low to high according to Table 2.7. The instantaneous aircraft value defines the number of aircraft expected at any given moment during the measurement period. Given the duration of the measurement and the average flight time, the simulator constantly spawns (adds to the simulation) aircraft at the same rate as these are removed from the simulation, to keep a constant traffic density. Density values were defined on the basis of current expectations. In 2017, the Netherlands had a maximum traffic density of 32 aircraft per 10,000 NM^2 in the upper airspace [142]. Given the traffic increase expectations [147], Netherlands may then expect up to 45 aircraft per 10,000 NM^2 by 2025. Unmanned aircraft are considered for a hypothetical situation where drones are used for light-weight parcel deliveries. For the urban area of Paris, this would represent over 1 million drones per 10,000 NM^2 by 2035 [22]. To keep computation times reasonable, lower densities were selected.

Table 2.7: Traffic volumes used in simulation.

		Traffic Density [ac/10,000 NM^2]	**Instantaneous Ac.**	**Spawned Ac.**
Manned Aviation	Low	32	648	3070
	Medium	37	768	3640
	High	45	911	4317
Unmanned Aviation	Low	12,000	1080	4629
	Medium	13,856	247	5345
	High	16,000	1440	6172

Conflict Resolution Methods

Four commonly used conflict resolution methods were chosen for direct comparison. The following section gives a description of these methods, their assumptions, and compares them in terms of planning, control, coordination, and multi-actor conflict resolution. The exact implementation of these methods, and the rest of the simulation set-up are available online [140, 141].

- Reactive: in this category, coordination is implicit and adopted by all aircraft; no negotiation is necessary. Here, we explore two different methods that use implicit coordination by adopting the 'shortest-way-out' principle. The minimum heading/speed displacement which moves the CPA between two conflicting aircraft to the edge of the intruder's PZ is calculated using the velocity obstacles (VO) theory. A VO is defined as the set of all velocity vectors of a moving agent which will result in a loss of separation with a (moving) obstacle at some future point in time [148, 149]. Figure 2.7(a) illustrates a traffic situation in which the ownship aircraft is in conflict with an intruder. As a first step, the collision cone (CC) is defined by lines from the ownship to the intruder, tangential to both sides of the intruder's protected zone. The ownship and intruder are in conflict when the relative velocity is inside the CC. By translating the CC with the intruder's velocity, the VO in Figure 2.7(b) is

obtained. This VO represents the set of ownship velocities that will result in a loss of separation with the intruder.
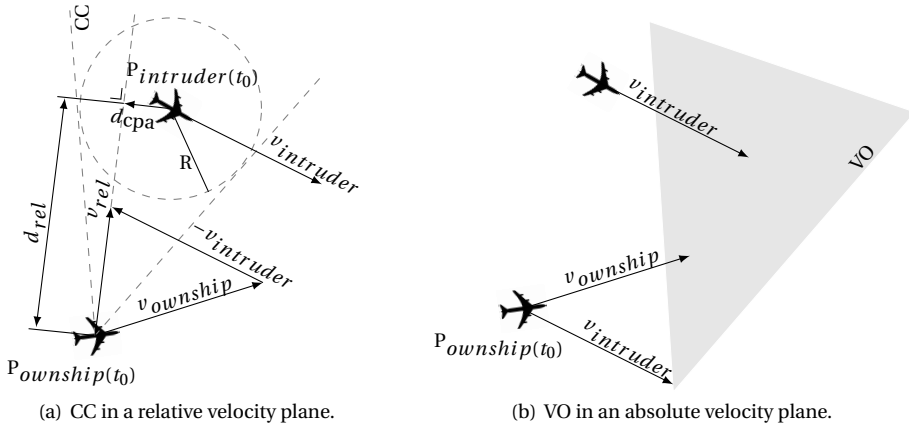

(a) CC in a relative velocity plane.
(b) VO in an absolute velocity plane.

Figure 2.7: Cone of collision (CC) and a velocity obstacle (VO) in a situation of a future loss of separation. R represents the radius of the protected zone. $P_{ownship}(t_0)$ and $P_{intruder}(t_0)$ denote the ownship's and the intruder's initial position, respectively. $v_{ownship}$ is the observed aircraft velocity vector, $v_{intruder}$ is the intruder velocity vector, and v_{rel} is the relative velocity vector. d_{rel} is the relative distance vector, and d_{CPA} indicates the distance at the closest point of approach (CPA).

The two reactive methods differ in how they deal with multi-actor conflicts, and will allow for a comparison between pairwise and joint resolution:

1. Potential field [9, 15]: predicted conflicting aircraft positions are represented by 'charged particles' which simultaneously push and are pushed away from the conflicting aircraft. In the evaluation in this chapter, this category of CR methods will be represented by a 'bare' version of the Modified Voltage Potential (MVP) method [15], for which the geometric resolution is shown in Figure 2.8. For conflicting aircraft, the predicted positions at the closest point of approach (CPA) 'repel' each other. This 'repelling force' is converted to a displacement of the predicted position at CPA, in a way that the minimum distance will be equal to the required minimum separation between aircraft. Such a displacement results in a new advised heading and speed, in the direction that increases the predicted CPA. Choosing this direction for each resolution ensures that the MVP is implicitly coordinated for 2-aircraft conflicts. Both aircraft will take complimentary measures to evade the other. In case of multi-aircraft conflicts, resolution vectors are summed for each conflict pair. This method has the advantage of simplicity; the resulting calculations are computationally light, and the geometric representation allows other possible constraints to be taken into account easily. On the other hand, because resolutions are solely based on the conflict geometry, they may oppose the desired flight direction as proposed by the flight plan.
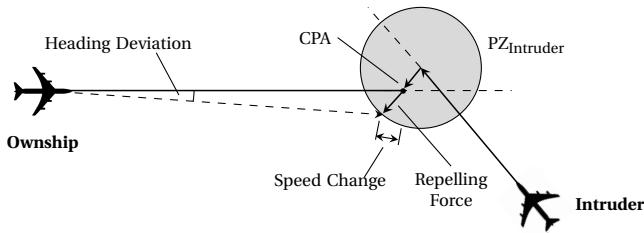
Figure 2.8: Modified Voltage Potential (MVP) resolution. Adapted from Hoekstra [15].

2. Solution Space [14, 43]: VO theory is used in combination with kinematic constraints to determine a set of reachable, conflict-free velocity vectors, and a set of reachable, conflicting velocity vectors. These two sets of velocities together form the solution space. Figure 2.9 shows this velocity space for aircraft: two concentric circles, representing the minimum and maximum velocities of an aircraft, bound all reachable combinations of heading and speed. Within this reachable velocity space, VOs are constructed for each proximate aircraft, each representing the set of reachable heading/velocity combinations that result in a conflict with the respective aircraft. When all relevant VOs are subtracted from the set of reachable velocities, what remains is the set of reachable, conflict-free heading/speed combinations. Solution space CR methods determine resolution manoeuvres by selecting heading/speed combinations from this set of conflict-free, reachable velocities. As a result, these methods provide resolutions that allow multiple conflicts to be solved simultaneously.

In two-aircraft situations, these methods behave similarly to potential field VO methods. In multi-aircraft situations, they act as described above. Implicit coordination is also an issue for these methods in multi-aircraft conflicts, and additional coordination rules are required in these situations. The algorithm herein used is the Solution Space Diagram (SSD) method as implemented by Balasooriyan [42]. Identification of a conflict-free resolution vector consists of finding a point within the set of spaces within the velocity limits that do not intersect the VOs [150]. The speed vector resulting in the 'shortest-way-out' manoeuvre (i.e., shortest speed/heading deviation) is picked.
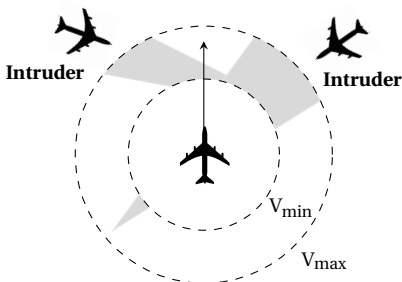

Figure 2.9: Solution space diagram (SSD) resolution. Adapted from Van Dam [43].

- Explicit coordination: this coordination works on the basis that aircraft communicate their intention, and thus there is no uncertainty regarding their future movement. We use a negotiation approach in which each aircraft sends its deconflicting policy to intruders until all broadcast policies result in a global solution. We assume a communication cycle similar to Yang [45], displayed in Figure 2.10. This was used due to its satisfactory performance in dealing with complex conflict scenarios as demonstrated by the authors. Two aircraft share information when they are in a pairwise conflict; 'neighbours' is the set of intruders the ownship is in conflict with. Aircraft work on the assumption that each aircraft primarily acts towards avoiding losses of minimum separation. First, each aircraft finds a set of conflict-free resolution manoeuvres. It must also be guaranteed that the manoeuvres within this set will not create new conflicts with other nearby aircraft. This set of solutions is found by identifying the safe interval between heading/speed displacements that cross the edge of intruders' protected zone. A preference for a more significant heading or speed change is based on the aircraft's own policy; the ultimate goal is to achieve an optimal solution for all aircraft. Each aircraft then identifies the preferred resolution manoeuvre and broadcasts it to the local neighbours.

Once an aircraft receives the neighbours' manoeuvres, it will verify whether all conflicts are resolved. If so, communication is terminated, and the aircraft adopts the previously computed resolution manoeuvre. Otherwise, aircraft use the received intent information from the neighbours to update the set of conflict-free solutions. A new resolution manoeuvre is selected from this set. However, now preference is for a manoeuvre within the smallest variation from the previously broadcast manoeuvre in an attempt to converge faster to a solution.

In a real-world situation, the time delay between generation and reception of a message is crucial. Studies, such as Yang [45], focus on optimising the convergence to an agreement and demonstrating that a reduced number of negotiation cycles is required to achieve a robust solution. Our objective, however, is to see how the method behaves within this limited number of negotiation iterations. Yang [45] obtained an average number of iterations below five, albeit for smaller traffic densities. We chose to use this value to limit computational effort. However, it should be noted that a higher limit could favor more robust resolution manoeuvres.
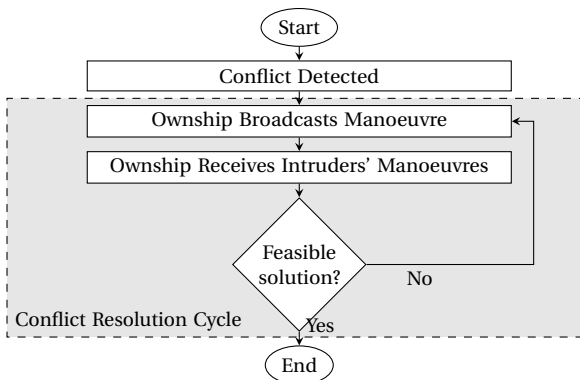

Figure 2.10: Iterations of an explicitly negotiated solution. Adapted from Yang [45].

2

- Sequential cost: in which a single agent is responsible for redirecting the aircraft. It is assumed that aircraft will follow the guidelines set by this agent and thus uncertainty is reduced. At each update step, if conflicts are found, conflicting aircraft are redirected towards preventing loss of separation. We follow a sequential approach, setting an order based on the time to loss of separation. Note that the aircraft order can be defined over multi-criteria and will have an impact on the final trajectories. With each aircraft, the possible paths are considered; these are a discrete set of possible heading/speed changes restricted by the aircraft's performance range. The cost for each trajectory is calculated and the path with the lowest cost is chosen. The cost definition used in the simulations performed here is similar to Hao's [85]:

$$\begin{cases} \text{Path cost} = w_l \Delta P_L + w_v \Delta V + w_d D + \delta P \\ w_l + w_v + w_d = 1 \end{cases}, \qquad (2.1)$$

where ΔP_L represents the variation of the total length of the path, ΔV the change in velocity, and D the distance to intruders. Lastly, a penalty value P is used to add an extra cost to trajectories which cross an intruder's PZ, as to make these more expensive and, therefore, less desirable. The weight coefficients, w_l, w_v, and w_D indicate the weights given to the path length variation, to the change in velocity, and to the distance to intruders, respectively. The value of the weight coefficients denotes their importance. If, for example, a lower fuel consumption is favoured over distance to threats, then w_l and w_v should be given higher values, as to make an increment in flight path or speed variation significantly expensive. When summed, the weight coefficients are equal to one. Note that other properties could be added to the cost equation according to preference.

The chosen weights have an influence on the overall results. When prioritisation is set over efficiency, it might have a negative effect on safety and vice-versa. In our work, we chose to emphasize lower fuel consumption, focusing on smaller nominal trajectory deviations. A penalty value for losses of separation is used, proportionally to its severity. The same weights were used both for manned and unmanned aviation, with the purpose of observing possible differences in performance.

Table 2.8 describes the main differences between the four CR methods that are considered in this comparison. All act on a tactical timescale, and all but the cost method, have distributed control. While the coord method focuses on explicit intent communication with other aircraft, in MVP and SSD each aircraft chooses its conflict resolution without negotiation. Instead, implicit coordination is introduced in pairwise conflicts through the use of the 'shortest-way-out' resolution strategy. MVP resolves pairwise conflicts, summing resolution vectors in case of multi-aircraft conflicts, whereas SSD decides upon a joint resolution manoeuvre which resolves conflict with all aircraft simultaneously. All methods can perform the same type of manoeuvre: heading and/or speed change. There is no limitation on the number of turns; every aircraft is free to perform the desired resolution manoeuvre. Conflict evaluation interval equals one second; each second, current conflicts and LoSs are detected and the CR method is computed if necessary. An aircraft adopts the manoeuvre output by the CR method, until it is past CPA. At this point, it will redirect to the next waypoint. Wind or performance uncertainties were not considered.

Table 2.8: Properties of the conflict resolution (CR) methods used in simulation.

CR Methods				
Planning:	Tactical			
Control:	Distributed			Centralised
Method Category:	Reactive		Explicitly Negotiated	Heuristic
Multi-Actor Resolution:	Pairwise Summed	Joint Solution	**Coord**	**Cost**
	MVP	**SSD**		

2.4. Experimental Design and Procedure

2.4.1. Minimum Separation

The value of the minimum safe separation may depend on the density of air traffic and the region of the airspace. However, for manned aviation, most CD&R studies use ICAO's [58] definition of 5 NM horizontal separation and 1000 ft vertical separation. For unmanned aviation, there are no established separation distance standards yet, although 50 m for horizontal separation is a value commonly used in research [59] and will therefore be used in the experiments herein performed. For vertical separation, 65 ft was assumed.

2.4.2. Conflict Detection

The experiment will employ state-based conflict detection for all conditions. This assumes a linear propagation of the current state of all involved aircraft. Using this approach, the time to CPA (in seconds) is calculated as:

$$t_{CPA} = -\frac{\vec{d}_{rel} \cdot \vec{v}_{rel}}{\vec{v}_{rel}}, \tag{2.2}$$

where $\vec{d}_{rel}$ is the cartesian distance vector between the involved aircraft (in meters), and $\vec{v}_{rel}$ the vector difference between the velocity vectors of the involved aircraft (in meters per second). The distance between aircraft at CPA (in meters) is calculated as:

$$d_{CPA} = \sqrt{\vec{d}_{rel}^2 - t_{CPA}^2 \cdot \vec{v}_{rel}^2}. \tag{2.3}$$

When the separation distance is calculated to be smaller than the specified minimal horizontal spacing, a time interval can be calculated in which separation will be lost if no action is taken:

$$t_{in}, t_{out} = t_{CPA} \pm \frac{\sqrt{R_{PZ}^2 - d_{CPA}^2}}{\vec{v}_{rel}} \tag{2.4}$$

These equations will be used to detect conflicts, which are said to occur when $d_{CPA} < R_{PZ}$, and $t_{in} \leq t_{lookahead}$, where R_{PZ} is the radius of the protected zone, or the minimum horizontal separation, and $t_{lookahead}$ is the specified look-ahead time. A five-minute look-ahead time is used for conflict detection for both manned and unmanned aviation. Note that the look-ahead distance will be bigger for manned aviation, as manned aircraft will cross a longer path in the five minutes.

This analytical calculation of the time to loss of separation herein performed has the advantage of not requiring pre-defined nodes. It should be noted that some conflict

detection models, especially when using a flight plan or intent information, opt for calculating distance at CPA through the discretisation of a 4D path, where spatial nodes represent all the possible states within the simulation space. Conflict detection is then based on checking if any aircraft occupy nodes closer than the minimum separation distance at any point in time.

2.4.3. Simulation Scenarios

We first define the measurement area: this is a square area with its dimensions determined by the average True Air Speed (TAS) and the average flight time. The aircraft spawn locations (origins) and destinations are placed in alternating order at the edge of this area, with a spacing equal to the minimum separation distance plus a 10% margin, to avoid conflicts between the spawn aircraft and the aircraft arriving at their destination. Additionally, to prevent very short-term conflicts between just spawned aircraft and pre-existing cruising traffic, aircraft are spawned at a lower altitude, after which they climb to a common cruise level. Unmanned aircraft are expected to climb almost vertically. Aircraft fly a straight line towards their destination, with a constant heading computed with a normal distribution random number generator, varying between 0° and 360°. This straight line is formed by several waypoints within the measurement area. These waypoints prevent the aircraft from leaving the measurement area in an attempt to avoid conflicts. Logging is restricted to the cruise phase of the flight. The cruise flight level is the same for all aircraft. The total planned flight distance is uniformly distributed between a pre-defined minimum and maximum value based on a minimum flight time and the average TAS. TAS values vary between TAS_{min} and TAS_{max}, as specified by the respective aircraft model. Note that no wind was considered.

Ideally, aircraft would only operate within the measurement area, thereby ensuring a constant density of aircraft within that area. However, aircraft may temporarily leave the measurement area during the resolution of a conflict and should not be deleted in this case. Therefore, a second, larger area encompassing the measurement area is considered: the experiment area. As a result, aircraft in a conflict situation close to their origin or destination are not deleted incorrectly from the simulation. Ultimately, an aircraft is deleted once it leaves the experiment area or comes close to the ground for landing. Note that we assume a no-boundary setting, with sufficient flight space around the measurement area, in order to avoid edge effects from influencing the results.

Each scenario consists of a build-up period to reach a steady state in terms of traffic volume and traffic pattern. The build-up is followed by the logging phase, during which traffic volume is held constant, and a build-down period, allowing aircraft created during the logging period to finish their flights. The experiment is repeated multiple times with different origin-destination combinations. More details are shown in Table 2.9.

2.4.4. Dependent Measures

Three different categories of measures are used to compare the simulated conflict resolution methods: safety, stability, and efficiency.

2

Table 2.9: Properties of the manned and unmanned aviation scenarios used in simulation.

	Manned Aviation	Unmanned Aviation
Scenario Duration [h]	3	
Number of Repetitions [-]	3	
Min Flight Time [h]	0.5	
Experiment Duration [h]	1 h 30 m (45 m–2 h 15 m)	
Measurement Area [NM^2]	202,500	900
Experiment Area [NM^2]	405,000	1800
Min Flight Distance [NM]	200	15
Max Flight Distance [NM]	250	20
Radius PZ Horizontal [NM]	5	0.027
Radius PZ Vertical [ft]	1000	65
Min TAS [kts]	450	5
Average TAS [kts]	470	30
Max TAS [kts]	500	35
Average Time Flight [min]	40	40
Flight Level [ft]	36,000	300

SAFETY ANALYSIS

Safety is defined in terms of the number and duration of conflicts and losses of separation, where fewer conflicts and losses of separation are considered safer. Additionally, losses of separation are distinguished based on their severity according to how close aircraft get to each other:

$$LoS_{sev} = \frac{R - d_{CPA}}{R}.$$ (2.5)

A low separation severity is preferred.

STABILITY ANALYSIS

Stability refers to the tendency for tactical conflict resolution manoeuvres to create secondary conflicts. Deviating from the nominal path, in order to avoid conflicts, often results in a longer flight path. At high traffic densities, conflict-free airspace is scarce, and when each aircraft requires a larger portion of the airspace it often results in more conflicts. Therefore, tactical resolution manoeuvres tend to create conflict chain reactions. In the literature, this effect has been measured using the Domino Effect Parameter (DEP) [151]. The latter can be calculated as follows:

$$DEP = \frac{n_{cfl}^{ON} - n_{cfl}^{OFF}}{n_{cfl}^{OFF}},$$ (2.6)

where n_{cfl}^{ON} and n_{cfl}^{OFF} represent the number of conflicts with CD&R ON and OFF, respectively. A higher DEP value indicates a more destabilising method, creating more conflict chain reactions.

EFFICIENCY ANALYSIS

Efficiency is evaluated in terms of the distance travelled and duration of the flight. The added flight distance and time are compared to the baseline case where no conflict resolution is performed, and aircraft follow their straight trajectories from origin to destination. A CR method that results in considerable path deviations, significantly increasing the path travelled and/or the duration of the flight is considered inefficient. Furthermore, for manned aviation, the work done (W) associated with fuel consumption can be calculated:

$$W = \int_{path} \vec{T} \cdot d\vec{s},\tag{2.7}$$

where $\vec{T}$ and $d\vec{s}$ represent the thrust vector and the displacement vector along the path, respectively. For unmanned aviation, we are not able to calculate the work done, as we do not currently have a drag model for drone vehicles.

2.5. EXPERIMENTAL HYPOTHESES

Naturally, it was hypothesised that as the traffic density increases, all safety, efficiency, and stability parameters worsen. More LoSs, more conflicts and more conflict chain reactions are expected. However, it was hypothesised that increasing traffic density would especially affect the performance of the SSD and coord methods. As more intruding aircraft are taken into consideration, it may be that these methods are unable to find a solution. In the SSD, if the VOs of all intruders occupy the complete solution space, no solution will be identified. In the coord method, more aircraft likely results in more iterations before a consensus is found. If the number of interactions exceeds the maximum number of iterations imposed, it will mean that aircraft do not reach a global solution.

Regarding safety, it was hypothesised that methods MVP and SSD would have fewer LoSs and fewer conflicts. The 'shortest-way-out' resolution strategy guarantees implicit coordination in pairwise conflicts, and minimal path deviations, which help limit conflict chain reactions. While in multi-actor conflicts this implicit coordination is no longer guaranteed, good results in previous research that used these methods [42, 152] indicate that this resolution strategy is still effective in multi-actor conflicts. In comparison, the coord method guarantees coordination in all cases. However, since each aircraft follows its own policy, it cannot be guaranteed that all aircraft resolution manoeuvres are optimal in terms of limiting the portion of airspace used. Finally, in the cost model, as LoSs with a low intrusion severity can be accepted in favour of not increasing flight path/time, it was hypothesised that it would have more LoSs than the other methods. Additionally, as a limited number of possible heading/speed manoeuvres are considered, it may be that an optimal manoeuvre for every conflict situation does not exist within the possible manoeuvres.

It was hypothesised that the cost method would have better efficiency, as its objective is to maximise the global efficiency. MVP and SSD methods also are expected to be efficient, as the resolution heading/speed employed represent the minimum deviation required to avoid LoS. When using the coord method, each aircraft tries to implement their optimal policy, which can be to either minimise flight path or flight time deviation. As a result, this method is not hypothesised to have the best flight distance of flight time efficiency, as not all aircraft work towards the same objective. For manned aviation, the

MVP and SSD methods which reduce the deviation from the nominal path, reducing the negative impact on flight distance, are expected to do less work.

Finally, stability wise, a higher DEP is expected for the coord and cost methods in comparison with MVP and SSD. The latter guarantee pairwise coordination based on the 'shortest-way-out' resolution strategy which is expected to benefit lower airspace area usage, reducing the amount of conflict chain reactions.

2.6. EXPERIMENTAL RESULTS

The effect of the independent variables on the dependent measures is presented in order to assess the effect of each conflict resolution method. Box-and-whisker plots are used to visualise the sample distribution over the several simulation repetitions. Efficiency, stability, and time in conflict values present outliers; the number of outliers (<10% of the total data) is consistent throughout. As these do not contribute to the comparison between the CR methods, these are not displayed for clarity.

2.6.1. SAFETY ANALYSIS

Figure 2.11 displays the mean total number of pairwise conflicts. A pairwise conflict is counted only once independently of its duration. The results for manned and unmanned aviation are comparable for each of the CR methods. The increase in number of conflicts, compared to the situation with CR-OFF, is due to secondary conflicts created by the tactical resolution manoeuvres. The number increases with the traffic density; with more aircraft it is progressively more difficult to avoid LoSs without triggering secondary conflicts. On average, as hypothesised, methods MVP and SSD display fewer secondary conflicts for both manned and unmanned aviation. These methods use the 'shortest-way-out' resolution strategy, limiting the space used by each aircraft, which limits conflict chain reactions. Within the two methods, the MVP method has more secondary conflicts than the SSD method, indicating that joint resolution to multi-actor conflicts is more efficient than pairwise resolution in limiting the number of secondary conflicts. Pairwise consideration of conflicts neglects constraints imposed by nearby aircraft not currently involved in the conflict. As a result, the chance of secondary conflicts is not considered in the calculation of a pairwise resolution. Additionally, contrary to hypothesised, the cost model has fewer conflicts than the coord method, although the difference between these two methods is negligible compared to the difference between them and MVP and SSD.

Figure 2.12 shows the amount of time spent in 'conflict mode' per aircraft. An aircraft enters 'conflict mode' when it adopts a new state computed by the CR method. The aircraft will exit this mode, once it is detected that it is past the previously calculated time to CPA (and no other conflict is expected between now and the look-ahead time). At this point, the aircraft will redirect its course to the next waypoint. The time to recovery is not included in total time in conflict. Based on this information and Figure 2.11, the number of conflicts is not directly correlated with the amount of time in conflict. For example, although the MVP method has a higher number of conflicts than SSD, it has a lower time in conflict. Time in conflicts for methods MVP, SSD, and cost are comparable. Method coord has the highest time in conflict, as well as a more pronounced tendency for the total time in conflict to increase with the traffic density. As the traffic density increases, there
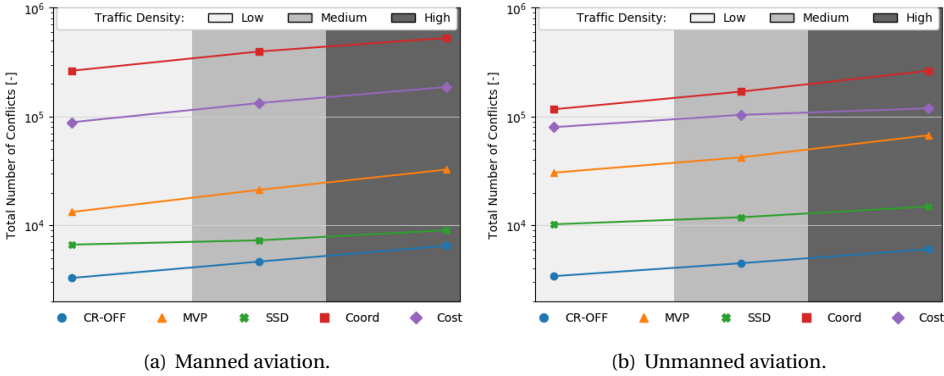
(a) Manned aviation.

(b) Unmanned aviation.

Figure 2.11: Total number of conflicts per CD&R method.

are potentially more situations where the break condition terminates the negotiation cycle before a global solution is found. A non-global solution will result in not all conflicts being resolved immediately, which in turn results in longer conflicts. Additionally, given that the coord method also has the highest number of conflicts (Figure 2.11), we can deduce that it has the highest tendency to create chain conflict reactions.

(a) Manned aviation.

(b) Unmanned aviation.
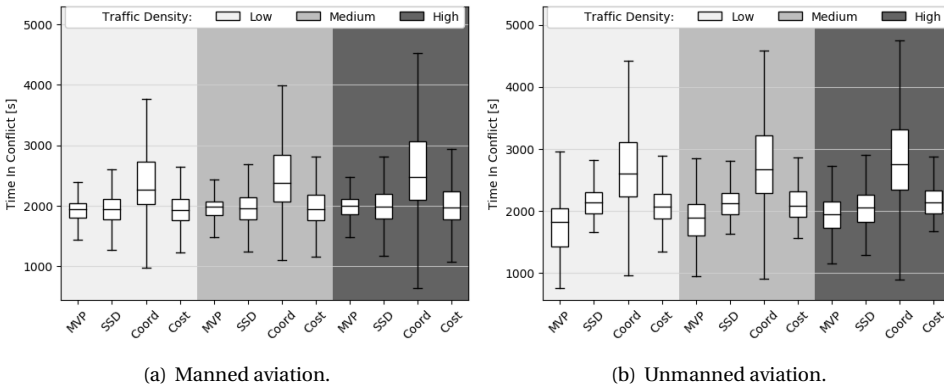
Figure 2.12: Time in conflict per flight and per CD&R method.

Figure 2.13 shows the mean total number of LoSs for each of the conditions. All methods significantly reduce the number of losses of separation, compared to the baseline condition where CR is OFF. MVP has the lowest number of LoSs in all traffic densities examined for both manned and unmanned aviation. Interestingly, a large number of conflicts (Figure 2.11) or time in conflict (Figure 2.12) does not directly result in a high number of LoSs. For example, the coord method has a high number of conflicts and time in conflict but few losses of separation. Thus, it should be considered that a large number of conflicts does not always have a negative impact on intrusions. In fact, Hoekstra [15] argues that a moderately positive number of secondary conflicts can be beneficial on a global scale; the effect of sequentially running into a new conflict creates a wave-like pattern, spreading the aircraft out in the available airspace thus 'creating' more airspace.

2

(a) Manned aviation.
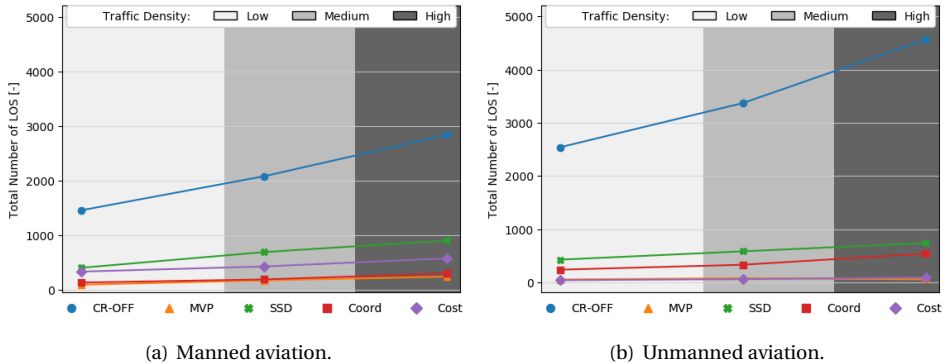
(b) Unmanned aviation.

Figure 2.13: Total number of losses of separation (LoSs) per CD&R method.

It was hypothesised that MVP and SSD methods woulds have the lower number of LoS. However, this is only true for MVP, which performs pairwise resolution; the SSD method, which performs joint resolution, has the highest number of LoSs of all tested CR methods. This is likely due to the fact that, as the traffic density increases, there are more situations when the solution space has no possible solution and thus, no manoeuvre is taken to avoid conflict situations. Additionally, it was hypothesised that the cost method would have a higher number of LoSs as low severity intrusions would be preferred over a significant deviation from either the nominal heading or nominal path. However, this is only true for manned aviation, whereas for unmanned aviation, the method has the lowest number of LoSs, alongside the MVP method. The cost calculation used displays a much better performance in the unmanned environment, proving that the weight coefficients should be adjusted and tested for the intended operational environment. Analogously, the coord method is better at reducing the number of LoSs in a manned environment than in an unmanned environment. In conclusion, when weights or policies are put in place, these should be aligned with the environment in which they are to be applied.

Figure 2.14 displays the intrusion severity for the losses of separation that occurred for each CR method. Although the overlap between conditions is large, MVP is most effective at minimising the intrusion when a loss of separation occurs. No direct correlation was observed between intrusion severity and traffic density for any of the methods.

2.6.2. STABILITY ANALYSIS

Figure 2.15 displays the mean DEP value for each CR method. A high positive value indicates the occurrence of conflict chain reactions that cause airspace instability. The coord method is the most unstable of all the CR methods, signifying that a resolution manoeuvre with this method is likely to trigger secondary conflicts. As seen in Figure 2.12, this model also has the highest time in conflict, resulting from longer negotiations or from a negotiation cycle ending without a global solution. When the start of a resolution manoeuvre is delayed, this alone can also lead to more conflicts. As hypothesised, MVP and SSD have the lowest DEP values. The 'shortest-way-out' strategy requires less airspace, reducing the number of conflict chain reactions.
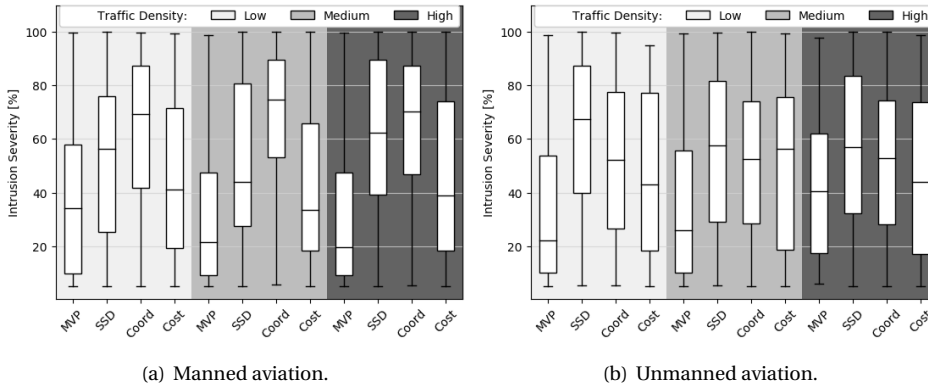
(a) Manned aviation.   (b) Unmanned aviation.

Figure 2.14: Intrusion severity rate per loss of separation and per CD&R method.

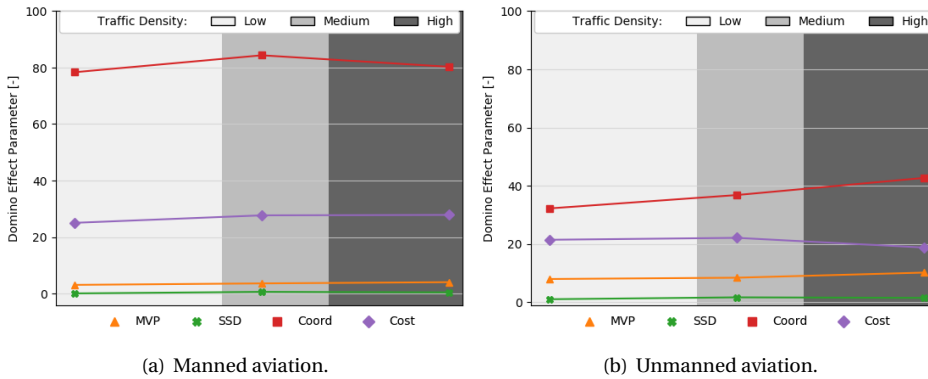(a) Manned aviation.   (b) Unmanned aviation.

Figure 2.15: Domino effect parameter (DEP) per CD&R method.

2.6.3. EFFICIENCY ANALYSIS

According to Figure 2.16, the MVP method results in the smallest path distance deviation. In other methods, either because aircraft perform longer deconflicting manoeuvres, or because they encounter more conflict situations which require a deviation from the nominal path, these travel for longer before reaching their destination. When assuming constant speed, increasing the flight path results in a longer flight. However, as seen in Figure 2.17, for manned aviation the SSD method has superior flight time compared with the coord method which has a larger flight distance variation. This indicates that the SSD method is favouring decreasing the speed of the aircraft as a deconflicting manoeuvre. MVP also has the smallest time deviation. It was hypothesised that the cost method would have better efficiency; however, overall, MVP and SSD methods proved more efficient. Having minimal path deviations for CR, reduced the effect of resolution manoeuvres on flight efficiency. The cost method has considerably better efficiency in the unmanned simulations versus the manned simulations, showing how cost calculations must be adjusted towards the characteristics of the environment. Finally, the coord method was significantly more efficient for manned aviation than for unmanned aviation, showing that the behaviour of specific policies is also highly dependent on the environment.
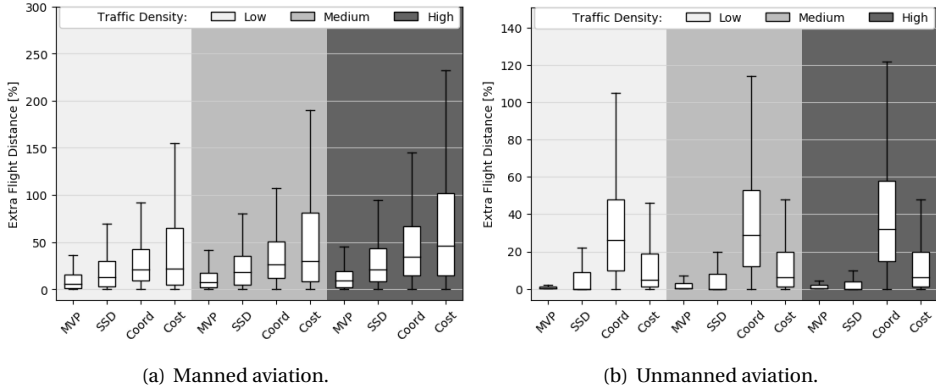
(a) Manned aviation. (b) Unmanned aviation.

Figure 2.16: Extra flight distance per flight and per CD&R method.

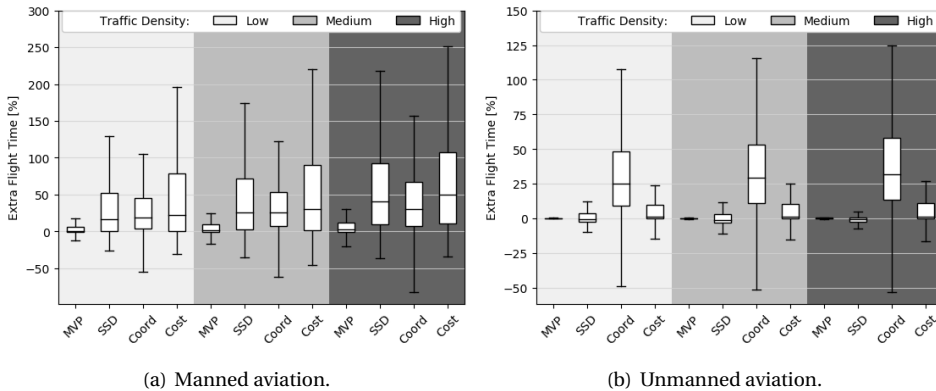(a) Manned aviation. (b) Unmanned aviation.

Figure 2.17: Extra flight time per flight and per CD&R method.

Figure 2.18 identifies the extra work done per flight performed by manned aviation. These values are directly comparable with the extra flight distance (Figure 2.16). The increase in work performed is a direct consequence of increasing the flight path due to conflict resolution manoeuvres. The MVP method has the smallest path deviation and, therefore, the smallest work increase. Note that the total work presented should not be used as exact absolute values as it is a generic relative indicator for fuel, which may be used for comparison.

2.7. DISCUSSION

2.7.1. EVALUATION OF CURRENT METHODS

From Tables 2.3 and 2.4, most current CR methods have tactical planning, distributed control, and focus on a nominal predictability assumption propagating the current state. Within manned aviation, there is no clear preference between centralised or decentralised control, whereas in unmanned aviation most of the models resort to decentralised control as there is no defined central processing point for unmanned aviation yet. A considerable
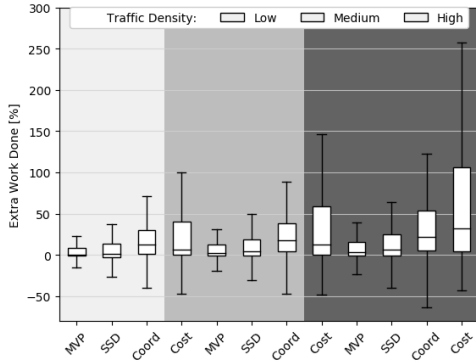
Figure 2.18: Extra work done per flight and per CD&R method for manned aviation.

number of methods for unmanned aviation still focus solely on static obstacles, indicating that further development is still required for beyond visual line of sight (BVLOS) unmanned operations, where avoidance of other traffic is required. Given the increasing use of drones in applications such as package delivery in an urban setting, with traffic densities that are orders of magnitude higher than any observed in manned aviation, the development of CD&R methods for unmanned aviation capable of avoiding both static and dynamic obstacles is a pressing issue.

2.7.2. COMPARISON OF CONFLICT RESOLUTION METHODS

The results displayed no significant disparity in terms of which type of CD&R method performs better between a manned and an unmanned environment. However, the differences in unmanned over manned aviation heavily favour the performance of the methods; lower speed of the involved aircraft and smaller minimum separation distance favour the prevention of LoSs. For the characteristics of the experiment performed, MVP and SSD methods showed better results overall, in particular the MVP method. Having minimum path deviations for CR, reduced the effect of resolution manoeuvres on flight efficiency while still guaranteeing minimal LoSs. At high densities, tactical conflict resolutions can trigger conflict chain reactions due to the scarcity of airspace [153].

Two different reactive methods were used, MVP and SSD, in order to directly compare the behaviour of pairwise-summed and joint resolution approaches. The latter is better at conflict prevention, showing a lower number of conflicts (Figure 2.11). While pairwise-summed methods like the MVP do tend to trigger more secondary conflicts during resolution in high-density traffic situations, the net result of this is often beneficial. The emergent behaviour of the traffic situation as a whole shows that these secondary conflicts are 'used' by the algorithm to distribute traffic, and 'create room' for resolutions that would otherwise not be apparent. As a result, its performance in terms of LoSs is superior compared to a joint-resolution method. In addition it can be noted that other works [31, 32] have shown, using the MVP method, that disallowing aircraft from turning into a conflict can help mitigate the number of secondary conflicts. While the SSD method has the lowest number of conflicts, it has the highest number of LoSs from the four simulated CR methods (Figure 2.13). When using the SSD method, having more surrounding

aircraft will likely result in fewer solutions within the solution space. In extreme cases, a single joint solution may not even exist. As a result, the behaviour of a joint resolution CR method should be carefully considered when used in high traffic density environments. Additionally, when comparing the number of conflicts (Figure 2.11) with the number of LoSs (Figure 2.13), it cannot be inferred that preventing secondary conflicts is always the best way to prevent LoSs, as there is no direct correlation between these two values. Indeed it can be argued that, for some situations, not moving towards solving all conflicts immediately may be beneficial; due to scattering traffic, further away conflicts may be easier to resolve later on. Additionally, a joint resolution manoeuvre often results in a larger path deviation, which has a negative impact on the stability of the airspace.

The cost and coord methods showed differences in terms of safety and efficiency performance between the unmanned and manned environments. The former was better at preventing losses of separation and was more efficient in an unmanned aviation environment, whereas the latter had better efficiency and better success at preventing intrusions in a manned aviation environment. This proves that the success of weight coefficients and employed policies is dependent on the operational environment. Naturally the optimal heading/speed deviations to avoid losses of minimum separation depend heavily on the speeds and manoeuvring space between neighbouring aircraft. Having weight coefficients or policies which enforce the optimal resolution manoeuvres is beneficially to safety. On the other hand, restraining aircraft from employing these optimal choices, when these differ from their preferred policies, may have a negative impact on the overall safety of the airspace. In comparison, methods MVP and SSD were not so sensitive to the differences between manned/unmanned environments.

2.7.3. OPEN AND COMMON SIMULATION PLATFORMS

This results should be considered alongside the results produced by other researchers in simulations environments with different conditions. For Piedade [152], who used BlueSky for manned aviation with different scenarios and smaller traffic densities (from 9 ac/10,000 NM^2 to 27 ac/10,000 NM^2), similarly to the results herein obtained, the MVP method showed fewer losses of separation than the SSD model. In their implementations, Yang [45] was able to guarantee safe separation of 48 UAVs in a space of 22 NM^2 and Hao [85] showed no LoSs for five manned aircraft in a 4400 NM^2 scenario. These results should also be taken into account when considering the performance of these methods. However, it is impossible to directly compare these results from other researchers alone given the differences in scenarios and traffic densities. Moreover, it is difficult to extrapolate these results beyond the specific environment conditions and employed traffic densities. Such shows the importance of creating repeatable evaluation conditions, by using open platforms, and publicly sharing implementations, metrics, and data. Developing an open repository of reference simulation scenarios would allow for direct performance comparison and a more precise evaluation of the diverse proposed methods.

2.7.4. IMPACT OF IMPLEMENTATION CHARACTERISTICS

Implementation characteristics, such as cost-function gains, can significantly affect the outcome of an evaluation. As previously mentioned, we observed that the overall efficiency of the simulated CD&R methods involving either policies or cost functions,

was highly dependent on the environment. It may be considered that further tuning of these policies/weights could improve the overall safety of the method, or even that in a different environment these methods would have significantly different performance. As a result, the several tuning options in CD&R methods should be carefully adjusted to the operating environment. Furthermore, several implementation criteria affect the output of the same algorithm. Some of these criteria have been mentioned in this work: update rate, performance, types of resolution manoeuvre, turn frequency. Naturally, any limitation on these properties is expected to deteriorate the performance of the model.

In the simulations herein performed, similar implementation characteristics were used for all CD&R methods, to the extent possible given the differences in the algorithms. We intended not only to provide a first approach at a direct comparison, but also to emphasise how results are conditioned by implementation settings, which are often overlooked. These settings should be directly associated with the results, with the understanding that different tuning values, policies, weights, and environments can yield different evaluations of the same algorithm.

2.7.5. IMPACT OF SIMULATION PROPERTIES
Fast-time simulations are often used to provide insights on the advantages and disadvantages of conflict resolution strategies. However, it can be time consuming to develop a simulation environment to a high level of realism. It is relevant to make clear assumptions regarding speed, altitude, and spatial distributions of the aircraft. Sunil [154] researched how these assumptions affect the conflict outcome; non-ideal altitude and spatial distributions have the largest negative impact on the accuracy of the simulation results. It is necessary to guarantee an uniform density distribution to prevent traffic concentrations. A density 'hotspot', either vertically or horizontally, results in a higher number of conflicts relative to the ideal case, providing the wrong insights on the overall safety.

2.8. CONCLUSIONS
More than a hundred conflict resolution (CR) methods for manned and unmanned aviation were evaluated under a taxonomy based on avoidance planning, surveillance, control, trajectory propagation, predictability assumption, resolution manoeuvre, multi-actor conflict resolution, obstacle types, optimisation, and method category. Currently, most models involve tactical planning, distributed control, and focus on a nominal propagation of the current state of all involved aircraft. For unmanned aviation, more CR methods must be developed focusing on assuring minimum separation with both static and dynamic obstacles to aid beyond visual line of sight operations in an urban setting.

Furthermore, commonly used CR methods were analysed using open-source, multi-agent ATC simulation tool BlueSky [25], both for manned and unmanned aviation. The differences between the results here presented and previous research show the importance of creating repeatable evaluation conditions, by using open platforms, and publicly sharing implementations, metrics, and data. CD&R methods aim at relieving the workload of ATC services and assuring safe integration of UAVs into the civil airspace. However, a better notion of how current methods behave for specific traffic scenarios is essential in order to determine a way forward for improvement.

PART I:
CONFLICT DETECTION & RESOLUTION IN A CONSTRAINED ENVIRONMENT

3

VELOCITY OBSTACLE BASED CONFLICT RESOLUTION IN URBAN ENVIRONMENT WITH VARIABLE SPEED LIMIT

The results of the Chapter 2 indicate that airspace concepts which reduce the average relative velocities between aircraft, and opt for the 'shortest-way-out' resolution, improve airspace safety. In this chapter, we apply one of these methods, the Solution Space Diagram (SSD), to an urban environment. The structure of the airspace plays a (positive) role in the capacity of the airspace. However, the use of airspace design to optimise distributed environments has been overlooked in previous research. Moreover, for unmanned aviation, the structure in place must respect the boundaries of the surrounding urban infrastructure.

In Section 3.2, we examine how to reduce the conflict rate by separating traffic into different layers according to heading-altitude rules. In Section 3.4, we use a reinforcement learning agent to implement variable speed limits to create a more homogeneous traffic situation between cruising and climbing/descending aircraft.

Cover-to-cover readers can choose to skip sections 3.3.1 and 3.3.2, which describe the theoretical background of the SSD method. This method has also been previously explained in Chapter 2.

This chapter is based on the following publications:

1. M. Ribeiro, J. Ellerbroek, and J. Hoekstra, Velocity Obstacle Based Conflict Avoidance in Urban Environment with Variable Speed Limit, Aerospace 8 (2021)
2. M. Ribeiro, J. Ellerbroek, and J. Hoekstra, The Effect of Intent on Conflict Detection and Resolution at High Traffic Densities, 9th International Conference for Research in Air Transportation (ICRAT) (2020)

3

ABSTRACT

Current research on urban aerial mobility, as well as the continuing growth of global air transportation, has renewed interest in conflict detection and resolution (CD&R) methods. The use of drones for applications such as package delivery would result in traffic densities that are orders of magnitude higher than those currently observed in manned aviation. Such densities not only make automated conflict detection and resolution a necessity, but it will also force a reevaluation of aspects such as coordination vs. priority, or state vs. intent. This chapter looks at enabling a safe introduction of drones into urban airspace by setting travelling rules in the operating airspace, which benefit tactical conflict resolution. First, conflicts resulting from changes in direction are added to conflict resolution with intent trajectory propagation. Second, the likelihood that aircraft with opposing headings meet in conflict is reduced by separating traffic into different layers per heading-altitude rules. Guidelines are set in place to ensure that aircraft respect the heading ranges allowed in every crossed layer. Finally, we use a reinforcement learning agent to implement variable speed limits towards creating a more homogeneous traffic situation between cruising and climbing/descending aircraft. The effect of all of these variables was tested through fast-time simulations on an open-source airspace simulation platform. The results show that we are able to improve the operational safety of several scenarios.

3.1. INTRODUCTION

If current predictions become reality, the aviation domain must prepare for the introduction of a large number of mass-market drones. According to the European Drones Outlook Study [10], approximately 7 million leisure consumer drones are expected to operate across Europe, and a fleet of 400K is expected to be used for commercial and government missions in 2050. Moreover, at least 150K are expected to operate in an urban environment for multiple delivery purposes. More recently, even more urban Unmanned Aerial System (UAS) applications have been explored, specifically inspection and monitoring of several urban infrastructures [155, 156]. Automation of safety within unmanned aviation is a priority, as drones must be capable of conflict detection and resolution (CD&R) without human intervention. Both the Federal Aviation Administration (FAA) and the International Civil Aviation Organisation (ICAO) have ruled that an UAS must have Sense & Avoid capability in order to be allowed in civil airspace [11, 26]. Over the past three decades, conflict detection and resolution methods have already been widely explored for manned aviation. However, there are several aspects that separate the urban applications currently considered from the concepts investigated in these previous studies. The most consequential difference from conventional aviation is the presence of constraints in an urban environment, such as obstacles and hyperlocal weather, which will bring additional considerations in the design of CD&R logic.

Although these differences separate urban air traffic from conventional aviation, they provide several similarities with the operation of road traffic that make it relevant to investigate research to prevent traffic congestion of road vehicles [157, 158]. First, in many of the current urban airspace concepts, unmanned aviation is expected to follow the existing road infrastructure. In addition, prevention of congestion is comparable to prevention of 'hotspots' of conflicts. Finally, collisions are reduced by guaranteeing

at all times a safe distance between road vehicles, comparable to the safekeeping of the minimum separation distance in aviation. Nevertheless, directly applying these methods poses new challenges: drones are (mostly) non-stationary as opposed to road vehicles, minimum separation is a bigger margin than normally employed with road vehicles. Additionally, we prefer not to employ prevention of traffic 'hotspots' through path planning, which increases in complexity with the number of operating agents. In a real-word scenario, with the expected number of UASs operating simultaneously [22], this would result in a system that is slow to respond to changes, as well as with limited capacity [159]. Instead, we focus on setting rules directly into the operational environment to guarantee safety.

In the current study we employ an urban environment where aircraft must go throw pre-set 'delivery points' simulating a delivery operation. Conflicts with static obstacles are immediately resolved by following a planned route around these obstacles. Conflict resolution (CR) is used to further prevent losses of minimum separation with dynamic obstacles. Normally, most conflict detection and resolution (CD&R) methods use heading changes as preferred by air traffic controllers. However, an urban environment requires a different approach to an unconstrained airspace. We favour a speed-based conflict resolution approach to ensure that the borders of the surrounding urban infrastructure are always respected. Heading-altitude rules will be used to separate traffic into different layers, reducing the likelihood of aircraft meeting in conflict. Additionally, we add intent-information to conflict resolution. Multiple works [27–30] have used waypoint information to improve the prediction of a single intruder's trajectory with favourable results. Given the high number of turns required to move through an urban setting, studies on the use of intent are of interest. Naturally, sharing intent information in a real-case scenario requires a mechanism for data transfer between aircraft or intent inference through trajectory prediction [160]. Both are a challenging problem. This work will analyse whether the improvements in safety from the addition of intent information justify its implementation. Finally, reinforcement learning is used to set variable speed limits (VSLs) in sections where altitude transitions are expected, towards creating a more homogeneous traffic situation during these transition phases.

Section 3.2 defines the urban environment. Sections 3.3 and 3.4 can be read interchangeably. The former describes how aircraft avoid conflicts by modifying their current speed. We use a velocity obstacle based CR approach (called Solution Space Diagram (SSD) in related work [43, 94, 161, 162]), which has proven to be efficient in reducing the effect of resolution manoeuvres on flight efficiency while still guaranteeing minimal losses of separation (LoSs) [162]. Section 3.4 refers to VSL implementation. As shown in Figure 3.1, this sets an upper limit to the speeds aircraft may select from. The Deep Deterministic Policy Gradient (DDPG) RL method [163], which has shown promising results in other studies [164], is used to determine the optimal variable speed limits. Sections 3.5 through 3.8 describe the experimental independent variables, design, hypotheses, and results, respectively. Finally, Chapters 3.9 and 3.10 present discussions and conclusion. This study employs the open-source, multi-agent ATC simulation tool BlueSky [25]. The implementation code can be accessed online at [165]; scenarios, result files are available at [166].
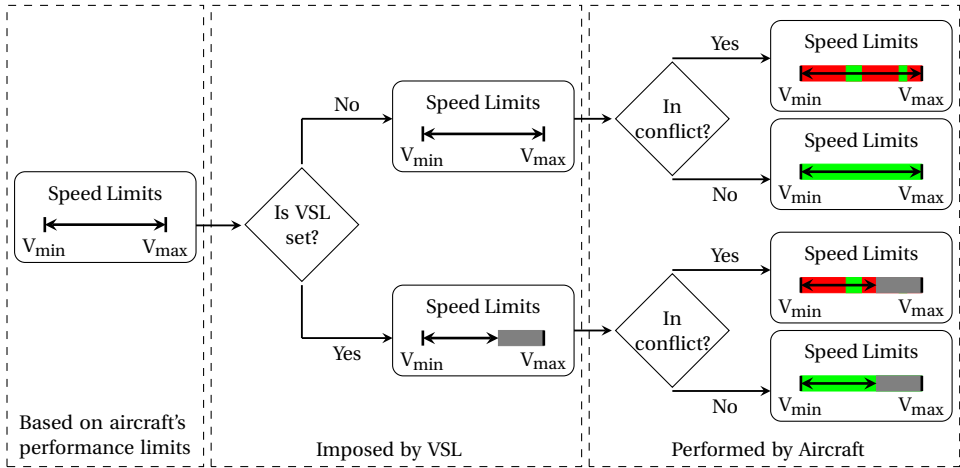
3

Figure 3.1: Prioritisation of rules over speed choice. Hard limits are first imposed by aircraft's performance limits. If set, the maximum speed must be respected. Additionally, aircraft perform conflict resolution. A conflict-free (displayed in green), allowed speed value is then picked.

3.2. URBAN SETTING

In this work, an urban setting is simulated using data from the Open Street Map network data [167]. We use an excerpt from the San Francisco Area, with a total area of $1.708\,\mathrm{NM}^2$, as represented in Figure 3.2. In the dataset, roads and intersections are represented by nodes. Each road is defined by the two adjacent nodes that represent the edges of the road. To reduce complexity, each node is considered to have at most four connecting roads. Naturally, some nodes may have fewer as only existing roads are used. Additionally, we assume that each road has only one lane. Having more lanes would signify that the road would need to be large enough to ensure proper separation between multiple lanes. As we do not make such assumptions or requirements from the urban setting, we define each road as having only one lane of traffic.
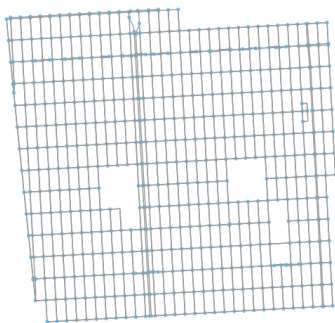

Figure 3.2: Urban setting used in this work. Data obtained from Open Street Map [167].

3.2.1. FREEDOM OF MOVEMENT

The exploration of environments with static obstacles has gained a new focus with the growth of unmanned aviation. Operations such as package delivery in an urban environment require collision resolution with the surrounding urban infrastructure. The latter is non-trivial. Most of the existing research on tactical conflict detection and resolution is directed at manned aviation, as methods are used to detect other dynamic traffic when manned aircraft are flying at cruise altitude. It is not guaranteed that a method directed at dynamic obstacles can also (simultaneously) avoid static obstacles. First, while most of these methods assume obstacles as a circle with radius equal to the minimum separation distance, a static object can have different sizes and shapes. These may be larger than other traffic and non-convex, requiring a route with multiple waypoints as solution. Second, most methods also assume some sort of coordination and non-zero speed.

Limited existing research on the resolution of tactical conflicts with static obstacles is based mainly on defining static obstacles as objects that the ownship must go around, as opposed to those that limit the area accessible to the ownship [168]. Recently, a new branch of research is integrating LIDAR technology into UASs to detect the distance from the closest obstacles [169, 170]. However, such systems do not protect against static obstacles with non-uniform shapes. For example, an aircraft might follow the edge of a static obstacle until it finds itself in a dead-end, in case this edge ends in a closed space. We consider that when the environment is known in advance, the most effective way to resolve conflicts with static obstacles is to strictly follow a known safe route around all static obstacles. This work assumes that waypoints are set at the centre of the roads, from which aircraft do not deviate.

3.2.2. TURN ESTIMATION

In an urban environment, the speed at which the aircraft turns is limited by the radius of the turn, as collisions with buildings must be avoided within the limited space available at intersections. The same conservative value is used for all aircraft. Naturally, in a real-case scenario, differences in turn performance can be expected between rotors and fixed-wing aircraft. Rotors may be able to hover in a stationary position and provide (almost) vertical take-off and landing.

We assume that during turns, aircraft remain at the same flight level and have constant speed throughout. In Figure 3.3, the waypoints of the aircraft are identified. As the heading post-waypoint$_{i+1}$, Ψ_{i+1}, is different than the current heading, Ψ_i, the aircraft initiates a turn assumed to start and end at a pre-determined distance, d, from waypoint$_{i+1}$.
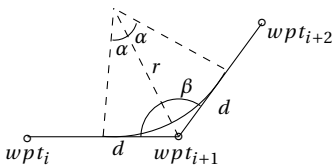

Figure 3.3: Geometry of a turn between waypoints. No wind assumed.

The radius of the turn, r, can be calculated by :

$$r = \frac{V^2}{g \times \tan(\phi_{nom})}, \tag{3.1}$$

where V represents the speed of the aircraft, and g the gravitational acceleration. Based on the geometry of Figure 3.3:

$$\alpha = \frac{\Delta\Psi}{2}. \tag{3.2}$$

The distance from the waypoint$_{i+1}$ at which the aircraft starts and ends the turn is thus given by:

$$d = r \times \tan(\alpha). \tag{3.3}$$

The turn rate, $\dot{\Psi}$, can be determined by:

$$\dot{\Psi} = \frac{g \tan(\phi_{nom})}{V}. \tag{3.4}$$

3.2.3. SPEED CHANGES THROUGHOUT THE ROUTE

We assume that aircraft prefer to adopt a high speed in order to reduce travel time, and complete their delivery route as soon as possible. However, due to the limitation imposed on the turn radius, the aircraft will reduce their speed prior to a turn to fit the confined space of the intersection. Figure 3.4 shows the assumed behaviour of the aircraft during the experimental simulations. When possible, aircraft will employ the maximum set cruise speed of 30 kts. Before a turn, the aircraft will start decreasing their speed, in order to start the turn at 10 kts. With such a low speed, it is guaranteed that the maximum turn radius of 3 metres is respected. As soon as the turn is completed, aircraft will again accelerate towards their desired cruising speed.
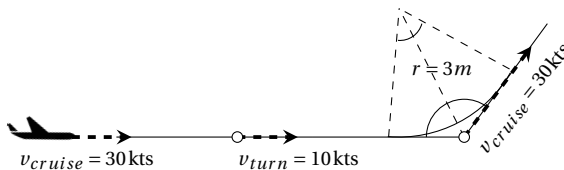

Figure 3.4: Speed changes employed by an aircraft in preparation for a turn.

These speed variations result in speed heterogeneity between aircraft, which is recognised as a causal factor for increased complexity in air traffic operations [171]. Part of the work in this chapter aims at reducing relative speeds, which is expected to improve safety.

3.2.4. HEADING-ALTITUDE RULES

(Near-)Head-on conflicts are practically impossible to resolve in a restricted airspace where aircraft cannot considerably alter their heading. The best way to prevent this situation is to separate aircraft into different layers according to their current heading, creating a more homogeneous traffic situation in each layer. Similar concepts were used in [13, 172–174]. The results showed that vertical segmentation of airspace, by separating

traffic with different travel directions into different flight levels, resulted in a lower conflict rate and thus enabled higher capacity. Two factors contributed to this reduction in conflict rate. First, by dividing the aircraft over separate layers of airspace, different groups of aircraft are created that remain separated from each other (segmentation effect). Second, within each layer, heading limitations enforce a degree of alignment between aircraft, thus reducing the relative speed between aircraft cruising at the same altitude, which in turn reduces the likelihood of conflicts within a layer of airspace (alignment effect) [175].

In this work, six altitude (traffic) layers are used as shown in Table 3.1. Heading-altitude rules are applied, defining the headings permitted per altitude band. It is assumed that each node has a maximum of four connecting edges. On each of these edges, traffic is assumed to have (near) equal headings. Therefore, we start by adopting one vertical layer for each possible direction, creating the four main traffic layers. In addition, two auxiliary layers are employed to allow aircraft, travelling in a main layer, to cross to a perpendicular road in any direction just by climbing or descending to the next layer. Given the defined layers, a heading turn will result in a transition of a maximum of three layers (i.e., when climbing from the 1st to the 4th layer or descending from the 6th to the 3rd layer).

Table 3.1: Quadrant rules per altitude layer.

1st Layer	2nd Layer	3rd Layer	4th Layer	5th Layer	6th Layer
Auxiliary Layer	Main Layers				Auxiliary Layer

Altitude

To move to a different layer, aircraft climb or descend into the traffic lane of that layer. Previous work [13] suffers from a considerable number of conflicts between cruising and climbing/descending aircraft, and between pairs of climbing/descending aircraft, as climbing and descending aircraft are exempted from the heading-altitude rules, and can violate them to reach their cruising altitude or destination. This means that aircraft are free to directly climb/descend to the final layer without respecting the heading ranges allowed in the mid layers. In these cases, the safety benefits of vertical layer separation apply only to cruising aircraft, as there are no procedural mechanisms to separate climbing/descending aircraft from each other or from cruising aircraft [175]. In this study, we add to this work by implementing rules during the climbing/descending process. First, during climb/descent, aircraft must adapt to the heading ranges allowed at each layer traversed. Second, aircraft are still restricted to a safe route through the surrounding urban infrastructure. Finally, we employ variable speed control to improve speed homogeneity between cruising and climbing/descending aircraft.

TRANSITION LAYERS

We employ transition layers to accommodate traffic slowing down before a turn. A transition layer is set between two traffic layers to be used only when transitioning between

the latter. Aircraft perform the heading turns within these transition layers, preventing conflicts resulting from heterogeneous speed situations caused by an aircraft decelerating in preparation for a turn. Naturally, conflicts can still occur in the transition layers. However, transition layers are expected to have a much smaller number of aircraft than traffic layers at any point in time, reducing the likelihood that aircraft meet in conflict.

Figure 3.5 shows the different layers used in the experimental simulations. Traffic layers (in blue) are used for the cruising traffic; transition layers (in grey) are only used for transitioning between traffic layers. Traffic and transition altitudes are set with a height of 30 ft. Note that there is an offset of 10 ft between the layers to prevent false conflicts.
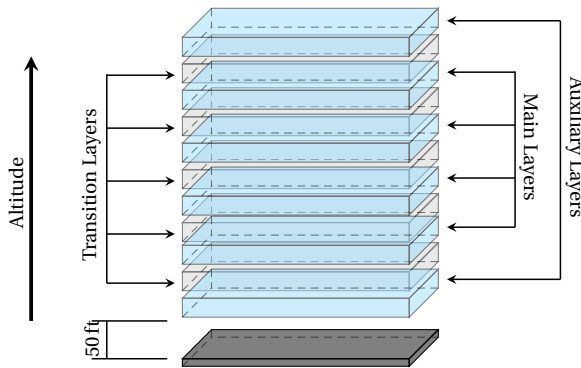
Figure 3.5: View of the different altitude layers used in the experimental simulations performed in this study.

Finally, turn mechanics are in place to enforce that aircraft perform the necessary climb/descent actions without crossing the borders of the surrounding urban infrastructure and/or violating the heading ranges allowed per traffic layer. Independently of the flight altitude, aircraft must respect the surrounding infrastructure as we make no assumptions regarding its height. As a result, this mechanism may be used independently of the maximum height of the urban architecture, the number of traffic layers, and/or altitude of each layer.

3.3. VELOCITY OBSTACLE BASED, SPEED-ONLY RESOLUTION

The biggest obstacle to ensuring minimum separation between aircraft in an urban environment is the limitation of movements caused by the limited available space. Most conflict prevention methods operate in the horizontal plane, and rely on turns to resolve conflicts. However, to guarantee safety in the presence of static obstacles (e.g., buildings, trees), movement within the horizontal plane is severely limited. This work employs a speed-only conflict resolution method, guaranteeing that aircraft do not deviate from their safe pre-set route. Vertical conflict resolution is not used as the available airspace is segmented into different flight levels reserved for different flight directions. For increased safety, aircraft must remain at their assigned flight level. Although variations in this vertical layer assignment are possible, these are considered outside the scope of this work.

3.3.1. Velocity Obstacle (VO) Theory

The conflict resolution method used in this work is based on the velocity obstacle theory [148, 149]. In Figure 3.6, a situation is represented in which the ownship (A) is in conflict with an intruder (B). A collision cone (CC) can be defined by lines tangential to the intruder's protected zone (PZ). A and B are in conflict when the relative velocity between these two aircraft lies inside the CC. By adding the intruder's velocity, the CC is translated forming the intruder's VO. This VO represents the set of ownship velocities which result in a loss of separation with the intruder. R represents the radius of the PZ. $P_A(t_0)$ and $P_B(t_0)$ denote the ownship's and the intruder's initial position, respectively. $P_B(t_c)$ identifies the position of the intruder at the moment of collision. Each intruder in the vicinity of an ownship results in a separate VO.
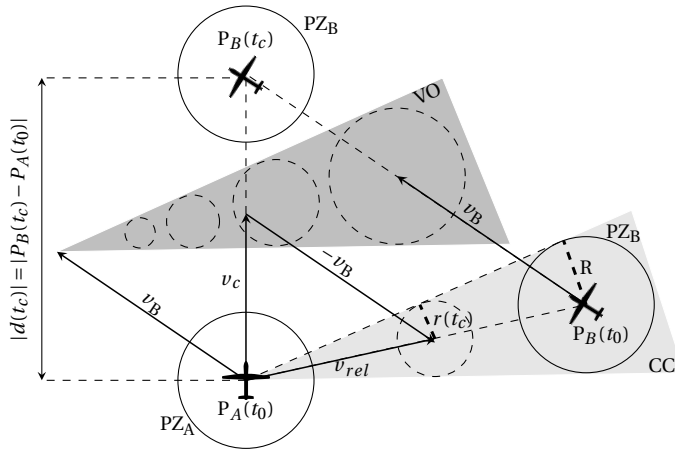

Figure 3.6: Representation of a VO imposed by intruder B, and the relationship between a circular velocity vector set and the protected zone (PZ) [94].

3.3.2. Solution Space Diagram (SSD) Resolution Method

The SSD method consists of finding the intersection between the VOs from all intruders and the performance limits of the ownship, in order to identify which sets of achievable velocity vectors result in a future LoS with intruders. Two concentric circles, representing the minimum and maximum velocities of an aircraft, bound all reachable speed vectors. Within this reachable velocity space, VOs are constructed for each proximate aircraft, each representing the set of speed vectors that would result in a conflict with the respective aircraft. When all relevant VOs are subtracted from the set of reachable velocities, what remains is the set of reachable, conflict-free speed vectors. Then a new advised speed vector is picked from this set and used for conflict resolution. Thus, SSD is able to solve multiple conflicts simultaneously. In two-aircraft situations, this method is implicitly coordinated, as the conflict geometry, represented by the velocity obstacle, can be used to select complementary measures to avoid each other.

The algorithm used is the Solution Space Diagram method as implemented by Balasooriyan [42]. Identification of a conflict-free resolution vector, consists of finding a point inside the set of spaces within the velocity limits that do not intersect the VOs [150].

3.3.3. CONFLICT RESOLUTION WITH SPEED VARIATION

This work employs speed-only conflict resolution with the SSD method. For reference, Figure 3.7 depicts the selection of a speed vector for conflict resolution that does not alter the heading of the aircraft; only the speed is altered. Note that the conflict-free speed vector that results in the smallest speed change is selected to resolve the conflict.
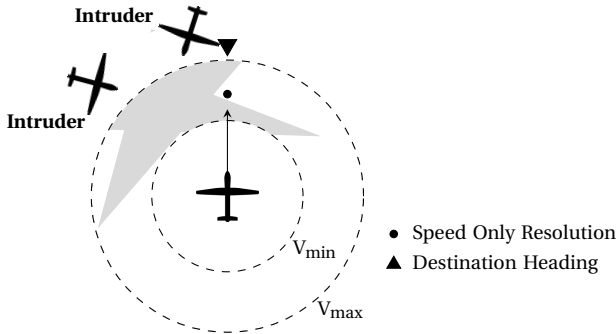

Figure 3.7: Representation of speed only based conflict resolution using the SSD method.

Speed-only resolution has previously been explored with flight-level assignments in [22, 61, 62, 176]. The results show that speed-only conflict resolution is only successful when aircraft in conflict have similar headings. For example, (near-)head-on conflicts require heading variations; a speed change is not sufficient to guarantee minimum separation. The likelihood of the latter kind of conflicts is dependent on the airspace structure and the heading difference between aircraft flying at similar flight levels. The introduction of heading-altitude rules is expected to favour the efficacy of this SSD method. First, (near-)head-on conflicts during cruising phase are no longer expected as, in each altitude layer, aircraft have similar headings. Second, when using SSD for speed resolution, having more surrounding aircraft will likely result in fewer solutions within the solution space. In extreme cases, a single joint solution may not even exist. As a result, the behaviour of the SDD method is severely hindered on a high traffic density layer. Dividing all traffic into several layers is likely to reduce the saturation of the solution space.

3.3.4. STATE-BASED VS INTENT-BASED RESOLUTION

Most tactical conflict resolution methods rely on nominal state-based extrapolations to determine the closest point of approach (CPA) between aircraft. State-based methods assume a projection based on the aircraft's current position and velocity vector. However, when future trajectory changes of all aircraft involved are not taken into account, false alarms may occur, and future LoSs may be overlooked. A state-based method can only adapt to a heading change once the aircraft completes the change and the new heading is the new state. A method which employs intent trajectory prediction can compute this future heading change before it starts, and therefore prevent last minute risk prone situations resulting from the change. Given the high number of turns necessary within an urban setting, research on the use of intent information in this environment is relevant.

Intent is commonly used in multi-agent coordination to improve safety [177]. For

example, in road vehicles, light signaling is used to indicate an imminent turn. With aircraft, explicit intent sharing is not so trivial. The future trajectory is defined by connecting future trajectory change points (TCPs), which must be shared and processed by other aircraft. As a result, only aircrafts that have sufficient technology to transmit and handle this data without considerable delay have access to the airspace. The complete TCP plan may be shared with one data transmission, reducing the number of necessary data exchanges. However, uncertainties increase throughout flight time as aircraft progressively deviate from their nominal intent to avoid conflicts. Another option is to share future TCPs up to a predefined look-ahead time. This is done in this work; we consider future TCPs up to the conflict detection look-ahead time to be known by all aircraft.

Nevertheless, state information can never be completely removed from the computation as, for imminent LoSs, it is often preferable to minimise the state change ('shortest-way-out' principle) than to follow the nominal intent. There are situations where considering propagation of both state and intent information result in non-intersection trajectories (e.g., near an almost reverse turn). In cases where considering both possibilities results in no available conflict-free solutions, one may have to be prioritised. Thus, the combination of state and intent information, and when to prioritise one of these, must be accounted for in advance. Speed-only conflict resolution, as used in this work, has the advantage of not moving aircraft away from their TCPs. However, it can delay or advance its crossing. Finally, the use of TCP points may limit conflict resolution coordination. Aircraft may be expected to move towards their next TCP instead of taking opposite directions to avoid each other. As a result, safety improvements resulting directly from the use of intent must always be considered in conjunction with the expense of its implementation.

Intent information can be added to the VOs considered in the SSD based on the work of Velasco [94]. Such will alter their shape, thus resulting in a different set of velocity vectors which do not intersect the intruders' VOs (see Figure 3.8). This section shows how a VO can be built with intent information.

(1) Using state information
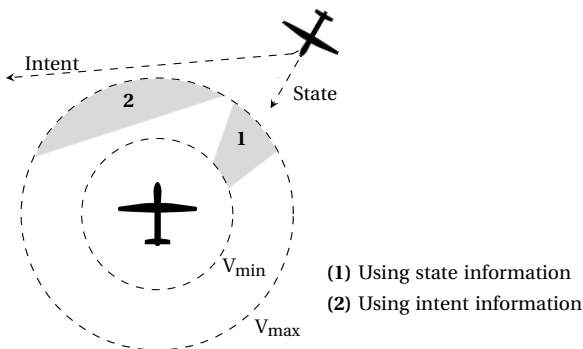**(2)** Using intent information

Figure 3.8: Shape of the VO depending on whether state or intent information is used to propagate the current trajectory of the intruder into the future.

The velocity, v_c, which will make the ownship occupy the same position as the intruder at a given time, t_c, is equal to:

$$v_c(P_A(t_c) = P_B(t_c)) = \frac{P_B(t_c) - P_A(t_0)}{t_c - t_0} = \frac{d(t_c)}{t_c - t_0}, \tag{3.5}$$

where $d_c(t_c)$ represents the distance the ownship aircraft must travel in order to collide with the intruder at time t_c. In theory, the VO of an intruder can be built from $t_c = t_0$ to $t_c \rightarrow \infty$. For each t_c, the distance $d(t_c)$ that the ownship would have to travel, and the necessary velocity to do so within $t_c - t_0$, can be identified. As $|v_c|$ increases, t_c decreases from $t_c \rightarrow \infty$ to $t_c = t_0$. However, in practise, the upper limit of the VO is set as the look-ahead time value for conflict detection. Given the symmetrical relationship between the radius of the circular set of velocities r and the radius of the protected zone R (see Figure 3.6), the former can be determined:

$$\frac{r(t_c)}{|v_c(t_c)|} = \frac{R}{d(t_c)}. \tag{3.6}$$

Given Equation 3.5, Equation 3.6 can be transformed into:

$$r(t_c) = \frac{R}{t_c - t_0}. \tag{3.7}$$

For each time to collision, t_c, a new VO circle can be calculated based on the predicted heading, velocity, and acceleration of the intruder at that time. The VO will then be formed by connecting these circles (Figure 3.9). For a VO without intent, lines connecting all the circles in the VO are straight, maintaining the same direction and size progression throughout time. However, when considering intent, circles do not follow the same progression.
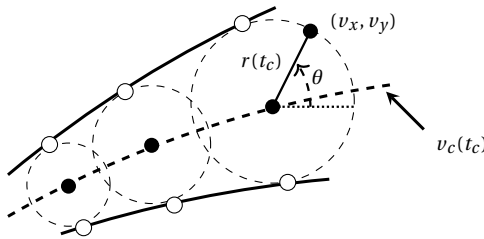

Figure 3.9: VO built with intent information. The VO circles are centered at $v_c(tc)$.

Considering that time can be expressed along the bisector of the VO, the VO itself can be identified as a family of circular curves, with their centre at $v_c(tc)$ along the VO bisector. The envelope of a family of curves is defined as [178]:

$$\begin{bmatrix} v_x \\ v_y \end{bmatrix} = v_c(t_c) + r_c(t_c) \begin{bmatrix} cos(\theta) \\ sin(\theta) \end{bmatrix}, \quad \forall \; \theta \in [-\pi, \pi], \quad t_c \in [t_c, \infty], \tag{3.8}$$

where v_x, v_y are the components of the velocity vector for each VO circle, and θ the angular coordinate. Deriving the envelope equation will result in the values of θ for which v_x, v_y are the tangent points on the envelope curve.

By assuming that the collision vectors are differentiable, the envelope of the family of circles defined in Equation 3.8, is [178]:

$$\begin{vmatrix} \dfrac{\partial v_x}{\partial t_c} & \dfrac{\partial v_x}{\partial \theta} \\ \dfrac{\partial v_y}{\partial t_c} & \dfrac{\partial v_y}{\partial \theta} \end{vmatrix} = 0. \tag{3.9}$$

By resorting to the following notation:

$$\dot{v}_{c_x} = \frac{\partial V_{c_x}}{\partial t_c}, \quad \dot{v}_{c_y} = \frac{\partial V_{c_y}}{\partial t_c}, \dot{r} = \frac{d_r}{d t_c} = \frac{-R}{(t_c - t_0)^2}, \quad \Theta \equiv \tan\left(\frac{\theta}{2}\right), \tag{3.10}$$

we can rewrite Equations 3.8 and 3.9:

$$\Theta^2(-\dot{v}_{c_y} + \dot{r}) + \Theta(2\dot{v}_{c_y}) + (\dot{v}_{c_x} + \dot{r}) = 0, \tag{3.11}$$

which can be solved as a second order polynomial. The solutions identify the values of Θ for the tangent points of the envelope. However, these are real coordinates only when the discriminant, $|\dot{v}_c|^2 - \dot{r}^2$, is greater than zero, i.e., $|\dot{v}_c| \geq \dot{r}$. As a result, VO circles can only be calculated when the variation of the radius of the VO circles is less than the variation of the centre of the circles. Through Equation 3.7, we can consider that VO circles are only possible when:

$$|\dot{v}_c| < \frac{R}{(t_c - t_0)^2}. \tag{3.12}$$

An important case to consider is that, when the minimum separation has already been lost, no tangent solutions are possible. Therefore, intent VOs are only possible before LoS.

3.4. VARIABLE SPEED LIMIT (VSL) IMPLEMENTATION

VSL systems set speed limits to avoid unstable traffic conditions. The objective is to create a more homogeneous traffic situation leading to fewer congestion 'hotspots'. VSL has been successfully implemented with road vehicles to prevent crashes. More specifically, Wu [179] has shown that VSL improves safety when used at highway entrances. There are common aspects between the behaviour of agents at highway entrances and altitude transitions, that make applying VSL systems in the latter appealing. First, an outsider vehicle enters the main traffic lane in both situations. Second, similarly to highway entrances, agents are not expected to stop or reduce their speed significantly during layer transitions. Finally, while safety is paramount in both cases, it is also beneficial to improve efficiency by reducing travel times. This section describes how VSL was implemented for layer transitions.

3.4.1. AGENT

Multiple works that have applied reinforcement learning within air traffic control define aircraft as agents [180–184]. However, for air traffic control flow, preference is often given to some structural element within the operational environment [185]. This allows for a general control over aircraft, without having to directly control each single aircraft. The latter approach is not feasible within the high traffic densities expected, for example, for drone delivery operations [22]. Such would result in a large multi-agent system where

with each action, the next state depends not only on the action performed by the ownship, but on the combination of that action with the actions simultaneously performed by the intruders. Current research [186, 187] shows that emergent behaviour and complexity arise from agents interacting and co-evolving. From the point of view of each agent, the environment is non-stationary and, as training progresses, modifies in a way that cannot be explained by the agent's behaviour alone. Additionally, in a real-world scenario, having a fixed point is expected to facilitate the collection of data. Finally, aircraft may not have complete observability over the environment, more specifically over spaces to which they will travel in the future. Fixed zones are expected to have sufficient knowledge within a surrounding radius and can be distributed in a way (almost) covering the entire environment.

We employ an RL agent whose objective is to learn to set optimal speed limits in 'roads' of the environment, creating a homogeneous speed situation that guarantees minimum separation between cruising and climbing/descending aircraft. These roads do not have hard set delimiting points as in other works where physical entrances to the roads are used as limits [185]. We chose to let aircraft transition at whatever road better benefits their trajectory. As a result, the roads at which speed limits are applied depend on the route of climbing/descending aircraft. Figure 3.10 displays the following sub-sections:

- Detection Section: where cruising traffic is detected.
- Control Section: where aircraft adjust to the maximum speed set by the VSL agent.
- Entrance/Exit Section: where aircraft from adjacent traffic layers are expected to enter the current layer and/or cruising aircraft are expected to exit the current layer. Aircraft are expected to comply with the maximum speed set by the VSL agent.
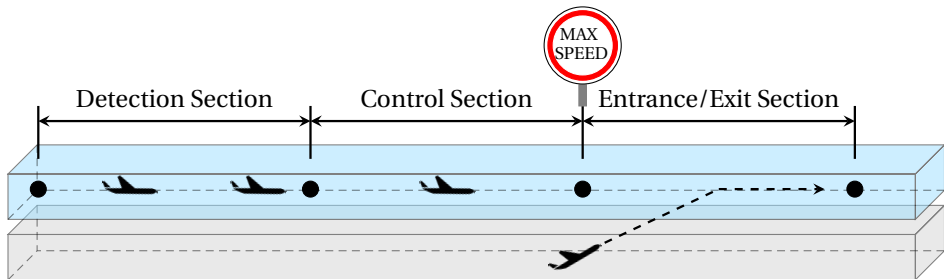

Figure 3.10: Sub-sections forming a road constructed around the movement of a climbing/descending aircraft. The RL agent will set a maximum speed limit for the entrance/exit section.

The entrance/exit sections of two different roads may not immediately follow each other. First, there would be not enough space for aircraft to adjust to the maximum speed on the second road. Second, it would not be possible to correctly assess the effect of each speed limit individually. As a result, one control section separating the two must be guaranteed. Figure 3.11 shows an example of entrance/exit sections formed around climbing/descending aircraft, while still maintaining minimum distance between each other. When this is not possible with the setting of the sections between two nodes, as is the case with the first and third roads, the length of the entrance/exit section is increased to include additional spatial nodes.
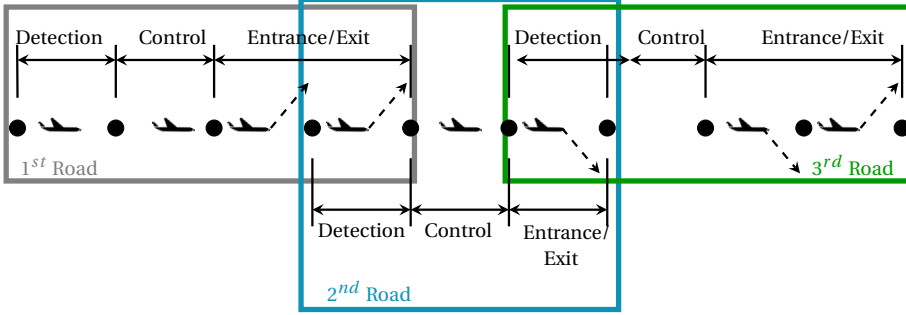
Figure 3.11: Two entrance/exit sections cannot follow each other. At least one control section must be set between the two.

It is assumed that all aircraft are able to adopt the set maximum speed. The maximum speed has a duration of 60 seconds. Afterward, if there are still aircraft climbing/descending to/from the road, a new maximum speed is requested with the state of the traffic in the road at that point. A 60 second time period was considered sufficient to correctly assess the consequences of the chosen maximum speed, while still allowing the RL agent to adequately respond to the changes in traffic flow over time.

3.4.2. LEARNING ALGORITHM

An RL method consists of an agent that interacts with an environment E in discrete timesteps. At each timestep, the agent receives the current state s of the environment and performs an action a for which it receives a reward s_t. The behaviour of an agent is defined by a policy, π, which maps states to a probability distribution over the available actions. The goal is to learn a policy that maximises the reward. Many RL algorithms have been researched in terms of defining the expected reward following action a. This work uses the Deep Deterministic Policy Gradient (DDPG), defined by Lillicrap [163].

Policy gradient algorithms first evaluate the policy, and then follow the policy gradient to maximise performance. DDPG is a deterministic actor-critic policy gradient algorithm, designed to handle continuous and high-dimensional state and action spaces. It has been shown to outperform other RL algorithms in environments with stable dynamics [164]. However, it can become unstable, being particularly sensitive to reward scale settings [188, 189]. As a result, the rewards must be carefully defined. The pseudo-code for DDPG is displayed in Algorithm 3.1.

DDPG uses an actor-critic architecture. The actor produces an action given the current state of the environment. The critic estimates the value of any given state, which is used to update the preference for the executed action. DDPG uses two neural networks, one for the actor and one for the critic. The actor function $\mu(s|\theta^\mu)$ (also called policy) specifies the output action a as a function of the input (i.e., the current state s of the environment) in the direction suggested by the critic. The critic $Q(s, a|\theta^Q)$ evaluates the actor's policy, by estimating the state-action value of the current policy. It evaluates the new state to determine whether it is better or worse than expected. The critic network is updated from the gradients obtained from a temporal-difference (TD) error signal at each time step. The

3

Algorithm 3.1 Deep Deterministic Policy Gradient

Initialize critic $Q(s|a^\mu)$ and actor $\mu(s|\theta^\mu)$ networks, and replay buffer R
for all episodes **do**
    Initialize action exploration
    while episode not ended **do**
        Select action a_t according to the current state s_t from environment and the current *actor* network
        Perform action a_t in the environment and receive reward r_t and new state s_{t+1}
        Store transition (s_t, a_t, r_t, s_{t+1}) in replay buffer R
        Sample a random mini-batch of N transitions from R
        Update *critic* by minimizing the loss
        Update *actor* policy using the sample policy gradient
        Update *target* networks
    end while
    Reset the environment
end for

---

output of the critic drives learning in both the actor and the critic. θ^μ and θ^Q represent the weights of each network. Updating the actor and critic neural network weights with the values calculated by the networks may lead to divergence. As a result, target networks are used to generate the targets. The target networks are time-delayed copies of their original networks, $\mu'(s|\theta^{\mu'})$, and target critic, $Q(s', a|\theta^{Q'})$, which slowly track the learnt networks. All hidden neural networks use the non-sigmoidal rectified linear unit (ReLU) activation function, as this has been shown to outperform other functions in statistical performance and computational cost [190].

The neural network parameters used in our experimental results are based on Lillicrap [163]. *Experience replay* is used to improve the independence of the samples in the input batch. Past experiences are stored in a *replay buffer*, a finite sized cache R. At each timestamp, the actor and critic are updated by sampling data from this buffer. However, if the *replay buffer* becomes full, the oldest samples are discarded. Finally, *exploration noise* is used to promote exploration of the environment; an Ornstein-Uhlenbeck process [191] is used in parallel to the authors of the DDPG method.

3.4.3. STATE

The state should provide enough information on the evolution of the traffic flow to allow the RL method to correctly respond to the emergent behaviour. Due to the complexity of the dynamics of traffic flow, it is non-trivial to precisely define this evolution. As suggested by other works [179], traffic flow is defined herein as the number of aircraft passing through a first measure point at the beginning of the road and exiting at a second measure point at the end of the road. In this work, these correspond to the start of the detection section and the end of the entrance/exit section represented in Figure 3.10, respectively. Furthermore, it is assumed that there is enough information available on the aircraft and speed limits on each road. A fixed state array (dim = 4) is used, with each position of the array identifying the following:

1. Number of aircraft expected to transition vertically into the entrance/exit section in the next 60 seconds.
2. Number of aircraft expected to transition vertically out of the entrance/exit section in the next 60 seconds.

3. Cruising aircraft expected to travel from the detection area into the entrance/exit section in the next 60 seconds.
4. Current maximum speed in the detection section.

3.4.4. ACTION

A softmax activation function is used for classification. This function normalises an input vector, $\vec{z}$, of K real values into a vector of K real values between 0 and 1 that sums up to 1. As a result, these values can be interpreted as probabilities. The mathematical definition of the softmax function is as follows:

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} \exp(z_j)},$$ (3.13)

where z_i are the elements of the input vector for the softmax function.

Probability values are set for the discrete options for maximum speed: 10 kts, 15 kts, 20 kts, 25 kts, or 30 kts. The speed value with the highest probability value is used.

3.4.5. REWARD

The reward given to the RL agent is based primarily on safety. However, within safety, several factors may be considered. The paramount objective is to lead the agent to favour maximum speeds that reduce the likelihood of LoSs. In previous work [182], we have seen that focusing mainly on the total number of LoSs is the best reward structure to reduce it. However, the number of LoSs per call to the RL agent might be too sparse to favour fast convergence to an optimal solution. As a result, to complement the number of LoSs, we consider near-LoSs, i.e., aircraft encounters that nearly result in a loss of minimum separation. Near-LoSs are identified on the basis of the time to LoS. However, naturally, a near-LoS has a weight lower than that of a LoS.

Although VSL is used primarily to improve safety and not efficiency [192], by favouring higher speeds, it is possible to reduce travel times. With this in mind, two elements favouring higher speeds are added to the reward structure: (1) a positive reward for when the final detected outflow matches/surpasses the expected outflow, and negative when it is inferior; (2) a positive reward when higher travelling speeds are selected. The expected outflow is calculated as follows:

$$\text{outflow} = \text{aircraft}_{\text{cruise}} - \text{aircraft}_{\text{out}} + \text{aircraft}_{\text{in}},$$ (3.14)

where $\text{aircraft}_{\text{cruise}}$ represents the aircraft detected at the start of the detection section, $\text{aircraft}_{\text{out}}$ the aircraft transitioning vertically out of the section, and $\text{aircraft}_{\text{in}}$ the aircraft expected to vertically merge into the section. Note that the expected outflow is only calculated for the 60 seconds period that the maximum speed is set for. The final outflow is then verified by checking the aircraft that cross the end of the entrance/exit section. In summary, the final reward value is obtained by summing the following components:

1. A negative reward for a LoS within the road (-10 per LoS).
2. A negative reward for near-LoS within the road (-4 when time to Los < 10 s; -2 when time to LoS > 10 s).

3. The difference between the final detected traffic flow and the expected traffic flow. Higher traffic flow is positively rewarded (+1 for each additional aircraft that leaves the road). An inferior traffic flow is negatively rewarded (-1 for each aircraft that does not exit the road as it was expected).
4. A positive reward for higher maximum speeds (0 for 10 kts; +1 for 15 kts; +2 for 20 kts; +3 for 25 kts; +4 for 30 kts).

3.4.6. Aircraft Compliance With The Maximum Speed

The success of VSL implementation is directly related to the percentage of aircraft that comply with maximum speeds. Otherwise, speed heterogeneity in the environment is not mitigated, and thus no improvement can be achieved. The effect of non-compliance per part of the operating aircraft is analysed within the experimental results.

3.5. Experiment: CR in Urban Environment with VSL

3.5.1. Apparatus and Aircraft Model

The Open Air Traffic Simulator Bluesky [25] was used to test the efficiency of speed-only based conflict resolution with SSD in an urban environment. Bluesky has an Airborne Separation Assurance System (ASAS) to which CD&R methods can be added, allowing for different CD&R implementations to be tested under the same scenarios and conditions. A DJI Mavic Pro model was used for the simulations. Speed and mass were retrieved from the manufacturers data, and common values were assumed for the turn rate (max: $15°/s$) and acceleration/breaking (1.0 kts/s).

3.5.2. Independent Variables

Four independent variables are included in this experiment: state/intent information usage, heading-altitude rules, compliance with variable speed limits, and traffic density.

State/Intent Information Usage

Two different situations with using state and intent information will be tested in order to establish how to maximise the effect of using intent information:
1. Only state (S) information: common application that will be used as a performance baseline for comparison.
2. State and intent information is used simultaneously ($S \wedge I$). Conflicts are detected and resolved by preparing for both situations: whether intruding aircraft continue their current state or follow their intent. This is a conservative approach, with aircraft working to prevent all possible risk situations. The disadvantage is that more VOs are included in the solution space and the number of velocity vectors which can avoid all conflicts becomes smaller; it can potentially even reach a situation where no solution exists.

Heading-Altitude Rules

Two different rules settings will be tested with:
1. All aircraft travel at the same altitude layer, independently of heading. Used for baseline comparison.
2. Multiple altitude layers are used. In each layer, aircraft have similar headings.

VARIABLE SPEED LIMITS COMPLIANCE

When multiple altitude layers are used, three different situations of VSL usage will be tested with:

1. No variable speed limits are applied, aircraft to follow the maximum cruise speed. Used for a baseline comparison.
2. Variable speed limits applied by the RL agent. Aircraft with compliance rate of 100%.
3. Variable speed limits applied by the RL agent. Aircraft with compliance rate of 90%.

TRAFFIC DENSITY

Traffic density varies from low to high according to Table 3.2. High densities spend at least more than 10% of their flight time avoiding conflicts [193].

Table 3.2: Traffic volume used in the experimental simulations.

	Low	Medium	High
Traffic density [$ac/10\,000\,\text{NM}^2$]	81,247	162,495	243,744
Number of instantaneous aircraft [-]	25	50	75
Number of spawned aircraft [-]	453	926	1366

The RL agent used to set variable speed limits is trained at a medium traffic density. Afterward, testing will use all three traffic densities: low, medium, and high. In this way, it is possible to assess the efficiency of an agent trained at a different traffic density.

3.6. EXPERIMENTAL DESIGN AND PROCEDURE

3.6.1. MINIMUM SEPARATION

The value of the minimum safe separation distance may depend on the density of air traffic and the region of the airspace. For unmanned aviation, there are no established separation distance standards yet, although 50 m for horizontal separation is a value commonly used in research [59], and will therefore be used in the experiments herein performed. For vertical separation, 30 ft was assumed.

3.6.2. CONFLICT DETECTION

The experiment will employ state-based conflict detection for all conditions. This assumes a linear propagation of the current state of all involved aircraft. Using this approach, the time to CPA (in seconds) is calculated as:

$$t_{CPA} = -\frac{\vec{d}_{rel} \cdot \vec{v}_{rel}}{\vec{v}_{rel}},$$ (3.15)

where $\vec{d}_{rel}$ is the cartesian distance vector between the involved aircraft (in meters), and $\vec{v}_{rel}$ the vector difference between the velocity vectors of the involved aircraft (in meters per second), pointed towards the intruder's protected zone.

The distance between aircraft at CPA (in meters) is calculated as:

$$d_{CPA} = \sqrt{\vec{d}_{rel}^2 - t_{CPA}^2 \cdot \vec{v}_{rel}^2}.$$ (3.16)

When the separation distance is calculated to be smaller than the specified minimal horizontal spacing, a time interval can be calculated in which separation will be lost if no action is taken:

$$t_{in}, t_{out} = t_{CPA} \pm \frac{\sqrt{R_{PZ}^2 - d_{CPA}^2}}{\vec{v}_{rel}} \tag{3.17}$$

These equations will be used to detect conflicts, which are said to occur when $d_{CPA} < R_{PZ}$, and $t_{in} \leq t_{lookahead}$, where R_{PZ} is the radius of the protected zone, or the minimum horizontal separation, and $t_{lookahead}$ is the specified look-ahead time. A look-ahead time of 30 seconds is used for conflict detection and resolution.

3.6.3. SIMULATION SCENARIOS

The geographic area used in the experiment is a small section of San Francisco with an area of 1.708 NM^2, as was illustrated in Figure 3.2. Roads and intersections are represented by edges and nodes, which aircraft can use to build their route. Aircraft can only travel from one node to another if there is a road connection between the two. The aircraft spawn locations (origins) and destinations are placed in alternating order on the edge of this area, with a spacing equal to the minimum separation distance plus a 10% margin, to avoid conflicts between spawn aircraft and aircraft arriving at their final destination. In the case of only one traffic layer, aircraft are spawned at that corresponding altitude. When multiple layers are used, aircraft spawn at the altitude of the layer that corresponds to the initial heading. In terms of climb rate, aircraft are expected to climb almost vertically. Take-off and landing are not simulated.

Each aircraft has three delivery points (or waypoints) through which it must pass. The delivery points are always nodes on the map. The exact nodes are randomly assigned. However, the pool of nodes to choose from is spread in such a way that each aircraft crosses the map. The total flight distance and time depend on the location of these nodes. During the generation of the scenario files, the total flight path/time of the already created aircraft was taken into account, so the desired instantaneous traffic densities are respected. These values will be presented in the experimental results for reference. Each scenario runs for 2 hours. Each traffic density is tested with three different repetitions, each with different trajectories.

Between the set delivery points, it is assumed that aircraft will favour safety and efficiency in their route planning, in this order. The main priority of any aircraft shall be to limit the number of altitude transitions, as crossing multiple layers is likely to result in both an increase of the total number of conflicts and of the travel time. Next, adoption of routes with the fewest turns is also preferable, as in our scenarios more turns lead to more altitude transitions. Lastly, routes with shorter distances are preferable efficiency-wise. As a result, aircraft calculate their trajectory prioritising, in decreasing order of preference:

1. Fewer altitude variations.
2. Fewer turns.
3. Shortest distance.

Ultimately, an aircraft is removed from the simulation once it leaves the simulation area. To prevent aircraft being removed incorrectly when travelling through an edge road,

aircraft are set to move out of the map once they finish their route and are removed once they move away from an edge node.

3.6.4. Dependent Variables

Three different categories of measures are used to evaluate the effect of the different operating rules set in the simulation environment: safety, stability, and efficiency.

Safety Analysis

Safety is defined in terms of the number and duration of conflicts and losses of separation, where fewer conflicts and losses of separation are considered safer. Additionally, losses of separation are distinguished based on their severity according to how close aircraft get to each other:

$$LoS_{sev} = \frac{R - d_{CPA}}{R}. \tag{3.18}$$

A low separation severity is preferred.

Stability Analysis

Stability refers to the tendency for tactical conflict resolution manoeuvres to create secondary conflicts. In literature, this effect has been measured using the Domino Effect Parameter (DEP) [151]:

$$DEP = \frac{n_{cfl}^{ON} - n_{cfl}^{OFF}}{n_{cfl}^{OFF}}, \tag{3.19}$$

where n_{cfl}^{ON} and n_{cfl}^{OFF} represent the number of conflicts with CD&R ON and OFF, respectively. A higher DEP value indicates a more destabilising method, creating more conflict chain reactions.

Naturally, conflict resolution manoeuvres that deviate from the nominal path are expected to create more secondary conflicts, due to the scarcity of free space at high travelling densities. Herein, speed-only based resolution manoeuvres are applied, and thus aircraft do not deviate from their path due to conflict resolution. As a result, the effect on stability from avoiding conflicts is not expected to be as pronounced. However, when multiple traffic layers are employed, aircraft increase their path to correctly adjust to the heading range of the crossed layers. The negative effect on stability resulting from this increase in flight path/time will be analysed.

Efficiency Analysis

Efficiency is evaluated in terms of the distance travelled and the duration of the flight. Significantly increasing the path travelled and/or the duration of the flight is considered inefficient. The effect on total flight path/time resulting from layer transitions will be analysed and compared with the baseline case of having only one traffic layer. Furthermore, conflict resolution and the application of variable speed limits with the RL agent are expected to have an effect on the average speed of the aircraft. The added flight time will be compared to the baseline case, where no conflict resolution is performed and no speed limits are set.

3.7. Experimental Hypotheses

3.7.1. Speed-Only Conflict Resolution

Speed-only conflict resolution naturally has its limitations: there are not as many options for resolution manoeuvres as when heading and/or altitude variations are also possible. It was hypothesised that the SSD method will have better efficacy when applying heading-altitude rules. (Near-)head-on conflicts are not expected, as aircraft in the same altitude layer, have similar headings. Independently of the airspace structure, efficacy of the speed-only based conflict resolution method is expected to deteriorate as traffic density increases. Existent research [61, 62] show that the efficacy of speed-only resolution depends on (1) the nominal minimal separation between the aircraft, and (2) the time available to loss of separation. As traffic density increases, the space between aircraft is expected to reduce, and consequently, so is the time to loss of separation.

3.7.2. State vs Intent Information in Conflict Resolution

It was hypothesised that using intent information alone is not sufficient for successful conflict resolution. At high traffic transitions, aircraft spent a considerable amount of time in conflict, where the speed vector output from the conflict resolution method is used instead of the intent speed vector. Ultimately, the current state information is the best indication of the state during conflict resolution, as aircraft will try to differ from it as little as possible (i.e., the conflict-free speed vector that constitutes the smallest deviation from the current state is always chosen for conflict resolution).

However, it was expected that considering intent information would improve safety. With state information only, heading/altitude variations would only be detected once intruders have completed the change, which may be too late to prevent LoSs. It was hypothesised that using both state and intent information simultaneously ($S \wedge I$) would increase the number of detected conflicts (i.e., false negative conflicts are added and false positive conflicts are not discarded), but would prevent more LoSs as all possible future cases (i.e., intruder following intent or entering conflict resolution) are defended from in advance.

It is not clear in which structure (i.e., with one layer or multiple layers) the use of intent is more beneficial. There are advantages and disadvantages in both cases. On the one hand, when all traffic operates at the same altitude, intent has the biggest impact, as it allows for removing false positive conflicts and adding false negative conflicts resulting directly from turns. On the other hand, given the high traffic density, adding intent may saturate the solution space and render finding an optimal solution impossible. Additionally, with multiple layers, the structure itself already defends from turns, as these are performed within the transition altitudes. In this case, intent information aids by removing false positive conflicts from intruders that are about to climb/descend and adds false negative conflicts from intruders about to join the layer of the ownship. However, here resolving all conflicts is non-trivial as there are conflicts in both horizontal and vertical layers. Even though the ownship is better informed regarding conflicts, this may not be enough to actually find a solution that successfully resolves them all. As a result, adding intent may not have a pronounced effect on safety.

3.7.3. Heading-Altitude Rules

The application of heading-altitude rules is expected to strongly reduce the number of LoSs and conflicts as both the traffic density and the likelihood of aircraft meeting in conflict decrease compared to having only one traffic layer. The weakness of this method is the added conflicts resulting from vertical transitions between layers. Having to resolve conflicts on both the horizontal and vertical dimensions, increases the complexity of finding a solution to resolve all conflicts. Having a high number of altitude transitions, which is expected at high traffic densities, hinders conflict resolution efficiency. Efficiency-wise, heading-altitude rules are expected to increase the 3D flight travel distance and, consequently, the flight travel distance.

3.7.4. Variable Speed Limits with Reinforcement Learning

It was hypothesised that setting variable speed limits would improve the speed homogeneity of the environment, which in turn improves safety between cruising and climbing/descending aircraft. Between the former and the latter, speed differences are expected. However, it was also hypothesised that VSL only improves safety when a large majority of the operating traffic complies with the speed limits. Safety levels are expected to decrease directly with the compliance rate.

   The RL agent will be tested with traffic densities similar and different from those of the training conditions. The agent is naturally expected to perform better at the densities at which it was trained. However, applying the agent at different densities allows one to assess the dependency of maximum speed solutions on traffic densities. It was hypothesised that the agent may be the least efficient at densities higher than the one in which it was trained, as the complexity of the emergent behaviour and of the consequent solution increases proportionally with the density.

3.8. Experimental Results

The final best scenario expected is when all the structural rules are applied to the environment: (1) heading-altitude rules divide aircraft into multiple layers, (2) variable speed limits are in place to improve speed homogeneity between cruising and climbing/descending aircraft, and (3) intent trajectory propagation is added to conflict resolution, allowing the CR method to prepare for all possible future cases (i.e., intruders following intent or entering conflict resolution mode). However, to properly analyse the effect of the multiple independent variables on the dependent measures, several baseline situations are presented alongside this scenario: (a) one layer scenario (i.e., all traffic operates at the same altitude), (b) a multi-layer situation without variable speed limits, (c) a multi-layer situation with only 90% compliance rate to the variable speed limits. All of the previous situations are tested with different traffic densities, and different state/intent information usage for conflict resolution, as well as a situation without conflict resolution (CR-OFF).

   Box-and-whisker plots are used on multiple occasions to visualise the sample distribution over several simulation repetitions. Efficiency, stability, and time in conflict values present outliers; the number of outliers is consistent throughout (<10% of the total data). As these do not contribute to the comparison between the different states, they are not displayed for the sake of clarity.

3.8.1. TRAINING OF THE RL AGENT FOR VARIABLE SPEED LIMITS

The RL agent responsible for setting the variable speed limits was trained at a medium traffic density. In total, 300 episodes were run. An episode is a full execution of the simulation environment, which runs for 2 hours. During training, conflict resolution was used with state information only, in order to increase computational speed.

SAFETY ANALYSIS

The episodes do not all have the same number of calls to the DDPG method. This is proportional to the maximum speeds set. Each maximum speed is set for 60 seconds. If lower speeds are used during the transition progress, traffic will move slower. As a result, after 60 seconds, the DDPG may be called again for the same section if aircraft transitioning between layers have not finished their transition yet. Figure 3.12 shows the evolution of the total number of calls to the DDPG per episode during training. The trained RL agent stabilised around 1755 calls.
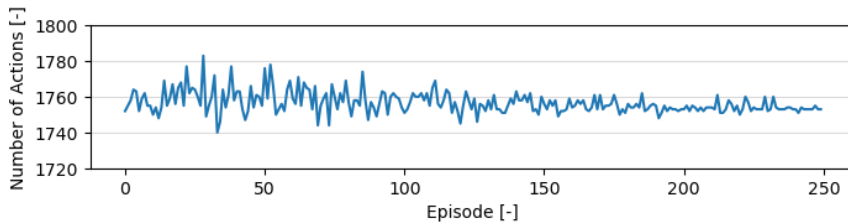
Figure 3.12: Number of calls to the RL agent per episode during training.

Figure 3.13 shows the evolution of the total number of LoSs per episode during training. The method can converge to a stable value after around 250 episodes.
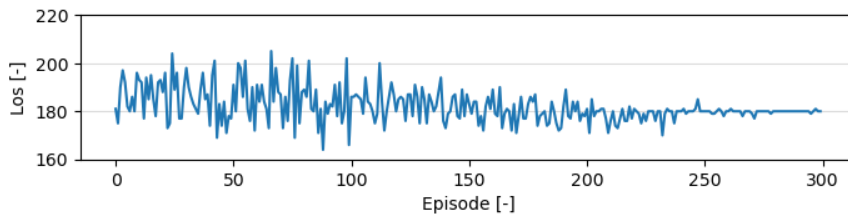
Figure 3.13: Total number of losses of separation per episode during training of the RL agent.

Figure 3.14 shows the speed limits applied in one episode that lead to a decrease in the total number of LoSs. At each step, the RL agent picks a speed limit from the set of discrete options displayed on the y-axis. In almost 95% of the times, a maximum speed of 25 kts was chosen. Favouring one speed value is a result of aircraft being able to climb/descend at any point. Consequently, the sections are very close together, and maintaining a homogeneous maximum speed between neighbouring sections is beneficial. The other discrete options were used in similar numbers, with no clear preference between the four options. From our experiments, we see that those singular cases where smaller maximum speed values (10 kts–20 kts) are used are crucial. These lead to better final

results safety-wise than an episode where all maximum speeds are set at 25 kts. However, from the results, it is not clear how or when the agent decides to apply lower speeds as a limit.

Why 25 kts? The reinforcement learning agent found this value to be the best balance between the desire for high speed, in order not to considerably increase travel time, and improving safety. This is naturally related with the performance limits of all aircraft, separation between traffic layers, rate of climbing. All these factors contribute to the best decision; different values will likely yield different maximum speeds.
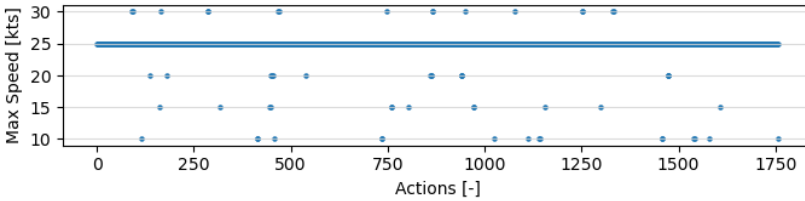
Figure 3.14: All maximum speeds set in one training episode.

Figure 3.15 shows the average reward per call to the RL agent in the same episode shown in Figure 3.14. In most steps, the RL agent achieves a positive reward. However, outliers indicate that, on some occasions, preventing LoSs/near-LoSs is practically impossible. Naturally, these rewards are directly related to the traffic density in which the agent is trained and the consequent number of LoSs and near misses.
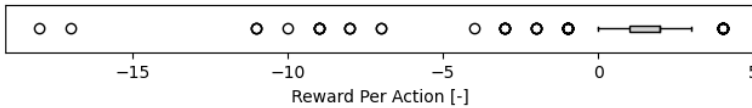

Figure 3.15: Average reward per action obtained by RL agent in one training episode.

Figure 3.16 shows the evolution of the total number of pairwise conflicts per episode during training. As seen in Figures 3.17 and 3.19, the total number of conflicts is not directly correlated with the total number of LoSs. During training, not all episodes with the fewest conflicts also had the fewest LoSs.
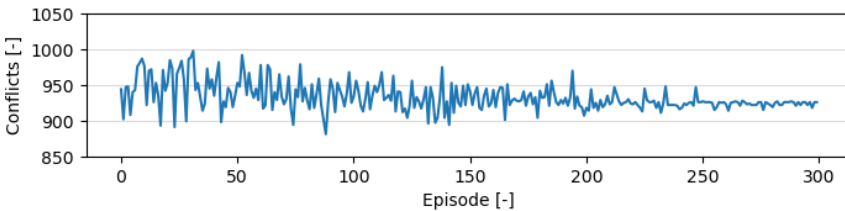

Figure 3.16: Total number of pairwise conflicts per episodes during training of the RL agent.

3.8.2. TESTING OF THE RL AGENT FOR VARIABLE SPEED LIMITS

SAFETY ANALYSIS

Figure 3.17 shows the mean total number of pairwise conflicts. A pairwise conflict is counted only once, independently of its duration. As hypothesised, applying heading-altitude rules reduces the total number of conflicts, on average, by 80%. As aircraft are dispersed per the several altitude layers, there is more free space in each layer. Additionally, conflict resolution only reduces the total number of conflicts in the one layer situation, with a bigger efficiency at a high traffic density. However, the lack of a strong reduction in the total number of conflicts is not necessarily a sign of poor efficiency, since conflicts are a necessary element for propagating speed reductions backward at intersections. Furthermore, as expected, when using both state and intent information, more conflicts are considered than when using state information alone. Finally, applying variable speed limits (VSL) on a multi-layer structure does not have a pronounced effect on the number of conflicts.
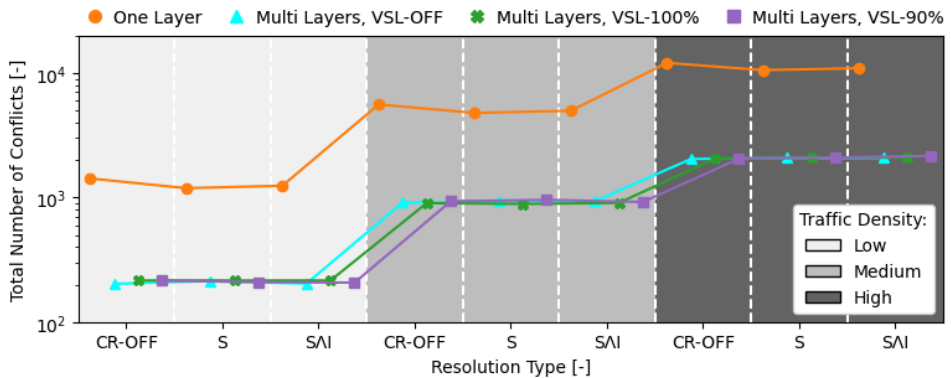

Figure 3.17: Mean total number of pairwise conflicts.

Figure 3.18 shows the amount of time spent in 'conflict mode' per aircraft. An aircraft enters 'conflict mode' when it adopts a new state computed by the CR method. The aircraft will exit this mode, once it is detected that it is past the previously calculated time to CPA (and no other conflict is detected). At this point, the aircraft will redirect its course to the next waypoint. The time to recovery is not included in the total time in conflict. Based on this information and Figure 3.17, the number of conflicts is not directly correlated with the amount of time in conflict. The considerable increase in number of conflicts with a high traffic density compared to a medium traffic density, does not have a direct correlation in the average time in conflict. Employing heading-altitude rules reduces the average time in conflict, albeit more significantly with a lower traffic density. Additionally, there is no pronounced difference in time of conflict resulting from employing variable speed limits. Finally, the addition of intent information only increases the time in conflict with a one-layer structure.

Figure 3.19 shows the mean total number of LoSs. As hypothesised, applying heading-altitude rules reduces the total number of LoSs on average by 85%. When all traffic
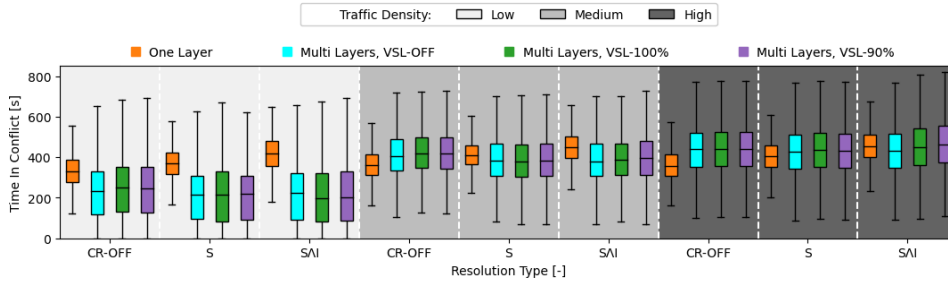
Figure 3.18: Total time in conflict per aircraft.

is contained in one layer, speed-only based conflict resolution is hardly capable of an improvement. At medium and high traffic densities, only about 5% of the total number of LoSs are prevented compared to a CR-OFF situation. With the high likelihood of aircraft meeting in conflict increasing with traffic density, it is progressively harder for the SSD method to find a solution that resolves all conflicts. Furthermore, by comparing Figures 3.19 and 3.17, we see that the relation between the total number of LoSs and conflicts is not linear; fewer conflicts do not necessarily equal fewer LoSs.

Unfortunately, the addition of intent results in a negligible reduction in the total number of LoSs with a one-layer structure. As hypothesised, at these high densities, the benefit of adding intent information is outweighed by the increase in saturation of the solution space. With a multi-layer structure, the benefit is more pronounced, albeit still small: adding intent reduces the total number of LoSs to about 5% at high traffic densities compared to a state-only conflict resolution. Adding intent allows aircraft to better assess the danger of climbing/descending intruders. However, speed-only based conflict resolution can do little with simultaneous horizontal and vertical conflicts. Additionally, note that a small look-ahead time reduces the differences between state and intent information. In these simulations, a look-ahead time of 30 s was used for conflict detection and resolution. With a higher look-ahead time, as the state of intruders is projected farther into the future, thus increasing uncertainties, the difference between intent and state information is greater. Thus, intent becomes progressively more beneficial as the look-ahead time increases. On the other hand, a larger look-ahead time results in more conflicts being accounted for, thus saturating the solution space and increasing the number of situations where no solution is available. All these factors must be taken into account.

Decreasing the number of losses of minimum separation is the paramount objective of employing variable speed limits with a reinforcement learning agent. With full compliance, there is an average decrease of 15% of the total number of LoSs at the medium traffic density in which the agent was trained. With different traffic densities, as hypothesised, the agent is more efficient with a lower density than with a higher one. As traffic densities increase, so does the complexity of the emergent behaviour, and more complex solutions need to be developed. Furthermore, as the compliance rate decreases, the benefit is lost. A 90% compliance rate is already not sufficient. Consequently, a 100% compliance rate must be guaranteed.
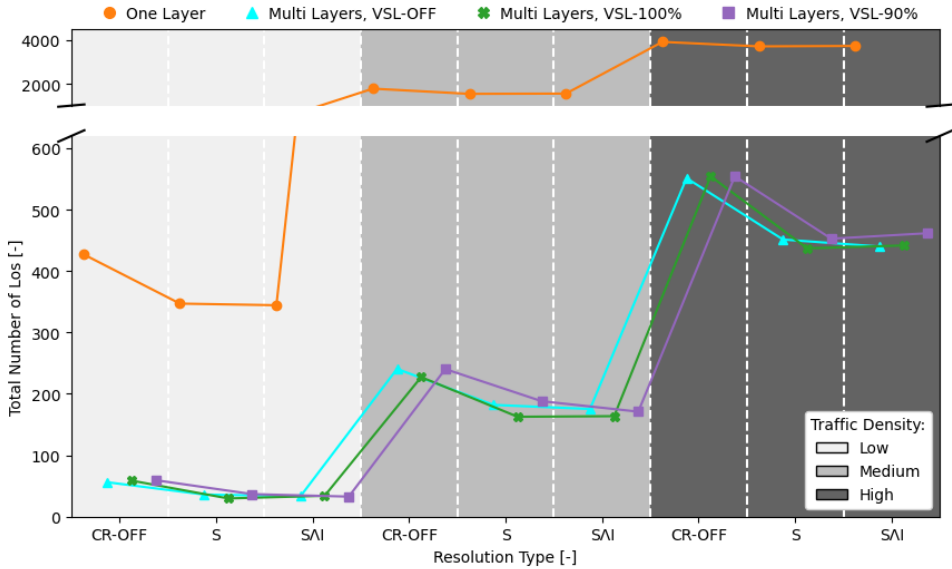
3

Figure 3.19: Mean total number of losses of separation.

Figure 3.20 displays the intrusion severity. No direct correlation was observed between the severity of the intrusion and the traffic density. As the one-layer situation has a much greater number of total LoSs (see Figure 3.19), there is a more heterogeneous set of values and the average severity is closer to the median of the total range. However, it is interesting to note that, with multiple layers, intrusion severity has a high average, meaning that aircraft in a LoS situation get very close at CPA. This is likely due to conflicts resulting between cruising and climbing/descending aircraft, which are very hard to defend from with only speed-based conflict resolution.
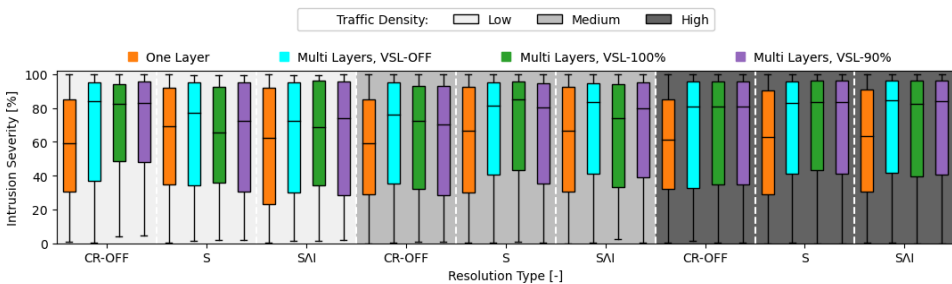

Figure 3.20: Intrusion severity rate.

Figures 3.21 and 3.22 focus on the multiple layers configuration in order to gain more insight into how to further prevent LoSs between cruising and climbing/descending aircraft. Figure 3.21 shows the relative speed between pairwise aircraft in a LoS situation.

More LoSs occur when there is a higher relative speed between aircraft. As expected, with a heterogeneous distribution of speed between aircraft, it is harder to maintain adequate spacing between aircraft. Interestingly, at both low and medium traffic densities, variable speed limits appear to have the same effect of reducing relative speeds as applying conflict resolution.
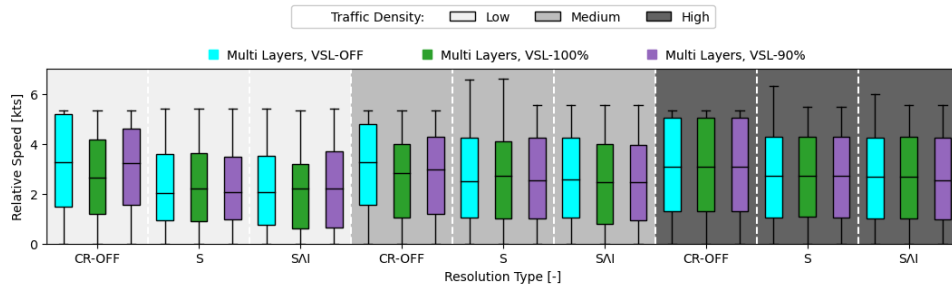

Figure 3.21: Relative speed between pairs of aircraft during losses of separation with multiple layers.

Figure 3.22 shows where LoSs occur in a multi-layer situation without VSL. As expected, most LoSs occur during the transition to different altitude layers. Improving safety during these transitions should thus be the focus when using a multi-layer structure.
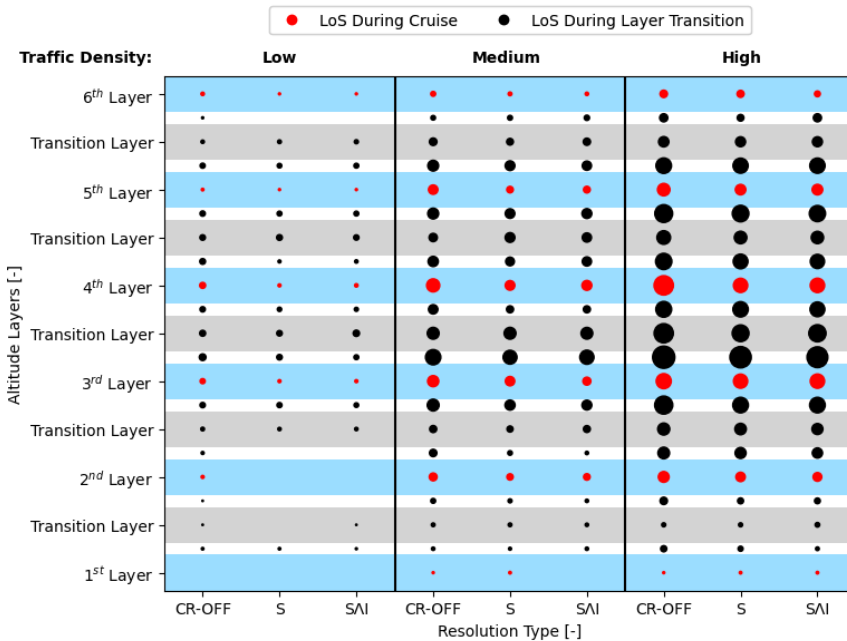

Figure 3.22: Schematic view of the altitude at which LoSs occur with multiple layers. The size of the points varies between a maximum value of 182 and a minimum value of 3 LoSs.

STABILITY ANALYSIS

Figure 3.23 displays the mean DEP value. A high positive value indicates the occurrence of conflict chain reactions that cause airspace instability. As seen previously with the total number of conflicts (see Figure 3.17), speed-only based conflict resolution does not greatly influence the stability of the environment.
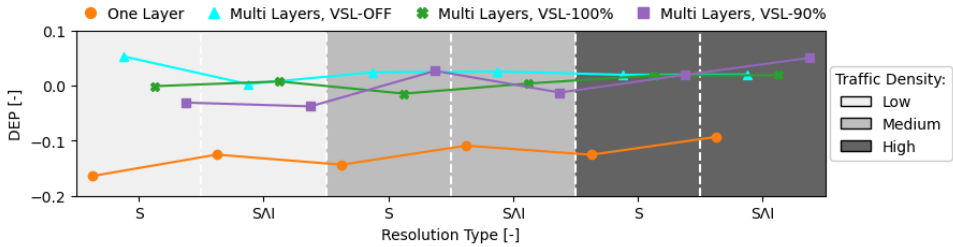

Figure 3.23: Domino effect parameter values.

EFFICIENCY ANALYSIS

For reference, Figures 3.24 and 3.25 show the average flight time and flight path per aircraft, respectively, without conflict resolution. As expected, with multiple layers, aircraft travel longer. Adding to their route, aircraft have to transition between layers which increases their 3D flight distance and, consequently, their flight time.
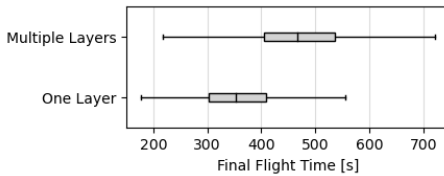

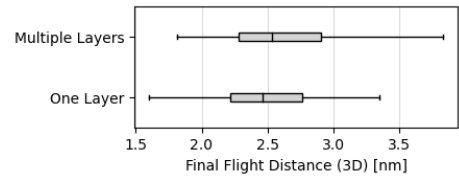Figure 3.24: Flight time per aircraft without CR.

Figure 3.25: Flight path per aircraft without CR.

Figure 3.26 shows the average number of instantaneous aircraft per timestep of an episode. The simulation scenarios were built taking into account an intended instantaneous traffic density of 25, 50, and 75 aircraft per low, medium, and traffic density, respectively. These values were calculated for a CR-OFF, one-layer situation. In a multi-layer situation, as seen in Figure 3.24, average flight time increases as a result of extra climbing/descending actions as well as of the extra horizontal path to correctly adjust to the traffic heading in each traversed layer. As a result, the average instantaneous traffic density also increases. Additionally, it was expected that the application of conflict resolution increases flight time, as aircraft employ resolution speeds instead of their preferred cruising speed, which is usually higher in order to decrease travel time. However, this effect is pronounced only in a one-layer structure.

Figure 3.27 shows the extra flight time resulting of employing conflict resolution vs a CR-OFF situation. Both situations, one-layer and multiple layers, have naturally different CR-OFF values, as previously displayed in Figures 3.24 and 3.25. With only one layer, conflict resolution has a worse efficiency. With a higher number of conflicts and time in
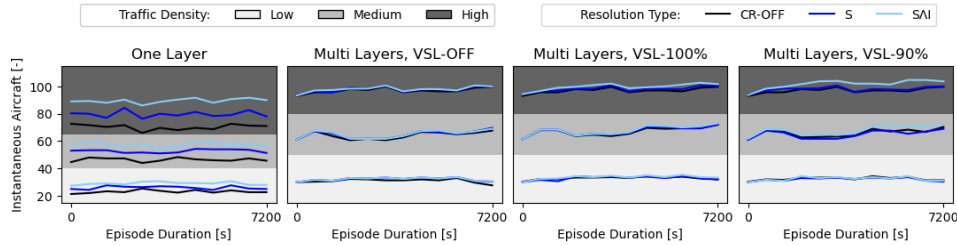
Figure 3.26: Mean number of instantaneous aircraft per timestep throughout simulation scenarios.

conflict (see Figures 3.17 and 3.18, respectively), conflict resolution tends to pick solutions with lower speeds, which increases flight time. When state and intent information are used simultaneously ($S \wedge I$), more conflicts are consider; the increase in flight time is visible below.
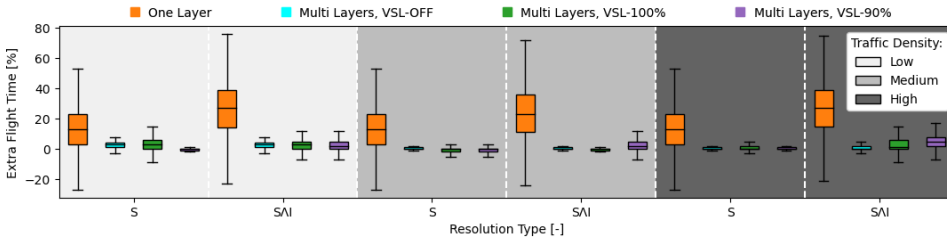

Figure 3.27: Extra flight time per aircraft.

3.9. DISCUSSION

The application of heading-altitude rules, VSL, and the combination of intent with state information had a positive effect in reducing the total number of LOSs (in decreasing order of effect). However, there are questions regarding their implementation: (1) the benefit of adding intent information is lost as traffic density increases, and thus its usage should be weighted against the expected densities and cost of implementation; (2) VSL implementation resulted in the same maximum speed value being employed most of the time, which raises questions regarding the ability of the method to adapt and personalise maximum speed values. Comparison with previous VSL research indicates that this might be due to the characteristics of the environment: adjacent sections, one unique lane with uniform cruising traffic, and rewards based on a safety factor that improves with speed homogeneity. Further work with different airspace structures is needed to better understand them. The following sub-sections dwell further into these subjects.

3.9.1. STATE VS INTENT INFORMATION IN CONFLICT RESOLUTION

Combining intent and state information reduces the number of LoSs compared to using state information alone. The efficacy of this method is due to the combination of information of the current state and intent which provides guidance regarding the future

state. However, a disadvantage of using both intent and state information simultaneously with the SSD method is that the solution space becomes saturated faster, especially as the traffic density increases. As a result, combining state and intent was more effective when more traffic layers were in place, since there are fewer conflicts per layer to consider.

Furthermore, the benefit of using intent is directly associated with the type of variations allowed for conflict resolution. In previous work [194], intent information was added to a no-boundary setting, with heading/speed variations for conflict resolution, and a higher look-ahead time. The previous characteristics improved the benefit of adding intent information. Being allowed to modify heading for conflict resolution, greatly increases the number of conflict-free speed vectors which can be selected from the solution space. Consequently, reduction of the amount of these vectors when intent information is added, is not as detrimental as when only speed variation is possible. Thus, when using a conflict resolution method such as SSD, using intent information might be beneficial only at low traffic densities and/or when both heading and speed variation are allowed, as more conflict-free resolution speed vectors are available.

Finally, the efficacy of all resolution manoeuvres is dependent on the speed, acceleration, of the involved aircraft. Applying different resolution methods, and/or aircraft type, may naturally produce different results. It may still be of interest to research how other conflict detection and resolution methods react to adding intent information and which differences may exist in the final resolution speeds selected. However, safety improvements that result directly from the use of intent information must be considered together with the expense of its implementation. First, the deterioration of safety improvements must be hypothesised in a real-case scenario. Delays in data transmission and processing may delay the reaction to state changes in neighbouring aircraft. Second, the effect on safety is directly associated with the number of aircraft that can share and analyse intent information. To achieve the desired improvement, most aircraft in the airspace would require this capability.

3.9.2. HEADING-ALTITUDE RULES
The paramount factor in safety is the number of minimum separation violations. Here, the airspace design can be seen as a first layer of protection, where structure is used to reduce the likelihood of aircraft meeting and, consequently, the likelihood of conflicts. Segmenting operating traffic into multiple altitude layers reduces both the number of conflicts and the number of losses of minimum separation. Moreover, these rules allow for prevention of (near-)head-on conflicts, which would otherwise be impossible to resolve when heading variation for conflict resolution is not possible.

The improvement in safety comes at the cost of decreasing efficiency, as aircraft must now add transition between altitude layers to their route. However, the decrease in efficiency was small compared to the reduction in the number of losses of separation. Ultimately, improving safety increases the number of aircraft allowed into the airspace. Thus, heading-altitude rules are a good option from an operational perspective.

3.9.3. VARIABLE SPEED LIMIT WITH REINFORCEMENT LEARNING
Experimental results have shown that DDPG-based control of the maximum speeds allowed in sections where vertical transitions are taking place, reduces losses of minimum

separation. However, the benefit of variable speed limits is dramatically limited by the following:

- The compliance rate of 90% already cancels out the benefit of employing speed limits. Consequently, the necessary infrastructure should be in place to ensure that the aircraft can identify and react correctly to these variable speed limits.
- Training in a specific traffic density proved somewhat inefficient for higher densities. The RL agent should at least be trained at the highest traffic density expected in actual operations. It may also be that different traffic densities require different resolution strategies, as also hypothesised in the Metropolis project [13]. In this case, the RL method must learn different responses per complexity of emergent behaviour resulting from increasing traffic densities.

The excerpt of actions chosen by the RL method during one training episode shows a recommendation of the same speed value for the majority of the episode. We assume that this is due to the following reasons:

- Aircraft were able to climb/descend at any point, setting variable speed sections in close proximity. A homogeneous maximum speed value between all sections proved beneficial.
- The reward values were based on the efficiency of conflict resolution. Having aircraft (rapidly) accelerate greatly reduces the efficiency of conflict resolution, as it increases uncertainty regarding intruders' trajectory propagation.
- A uniform distribution of the traffic density was favoured to establish a relation between the allowed traffic density and the resulting safety level. Throughout one episode, the number of instantaneous aircraft is expected to remain (almost) constant, with variations resulting only from conflict resolution and/or randomisation of trajectories.

Previous research [179, 195, 196] commonly employed highway sections far apart. Thus, these do not have as much influence on each other. Additionally, traffic variation was more pronounced (off-peak vs. peak traffic). Additionally, in a real-case scenario, vehicles slow down to a halt to avoid collisions. In these cases, lower maximum speeds are applied to limit frequent speed breaks. This behaviour is not present in our simulations, and thus the RL method is free to favour higher speeds, which optimise traffic outflow. From Wu [179], we learnt that the maximum speed variability is influenced both by the reward formulation and by the traffic scenario in the lane. We advise future research to focus on the validation of VSL behaviour with different airspace rules (e.g., predefined, fixed climb/descent points; non-uniform traffic scenarios) for a better understanding of the relation between airspace properties and speed control.

3.9.4. ADVICE FOR FUTURE WORK

In this work, a DDPG method was employed. As seen in previous research, this method showed a fast convergence to an optimal solution. However, previous research has also shown that it is sensitive to unstable dynamics [164]. This should be taken into account when applying to different types of agents. In terms of further improvements with the reinforcement learning method, the following is also advised:

- Exploring more powerful states and reward formulations.

- Exploring different time periods for the duration of a maximum speed in a section. Duration may be based instead on observable changes in the traffic scenario in the section.
- The current implementation is unaware of the congestion that is building up some distance ahead. Greater observability over the environment could be obtained by adding knowledge within a larger surrounding radius to the state formulation. Such a situation introduces more complexity to the system, but should be considered in favour of a more homogeneous traffic situation throughout the entire environment.
- Further testing with more heterogeneous environments (e.g., different aircraft types, different performance limits, different separation between layers, different climbing/descending rates, different minimum separation distances).

Finally, when employing a multi-layer structure, most of the LoSs result from interactions between cruising and climbing/descending aircraft. Speed-based conflict resolution is not sufficient to defend against simultaneous vertical and horizontal conflicts. More operating rules can be added to the environment to improve safety between cruising and climbing/descending aircraft. For example: (1) airspace structuring can be extended to warrant sufficient space for vertical resolution manoeuvres; (2) setting multiple steps during climb/descent, in order to delay the final approach in case the upcoming layer is too congested.

3.10. CONCLUSIONS

This chapter looks at enabling the safe introduction of drone operations into urban airspace. The results show that the separation of traffic into different altitude layers by employing heading-altitude rules greatly reduced the total number of conflicts and losses of minimum separation. With this structure, interactions between cruising and climbing/descending aircraft should be the main focus in order to improve safety. Training a reinforcement learning (RL) agent to apply variable speed limits (VSL) enabled a more homogeneous traffic situation during the layer transition phase. When aircraft fully comply with these speed limits, these increase the distance between aircraft, reducing the total number of violations of minimum separation.

As traffic densities increase, so does the complexity of emergent behaviour from neighbouring aircraft. In these cases, simple sets of rules and analytical methods implemented by common conflict detection and resolution methods are no longer sufficient. In addition to VSL, future work may also consider the use of RL to improve the structure of the operational environment. The number of traffic layers, and the heading ranges allowed in each, can potentially be defined by an RL agent. Additionally, movement within the transition layers can also be further enhanced. For example, the implementation of several steps during climb/descent, delay of the final approach to the main traffic lane, can reduce the likelihood of cruising and climbing/descending aircraft meeting in conflict. Finally, the research presented here can be extended to more competitive operational environments, in terms of differences in performance limits, as well as preference for efficiency over safety.

4

USING REINFORCEMENT LEARNING IN LAYERED AIRSPACE TO IMPROVE LAYER CHANGE DECISION

Chapter 3 concluded that merging conflicts severely affect safety within a layered airspace. First, simultaneous vertical and horizontal conflicts severely hinder the efficacy of conflict resolution. Second, merging actions can force a conflict chain reaction where the follower aircraft have to readjust their speed to avoid collision. Chapter 3 proposed to use reinforcement learning (RL) to improve the steps taken during climb/descent, or even to delay the final approach to a traffic lane until is it safe to merge. This chapter explores these hypotheses.

Two RL methods are developed: a decision-making module (Section 4.4.2), and a control-execution module (Section 4.4.3). The former issues a lane change command based on the planned route. The latter performs operational control to coordinate the longitude and vertical movement of the aircraft for a safe merging manoeuvre. These two methods are tested independently and together.

Cover-to-cover readers may chose to skip Section 4.4.1, which describes the theoretical background of a Deep Deterministic Policy Gradient (DDPG) algorithm. This is very similar to its counterpart in previous Chapter 3.

4

ABSTRACT

Current predictions for future operations with drones estimate traffic densities orders of magnitude higher than any observed in manned aviation. Such densities call for further research and innovation, in particular, into conflict detection and resolution without the need for human intervention. The layered airspace concept, where aircraft are separated per vertical layer according to their heading, has been widely researched and proven to increase traffic capacity. However, aircraft traversing between layers do not benefit from this separation and alignment effect. As a result, interactions between climbing/descending and cruising aircraft can lead to a large increase in conflicts and intrusions. This chapter looks into ways of reducing the impact of vertical transitions within the environment. We test two reinforcement learning methods: a decision-making module, and a control-execution module. The former issues a layer change command based on the planned route. The latter performs operational control to coordinate the longitude and vertical movement of the aircraft for a safe merging manoeuvre. The results show that reinforcement learning is capable of optimising an efficient driving policy for layer change manoeuvres, decreasing the number of conflicts and losses of minimum separation compared to manually defined navigation rules.

4.1. INTRODUCTION

The European Drones Outlook Study [197] estimates that as many as 400 k drones will be operating in the airspace by 2050. Moreover, this study underlines the need to further develop and validate current conflict detection and resolution (CD&R) for high-density operating environments without the need for human intervention. The use of machine learning in tactical CD&R is highlighted as a potential tool to support advanced and scalable U-space services. The present work aids this research by developing reinforcement learning modules that decrease conflict rate and severity for an unmanned aviation operation in an urban environment.

In the current study, we employ a layered airspace, a concept developed by the Metropolis project [175], which separates traffic vertically based on their heading. The separation of the existent traffic density into smaller groups of aircraft travelling in similar directions helps reduce occurrences of conflicts and losses of minimum separation (LoSs) during the cruising phase. However, in an environment where aircraft cannot fly a straight line from the origin to destination, changes in heading force the vertical deviations to a different layer where the new direction is allowed. These vertically manoeuvring aircraft may have to cross through multiple layers of cruising aircraft, potentially running into multiple conflicts and intrusions.

Using reinforcement learning (RL) to improve lane change decision-making has been widely used with road vehicles [198, 199]. Optimal lane selection and longitudinal/lateral merging control can lead to better separation of agents, preventing traffic flow disruptions and collisions. Urban air traffic has several similarities with road traffic that justify exploring machine learning techniques that have been successfully applied in the latter. First, unmanned aviation is set to follow road infrastructure. Thus, the effects of the environment topology on traffic agglomeration are similar in both cases. Highway lane merging is comparable to layer merging in a layered airspace: (1) aircraft must also keep a

minimum separation distance from each other; (2) both types of agents prefer to remain close to their desired cruising speed so as not to increase travel time. Thus, we apply these same methods to a layered airspace environment. First, a decision-making module determines the layer that the aircraft should move into based on the cruising traffic and the distance to the next turning point. Second, a control-execution module decides on the best longitudinal/vertical control for a safe merging manoeuvre. However, there are remarkable differences between drones and road vehicles; the latter can become stationary, contrary to (most) drones. Additionally, in aviation, minimum separation distances are typically larger. These challenges will be further examined in this work.

Experiments are conducted with the open-source, multi-agent ATC simulation tool BlueSky [25]. During flight, aircraft follow a pre-planned route avoiding collision with the static surrounding infrastructure. To avoid LoSs between operating aircraft, all employ the conflict resolution method Modified Voltage Potential (MVP) [15]. Finally, the RL modules for the layer change procedure make use of the Deep Deterministic Policy Gradient (DDPG) model, as created by Lillicrap [163]. The operational efficiency of these modules, both individually and when working together, is directly compared with previous analytical rules for layer change behaviour.

Section 4.3 describes the characteristics of a layered airspace in more detail and how it is used in the simulation environment. This information is necessary to better understand how layer change behaviour is set with reinforcement learning, as specified in Section 4.4. Section 4.5 further details the experiment herein performed, and Section 4.6, the hypotheses considered. Section 4.7 shows the results of the experiments, comparing usage of the decision-making and control-execution RL modules to the baseline navigation rules. Finally, Sections 4.8 and 4.9 present the discussion and conclusion, respectively.

4.2. Related Work

Given the interdisciplinary nature of this work, this section analyses the state-of-the-art in two different areas. First, we go over how the RL methods employed in this work have been used in previous research related to road vehicles. Second, we describe the main methods used to improve safety in a layered airspace operational environment, especially regarding layer change decisions.

RL has been widely applied to road vehicles. The research includes, but is not limited to, controlling traffic flow to prevent agglomeration of traffic [200, 201], implementing velocity speed limits resulting in a more homogeneous traffic situation [157, 158], and ensuring minimum distance gaps between vehicles during lane change [198, 199, 202, 203]. We focus on the latter. Wang [198] showed that an RL-based vehicle agent was capable of successfully learning a lane change policy and ensuring a minimum safety distance under current speeds. Hoel [199] developed a deep Q-Network agent that matched or surpassed the performance of hand-crafted rules and emphasised that, rather than depending on rules laboriously created by domain experts, RL can create a much larger set of rules adapted to a multitude of different traffic situations. Alizadeh [202] showed that RL can adapt to the performance limits of each individual vehicle, achieving better performance within uncertain and stochastic environments than hand-crafted methods. Finally, Shi [203] demonstrated that an RL method can smoothly move a vehicle towards the target lane.

The layered airspace concept was first introduced by the Metropolis project [175]. In previous work, the authors point out that the safety benefits of this concept apply only to the cruising phase. Transitions between vertical layers can trigger a substantial number of merging conflicts, thus cancelling out a large part of the benefits gained from airspace structuring. Recently, a hand-crafted method for improving safety during merging manoeuvres was developed by Doole [204]. Results show that, with a limited number of rules, it is difficult to create solutions that can defend against the different topology of every street, as well as the relative relationship between the merging and cruising aircraft in different conflict geometries. These are comparable to the limitations previously seen with hand-crafted rules in lane change manoeuvres with road vehicles. Thus, we attempt to apply the same solutions that researchers found in that field. To the best of the author's knowledge, this is the first time that RL methods, previously successfully applied to lane change decision, are applied to layer merging in an aviation environment. Nevertheless, several questions remain on whether it is possible to translate the success of RL methods for road vehicles to aviation. The present work adds to this discussion.

4.3. LAYERED AIRSPACE DESIGN

Operating in an urban environment raises several challenges. First, aircraft must avoid collision with the surrounding urban infrastructure. Although detection of edges of static obstacles is possible through instrumentation, the only way to make sure that aircraft follow the shortest path towards the destination, or even that they do not end up in a closed space when following the edge of an obstacle, is by setting a pre-defined route based on the known characteristics of the environment. Second, conflict resolution manoeuvres must be adapted towards respecting the borders of the static obstacles. To guarantee that knock-on effects of successive manoeuvres do not lead to collisions, especially near non-uniform static obstacles, conflict resolution is limited to speed and altitude variation.

Limiting the freedom of conflict resolution manoeuvres naturally limits the number of conflict geometries that aircraft are capable of successfully resolving. The focus must then be put on additional elements that decrease conflict rate and severity. One of these elements is the structure of the airspace, which directly influences the likelihood of aircraft meeting in conflict. The Metropolis project has shown that a layered airspace structure considerably reduces the rate of conflicts [175]. Two effects contribute to this reduction: (1) segmentation: the total traffic density is divided into groups of aircraft allocated in different altitude layers; (2) alignment: the groups are divided per aircraft's heading, enforcing a degree of alignment between aircraft, which decreases the likelihood of conflicts in each layer.

4.3.1. SIMULATED ENVIRONMENT

The urban operational area is built using the Open Street Map networks (OSMnx) python library [167], an open-source tool for street network analysis. We use an excerpt from the San Francisco Area, representing an orthogonal street layout with a total area of $1.708\,\text{NM}^2$, as shown in Figure 4.1. Note that the RL method herein developed could, in theory, be

used in any environment. We make use of an orthogonal layout for simplification, as non-orthogonal layouts typically have a high number of conflicts associated with merging streets and non-regular street shapes. This simplification allows us to focus on the conflicts resulting from vertical deviations. The OSMnx library returns a set of nodes, with two adjacent nodes defining the edges of a road. A flight route is formed by connecting adjacent nodes that form a road. To reduce complexity, an intersection is considered to have at most four connecting roads. Each road is unidirectional per altitude level.

We allow directions per altitude, as defined in Figure 4.2. In conventional aviation, temporary altitude layers are often used as a level-off at an intermediate flight level along a climb or descent to avoid conflicts [205]. In our urban airspace, we apply the same concept: for each direction, three vertical layers exist, with increasing altitude. These are comparable to lanes on a highway. Each layer may adopt different uses for optimising cruising and turning. For example, following the rules of a highway, the middle layer may be used for longer cruising while the 1st and 3rd are used by aircraft about to turn in the directions below and above, respectively.
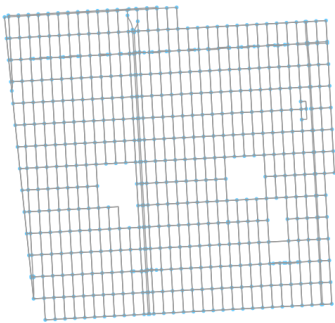

Figure 4.1: Map of the urban environment used in this work. Data obtained from the OSMnx python library [167].
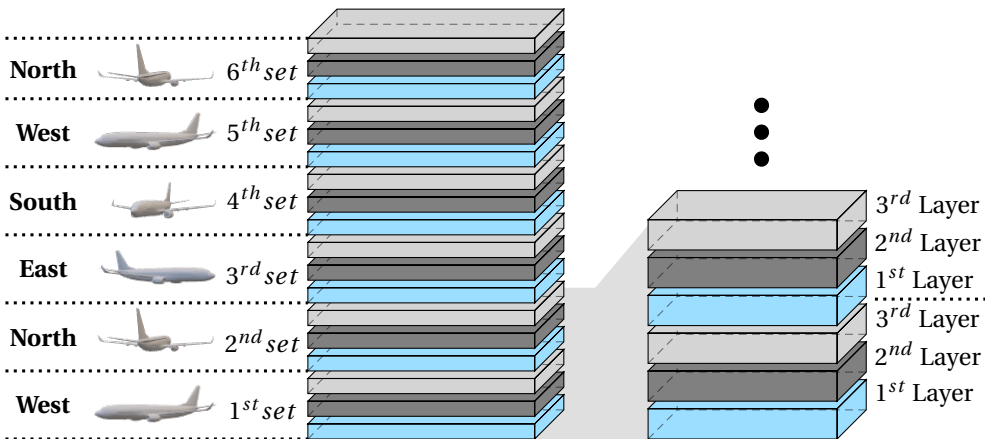

Figure 4.2: Altitude sets employed in this study. All layers have a height of 15 ft. A margin of 5 ft between the layers is used to prevent false conflicts.

4.4. LAYER CHANGE BEHAVIOUR WITH REINF. LEARNING

Research into automated lane-changing manoeuvres with road vehicles can broadly be divided into two functional categories. First, a decision-making module determines which layer the agent should move into and emits a lane change command. Second, a control-execution module receives this command and coordinates the longitudinal and lateral movement of the vehicle for an efficient lane merging manoeuvre [198]. Figure 4.3 depicts the decisions taken by each module, translated to an aviation environment.
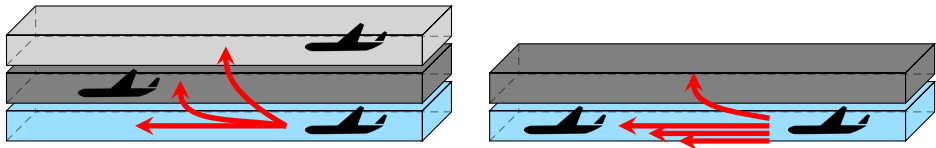
(a) Decision-making module: issues layer change commands based on the planned route and the current traffic scenario.

(b) Control-execution module: longitudinal and vertical coordinates control during vertical manoeuvre based on the traffic surrounding the ownship.

Figure 4.3: Visual depiction of the range of the responsibilities of each reinforcement learning module.

Figure 4.4 depicts a high-level functioning of the decision-making and the control-execution modules. The decision-making module is called upon whenever a new aircraft is created (and thus requires a starting layer) or whenever an aircraft is about to perform a turn, which results in a vertical deviation to a new set of layers where its new heading is allowed. When this module is used independently, the layer change action is immediately performed. In turn, when the control execution module is used independently, the target layer is dictated by the baseline rules (see Section 4.5.4 for more detail). When the two modules are used together, the decision-making module decides upon the target layer, and the control-execution modules can decide whether to merge immediately towards the target layer or to delay the action.

In a setting with an extremely high number of agents, as is the case with the expected traffic densities for unmanned aviation, representing the full state of the environment is too complex to train an RL method within an acceptable amount of time. Moreover, we assume that, in a real-world implementation, each aircraft would only (have to) be aware of its immediate surroundings. However, during training in a simulated environment, we have access to additional information. Thus, although the policies of the modules are based only on the surrounding information and executed in a decentralised manner, during training, the reward is based on a larger amount of information, specifically conflict/LoSs in each layer. Furthermore, each module should be able to learn an optimal policy independently of the other module. In theory, when used together, their improvements in the airspace should accumulate. In practice, it may be that actions of one module modify the environment in a way that reduces the efficacy of the actions of the other module. Past research has often focused on one of the modules; their conjugation is normally not tested. Thus, it is of interest to examine how these two modules work together.
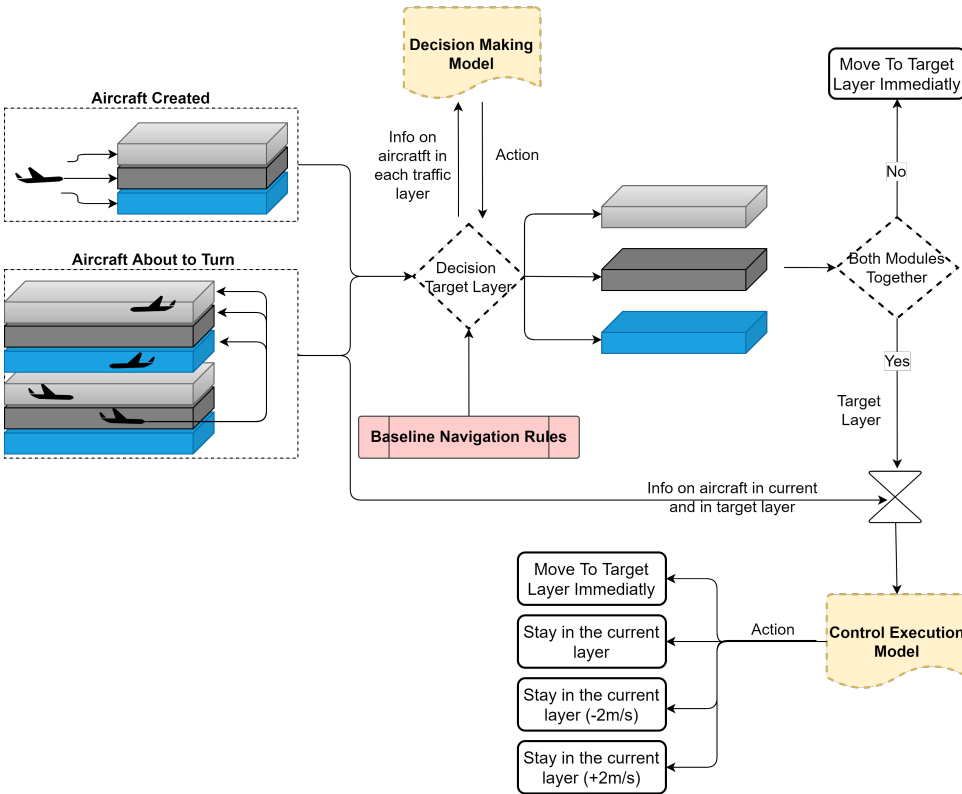
Figure 4.4: High level diagram of the functioning of the decision-making and the control-execution modules, both independently and together.

4.4.1. REINFORCEMENT LEARNING ALGORITHM

An RL method consists of an agent that interacts with an environment E in discrete timesteps. At each timestep, the agent receives the current state s of the environment and performs an action a in accordance, for which it receives a reward s_t. An agent's behaviour is defined by a policy, π, which maps states to a probability distribution over the available actions. The goal is to learn a policy which maximises the reward. Many RL algorithms have been researched in terms of defining the expected reward following the action a. In this work, we used the deep deterministic policy gradient (DDPG), defined by Lillicrap [163].

Policy gradient algorithms first evaluate the policy and then follow the policy gradient to maximise performance. DDPG is a deterministic actor–critic policy gradient algorithm designed to handle continuous and high-dimensional state and action spaces. It has been proven to outperform other RL algorithms in environments with stable dynamics [164]. However, it can become unstable, being particularly sensitive to reward scale settings [188, 189]. The pseudo-code for DDPG is displayed in Algorithm 4.1.

DDPG uses an actor–critic architecture. The actor produces an action given the current state of the environment. The critic estimates the value of any given state, which

is used to update the preference for the executed action. DDPG uses two neural networks, one for the actor and one for the critic. The actor function $\mu(s|\theta^\mu)$ (also called policy) specifies the output action a as a function of the input (i.e., the current state s of the environment) in the direction suggested by the critic. The critic $Q(s, a|\theta^Q)$ evaluates the actor's policy by estimating the state–action value of the current policy. It evaluates the new state to determine whether it is better or worse than expected. The critic network is updated from the gradients obtained from a temporal-difference error signal from each time step. The output of the critic drives learning in both the actor and the critic. θ^μ and θ^Q represent the weights of each network. Updating the actor and critic neural network weights with the values calculated by the networks may lead to divergence. As a result, target networks are used to generate the targets. The target networks are time-delayed copies of their original networks, $\mu'(s|\theta^{\mu'})$ and target critic $Q(s', a|\theta^{Q'})$ that slowly track the learnt networks. All hidden neural networks use the non-sigmoidal rectified linear unit (ReLU) activation function, as this has been shown to outperform other functions in statistical performance and computational cost [190].

Algorithm 4.1 Deep Deterministic Policy Gradient

Initialise critic $Q(s|a^\mu)$ and actor $\mu(s|\theta^\mu)$ networks, replay buffer R, and action exploration
for all episodes **do**
    while episode not ended **do**
        Select action a_t according to the state s_t from environment and the *actor* network
        Perform action a_t in the environment and receive reward r_t and new state s_{t+1}
        Store transition (s_t, a_t, r_t, s_{t+1}) in replay buffer R
        Sample a random mini-batch of N transitions from R
        Update *critic* by minimising the loss
        Update *actor* policy using the sample policy gradient
        Update *target* networks
    end while
    Reset the environment
end for

The neural network parameters used in our experimental results are based on Lillicrap [163]. Other hyperparameters may be used; nevertheless, the parameters defined by Lillicrap [163] has shown promising results. Experience replay is used in order to improve the independence of samples in the input batch. Past experiences are stored in a replay buffer, a finite-sized cache R. At each timestamp, the actor and critic are updated by sampling data from this buffer. However, if the replay buffer becomes full, the oldest samples are discarded. Finally, exploration noise is used in order to promote the exploration of the environment; an Ornstein–Uhlenbeck process [191] is used in parallel to the authors of the DDPG model.

4.4.2. DECISION-MAKING MODULE

The decision-making module chooses the layer that the ownership should move into based on the current traffic scenario and planned route. This decision is made when an aircraft enters the airspace at the beginning of its flight and when a heading turn requires a deviation to a different layer where the new direction is allowed. It should be noted that this module could potentially be used for aircraft to move to adjacent intermediate layers during cruising in order to overtake a slower leading aircraft, for example. However, this

would heavily increase the complexity of the learning environment. When evaluating the results of a layer change action, the module can only learn if most of the alteration to the environment was caused by that one action. With multiple simultaneous deviations, the change to the environment is a result of the impact of all actions combined. It would, thus, be near impossible to connect an action to a direct alteration of the environment.

STATE

The state input must contain the necessary data for the module to be able to successfully determine an optimal solution. Ideally, the complete environment would be represented. However, a large state formation leads to a large number of possible states and state-action combinations. In practice, such results in the RL method having an exponential number of solutions to test, which may increase training time to an impracticable amount. Thus, we focus on the information we find essential: the current state of the ownship, the number of aircraft currently in each layer, the time to loss of separation to the leader and follower aircraft (if the ownship would move to the layer in this longitudinal/lateral position), and the number of waypoints until the next turn as per the planned route (see Table 4.1).

Table 4.1: State formulation for the decision-making module.

State	Element
s_0	Ownship's current speed
s_1	Ownship's current layer
s_2	Number of aircraft in 1st layer
s_3	TLoS to front aircraft in 1st layer
s_4	TLoS to back aircraft in 1st layer
s_5	Number of aircraft in 2nd layer
s_6	TLoS to front aircraft in 2nd layer
s_7	TLoS to back aircraft in 2nd layer
s_8	Number of aircraft in 3rd layer
s_9	TLoS to front aircraft in 3rd layer
s_{10}	TLoS to back aircraft in 3rd layer
s_{11}	Number of waypoints until next turn

We consider the relation between the leader and follower aircraft to be the most important information. The distance to the surrounding aircraft will, at least, account for the LoSs directly suffered by the ownship. The module must decide whether the gap available for merging is adequate to ensure a minimum safety distance at the current speed. Moreover, the module should also give preference to layers with fewer cruising aircraft. A merging manoeuvre can cause the follower aircraft to reduce its speed to prevent getting too close to the ownship, for example. When there is not enough distance between aircraft in the layer, this deceleration can cause a propagation of conflicts as aircraft slow down in succession to prevent becoming too close to the leader aircraft.

ACTION

The module determines the action to be performed for the current state. We use a softmax activation function that turns an input vector into an output vector with values between

zero and one, which sum to one. These values represent a probability distribution and are used to define which layer the ownship should move into, as per Table 4.2. Staying in the same layer is also possible.

As described in Section 4.3.1, there are three layers for each direction set. These decrease conflict likelihood and speed heterogeneity during turns. However, they can be used with a different rationale. As per Figure 4.2, the first layer has the closest access to the direction just below, and the third layer is the closest to the direction above. An aircraft in the first layer, which needs to turn into the direction just above the current direction, will need to cross the second and third layers; an aircraft in the third layer would only have to climb towards the top layer.

Table 4.2: Action formulation for the decision-making module. The layers increase in altitude from the 1st to the 3rd layer. For a visual representation, see Figure 4.2.

Action	Element
a_0	Move to 1st layer
a_1	Move to 2nd layer
a_2	Move to 3rd layer

REWARD

The objective is for the module to favour layer change decisions that reduce the likelihood of LoSs. However, often the number of LoSs, especially in environments where CD&R is applied, is too scarce to provide enough information for the module to train within an optimal amount of time. Thus, we consider conflicts as well: the module receives a reward of -1 for each conflict, and of -10 for each LoS. As conflicts represent future detected LoSs, reducing the total number of conflicts is expected to also reduce the total number of LoSs. Although not an ideal reward formulation, as this should be as simple as possible, it was found necessary for the module to converge towards optimal decisions. Note that it is the relative relation between the values, -1 and -10, not the absolute values, that influence the behaviour of the RL method. The method follows the highest rewards. Nevertheless, a different weight relation could have been applied. These weights were found to be the best empirical values for the particular operational environment/traffic scenarios herein employed. Nevertheless, it was taken into consideration that LoSs are the paramount values and should (heavily) outweigh the value of each conflict to make sure that the RL method does not opt for having one LoS in favour of preventing a small number of conflicts.

Moreover, the conflicts and LoSs included in the reward are not only the ones that the ownship (which performed the action) is involved in. The reward will also consider the effect on the layer that the ownship moves into. Such was found to be an important factor in guaranteeing that the module converged to optimal solutions, which have a positive effect on the global environment. It may be that the ownship does not suffer a conflict/LoS situation due to the follower aircraft decreasing its speed, or the leader aircraft increasing its speed, in order to keep a minimum distance from the ownship. However, these changes in speed disrupt the traffic flow; successive acceleration/deceleration over the following aircraft may result in a LoS further down the layer.

Finally, an important question related to the decision-making module is whether it should consider conflicts/LoSs occurring in the intermediate layers between the initial and target layer, as depicted in Figure 4.5. On the one hand, considering intermediate conflict/LoSs may decrease the total number of conflicts/LoSs during layer change actions. On the other hand, it may be considered that the decision-making module should focus only on finding the best target layer for cruising and not having this decision hindered over favouring nearer layers (i.e., that do not require crossing a great vertical distance). Within the total duration of their flight, aircraft will spend more time cruising than vertically manoeuvring between layers (which the module is unaware of). Thus, selecting the optimal cruising layer may be better for the global number of conflicts/LoSs. In this case, the control-execution module is then solely responsible for decreasing conflicts/LoSs encountered during transition between the initial and target layers. The effects of considering intermediate conflicts/LoSs will be analysed with the experimental simulations.
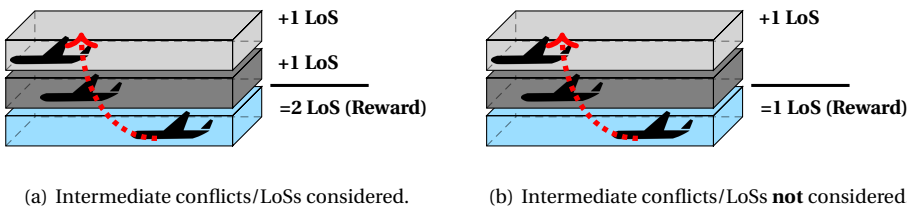

(a) Intermediate conflicts/LoSs considered.   (b) Intermediate conflicts/LoSs **not** considered.

Figure 4.5: Difference in the final reward received by the decision-making module.

4.4.3. Control-Execution Module

Once a layer change decision is produced, the control-execution module takes over to guide the ownship towards the best action to prevent conflicts/LoSs with aircraft both at the current layer and target layer. The existing gap on both layers is evaluated, and the RL module may choose to move the ownship vertically towards the target layer or to modify its current state in order to improve the gap in the future. When the module decides on an action that keeps the ownship in the current layer, this action is considered to have a duration of five seconds. After these five seconds, the execution module is called again to decide which longitudinal/vertical movement the ownship should follow now. This process is repeated until the ownship reaches its target layer.

State

The state formulation, as shown in Table 4.3, focuses on giving enough information to the module to decide whether the gaps in the current and target layers are sufficient to guarantee minimum separation between the leader and follower aircraft. Moreover, the number of layers until the target may influence how long the ownship delays the move to the next layer, as it may affect the following merging actions that the ownship must perform until reaching the target layer.

Action

This module also uses a softmax activation function for classification. As displayed in Table 4.4, the control-execution module controls the ownship's longitudinal and vertical

Table 4.3: State formulation for the control-execution module.

State	Element
s_0	Ownship's current speed
s_1	Relative heading of current layer
s_2	TLoS to front aircraft in current layer
s_3	TLoS to back aircraft in current layer
s_4	Relative heading of target layer
s_5	TLoS to front aircraft in target layer
s_6	TLoS to back aircraft in target layer
s_7	Number of layers until target layer

4

movements. When merging into the target layer is not yet safe, the module may opt instead for: (1) accelerating the ownship, (2) decelerating the ownship, or (3) keeping the same speed, while remaining in its current layer. Naturally, this decision must also consider the time to LoS with the neighbouring aircraft in the current layer.

Table 4.4: Action formulation for the control-execution module.

Action	Element
a_0	Stay in current layer, keep current speed
a_1	Stay in current layer, change speed: $+2\,\mathrm{m/s}$
a_2	Stay in current layer, change speed: $-2\,\mathrm{m/s}$
a_2	Move to target layer

Note that different speed change values could have been employed. Nevertheless, these should always take into account the performance limits of the operational vehicles. The main reason for this (low) speed change value was the acceleration performance of the simulated aircraft. At each timestep, there is a maximum state variation that an aircraft may achieve. With great state variations, the reward received by the RL method may not be based on the results with the state output by the method but, instead, on the maximum variation that the aircraft was able to achieve within the available time. This may make it harder for the RL method to correctly relate actions to expected rewards.

REWARD
The reward received by the module is based not only on the conflicts/LoSs suffered by the module but also on the immediate effect on the layer occupied by the ownship (this is the target layer when the module moves the ownship vertically, or the current layer otherwise). Similarly to the decision-making module, the control-execution module receives a reward of -1 for each conflict, and of -10 for each LoS. Additionally, $+1$ is given for a completed merging manoeuvre, guaranteeing that in a safe situation, the module will favour moving to the target layer.

4.5. EXPERIMENT: SAFETY OPTIMISED LAYER CHANGE
The following subsections describe the properties of the performed experiment. Note that the experiment is divided between the training and testing phases. First, the two RL

modules are trained continuously with a fixed training scenario. Second, they are tested with a set of previously unknown traffic scenarios; the performance of these modules is directly compared to baseline navigation rules.

4.5.1. SCENARIO DESIGN

Aircraft spawn on the edge of the simulation area in a layer that allows for the initial direction. Origin points are separated by at least a minimum separation distance to avoid conflicts between just spawned aircraft. Each route is formed by connecting adjacent nodes of the map. Aircraft are removed from the experiment when they move away from an edge node, once they finish their route. Different trajectories will be tested, with the objective of evaluating the performance of the RL modules in multiple situations. The following settings are defined per traffic scenario:

- Heading distribution: the heading adopted by the simulated aircraft. Having the majority of the aircraft following the same direction leads to an agglomeration of a high number of aircraft in one layer, which likely decreases the average distance between aircraft and, in turn, increases the number of conflicts. A more uniform heading distribution increases the distribution of aircraft per the airspace, reducing the likelihood of conflicts. Using different heading range distributions tests the capacity of the RL modules to successfully segment different traffic scenarios over the available airspace. The heading distribution (per percentage) is defined in Table 4.5. In traffic scenario #1, for example, 100% of the aircraft travel East. In practice, if 100 aircraft are simulated, all 100 will be travelling East. In comparison, traffic scenario #15 has a uniform distribution: with 100 aircraft, 25 aircraft would travel east, 25 travel south, 25 travel west, and 25 travel north. Note that all aircraft start at a side of the map, which allows for a straight route towards their initial direction (e.g., an aircraft with initial direction east will start at the west end of the map).

Table 4.5: A total of 15 different heading distributions are used.

Traffic Scenario:	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12	#13	#14	#15
East (E):	100	0	0	0	50	50	50	0	0	0	33	33	33	0	25
South (S):	0	100	0	0	50	0	0	50	50	0	33	33	0	33	25
West (W):	0	0	100	0	0	50	0	50	0	50	33	0	33	33	25
North (N):	0	0	0	100	0	0	50	0	50	50	0	33	33	33	25

(% A/C Heading Distribution)

- Different number of turns: a turn in a layered airspace signifies a necessary change in the vertical layer. Thus, a different number of turns are used to elicit sufficient layer changes to analyse (1) the effect of a different number of vertical deviations in the environment, and (2) the ability of the modules to protect against successive changes in heading distribution. Five different turning settings are employed, as per Table 4.6. If there are no turns, aircraft will travel towards their initial direction throughout the complete route. For example, running traffic scenario #1 with turning option #A means that all aircraft travel East throughout the complete duration of the traffic scenario. In comparison, running traffic scenario #1 with turning

option #E signifies that each aircraft will perform five turns throughout their flight route. Thus, they all start their flights heading east but then change direction a total of five times. Note that a turn to the right from an aircraft with an initial direction east indicates that the aircraft will turn towards the direction south during its route. A turn to the left would result in this aircraft turning towards the north.

Table 4.6: A total of 5 different turning options aree used.

Turning Option	Number of Turns
#A	No Turns
#B	2 Turns to the Right
#C	4 Turns to the Right
#D	2 Turns to the Left
#E	4 Turns to the Left

Each heading distribution is performed five times with a different turning option, i.e., heading distributions #1 to #15 are each run with turning options #A to #E. In total, 75 traffic scenarios (15 heading distributions × 5 turning options) are run for each traffic density. A total of three different traffic densities are tested: low, medium, and high traffic densities. More detail on these is given in Section 4.5.4.

4.5.2. VEHICLE/AGENT CHARACTERISTICS

This work uses the open Air Traffic Simulator Bluesky [25] to test the efficacy of the layer change RL modules. Aircraft are defined per the performance characteristics of the DJI Mavic Pro drone model. Speed and mass were obtained from the manufacturer's data, and common conservative values were assumed for turn rate (max: 15°/s) and acceleration/breaking (1.0 kts/s). Aircraft have a preferred cruising speed of 30 kts. However, in line with their performance limits, aircraft must decrease their speed prior to a turn to ensure that the turning radius does not lead to a collision with any surrounding static obstacle. Once the aircraft has completed a turn, the aircraft will again accelerate towards its desired cruising speed. It is assumed that the aircraft have constant altitude and speed during a turn.

4.5.3. CONFLICT DETECTION AND RESOLUTION

This work employs a horizontal separation of 50 m, which is commonly used in works with unmanned aviation [59]. A vertical separation of 15 ft is assumed based on the dimension of the vertical layers. Conflicts are detected by linearly propagating the current state of all aircraft involved and determining if two aircraft will be closer than the minimum separation distance within a look-ahead time of 30 seconds. For conflict resolution, we employ the Modified Voltage Potential (MVP) method, as defined by Hoekstra [2, 15]. Once a conflict is found, MVP displaces the predicted future positions of both ownship and intruder at the closest point of approach (CPA) in the shortest way out of the protected zone of the intruder. More details on state-based detection and the resolution manoeuvres calculated by MVP can be obtained from previous work [206].

4.5.4. Independent Variables

During training, the only independent variable is the reward formulation for the decision-making module. During testing, different traffic densities are introduced to analyse how both RL modules perform at traffic densities they were not trained in. Additionally, we compare the use of decision-making and control-execution modules with baseline analytical rules.

Reward Formulation for the Decision-Making Module

In Section 4.4.2, it was mentioned that it is not clear whether considering intermediate conflicts will limit the ability of the module to select the best layer for the cruising phase. This module will be trained with and without considering intermediate conflicts. The results will be directly compared.

Using Reinforcement Learning vs. Baseline Analytical Rules

The effect of employing either the decision-making or control-execution module will be compared with resorting to baseline analytical rules. With the latter, aircraft initially always move into the first layer, which is the main cruising layer. The second layer is used for vertical conflict resolution. The third layer is used for deceleration and turning before moving to a different traffic layer with a different direction. This prevents conflicts resulting from heterogeneous speed situations caused by aircraft reducing speed in preparation for a turn. In this baseline situation, the aircraft immediately perform the layer change command.

Both the decision-making and the control-execution modules are first trained and tested individually to directly analyse their effect. When the decision-making module is tested alone, the aircraft follows its layer change commands and immediately performs them. Regarding the control-execution module, when tested individually, aircraft follow the layer change commands as defined by the baseline rules and perform/delay this manoeuvre as instructed by the control-execution module.

Traffic Density

Three traffic densities, in an increasing number of operating aircraft, are used. The exact values are shown in Table 4.7. At high densities, aircraft spend more than 10% of their flight time in conflict resolution mode [193]. Both RL modules are trained first in a medium traffic density and then tested with low, medium, and high traffic densities to assess their efficiency in lower/higher traffic densities.

Table 4.7: Traffic volume used in the experimental simulations (in 1 hour of simulation time). The range of results from different flight paths as the necessary time to traverse the environment is dependent on the initial direction(s) and number of turns.

	Low	Medium	High
Number of instantaneous aircraft (-)	50	100	150
Number of spawned aircraft (-)	242–1190	483–1972	721–2958

4.5.5. DEPENDENT VARIABLES

The effect of the RL modules on the environment is measured on multiple metrics: safety, stability, and efficiency. The first includes the occurrences and duration of conflicts and LoSs. Inclusion of RL modules in the operational environment should reduce these elements. Additionally, LoSs are evaluated on their severity according to how close aircraft get to each other:

$$LoS_{sev} = \frac{R - d_{CPA}}{R}. \tag{4.1}$$

Stability evaluates the secondary conflicts created by tactical conflict resolution manoeuvres. When free airspace is scarce, having aircraft move laterally and occupying a larger portion of the airspace often results in conflict chain reactions [175]. This effect has been measured using the Domino Effect Parameter (*DEP*) [151]:

$$DEP = \frac{n_{cfl}^{ON} - n_{cfl}^{OFF}}{n_{cfl}^{OFF}}, \tag{4.2}$$

where n_{cfl}^{ON} is the number of conflicts with CD&R ON, and n_{cfl}^{OFF} the number with CD&R OFF. Higher DEP values signify destabilising behaviours.

Finally, efficiency is evaluated in terms of the total distance travelled by the aircraft and the duration of the flight. Methods that do not result in a considerable increase in the path and/or the duration of the flight are considered more efficient.

4.6. EXPERIMENT: HYPOTHESES

4.6.1. EFFICACY OF THE DECISION-MAKING MODULE

It is hypothesised that the decision-making RL module decreases the number of conflicts/LoSs by segmenting aircraft optimally per the available layers, with special emphasis on scenarios where all aircraft are placed in the same layers (i.e., the scenarios where the majority of aircraft start from the same direction). Regarding the decision of whether to include conflicts/LoSs resulting from the ownship crossing the intermediate layers between the initial and target layer, it is hypothesised that not including them will lead to the module picking a more optimal cruising segmentation. Since aircraft spend more time cruising than manoeuvring vertically, it is expected that this will lead to a reduction in the global number of conflicts/LoSs.

4.6.2. EFFICACY OF THE CONTROL-EXECUTION MODULE

The control-execution RL module is hypothesised to decrease the number of conflicts/LoSs compared to a situation where aircraft simply move to the target layer when a layer change command is received. However, the effect of this module will only be noticeable in an environment where a high number of turns is expected.

4.6.3. WHEN THE TWO MODULES WORK TOGETHER

In theory, the best-case scenario is when both modules are used together. The decision-making module will output a layer-changing command towards the best cruising layer,

and the control-execution module will control the longitudinal and vertical moment of the ownship, making sure that the merging action is as safe as possible. In practise, it may be that one of these modules alters the environment in such a way that it decreases the efficiency of the other module. Nevertheless, the control-execution module is hypothesised to reduce the extreme LoSs cases resulting from layer change commands that cross multiple intermediate layers.

4.6.4. Effect of Traffic Densities

The RL modules will be tested within the same traffic density in which they were tested, and at lower and higher densities for comparison. Different traffic densities help analyse the capability of the modules to generalise to unseen and more complex multi-actor conflict geometries. It is hypothesised that the agents will perform better under the exact conditions in which they were trained in and that, under different conditions, the agents may be the least effective in higher traffic densities.

4.7. Experiment: Results

### 4.7.1. Training of the Reinforcement Learning Modules

Both RL modules, decision-making and control-execution, are first trained in a medium traffic density. Each module is trained repetitively on one traffic scenario; an episode corresponds to a repetition of this traffic scenario, which runs for 1 hour. Within one episode, each module is called thousands of times. Here, we focus on analysing the choices made by the modules. Only speed conflict resolution was added to the environment during training.

Safety Analysis

Figures 4.6 and 4.7 display the evolution of the actions chosen by the decision-making module throughout training. In each episode, the module is called upon when an aircraft is introduced into the environment to decide the layer that the aircraft is to be introduced in, and this occurs before a turn as the aircraft will have to change to a different layer set where the new direction is allowed.

In Figure 4.6, intermediate conflicts/LoSs are considered. As mentioned above, the graphs show the evolution of the decisions of the RL method during training. At the end of its training, in practise, the RL module 'discards' a layer, allocating aircraft mainly in the second and third layers. This division is optimal in decreasing intermediate conflicts/LoSs when aircraft are divided per the second and third layers according to their next turn. Aircraft in the third layer only climb one layer towards the next direction. Aircraft in the second layer can move to the direction below by descending two layers; however, the first layer does not contain cruising traffic, and thus, the ownship is not likely to run into conflicts/LoSs here. Finally, Figure 4.6(b) shows that this behaviour adopted by the module leads to a reduction in conflicts/LoSs per action.

Figure 4.7 shows the evolution of the decision-making module when intermediate conflicts are not added to the reward. Compared to Figure 4.6, this version of the module strongly prioritises proper segmentation of the aircraft per the available vertical space. Keeping the traffic density to a minimum in each layer helps reduce conflicts/LoSs during the cruising phase.
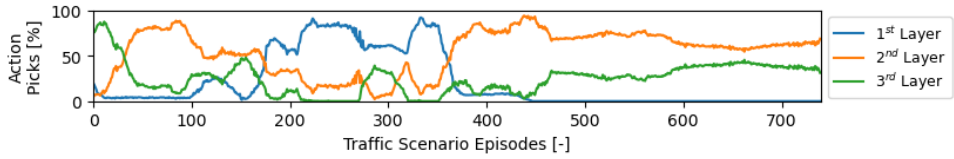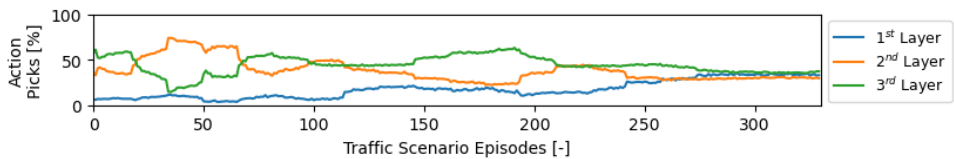
(a) Evolution of the balance between possible actions throughout episodes during training.
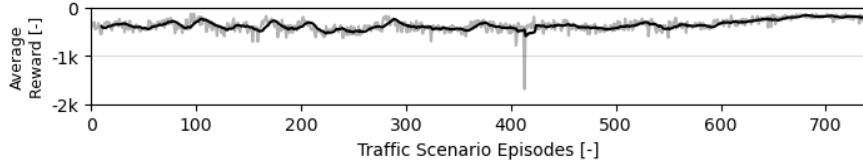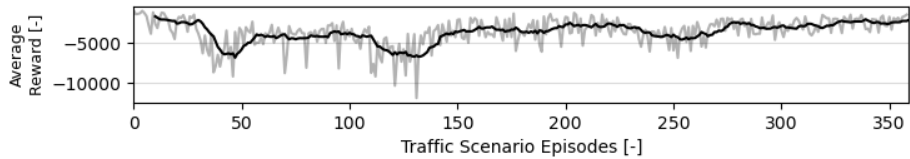

(b) Evolution of the average reward per action throughout episodes during training.

Figure 4.6: Evolution of the actions and rewards during the training of the decision-making module when intermediate conflicts/LoSs are considered. Roughly 3.8 M actions were performed.

(a) Evolution of the balance between possible actions throughout episodes during training.

(b) Evolution of the average reward per action throughout episodes during training.

Figure 4.7: Evolution of the actions and rewards during the training of the decision-making module when intermediate conflicts/LoSs are **not** considered. Roughly 1.7 M actions were performed.

Figure 4.8 displays the time to LoS to the leader and follower aircraft in the target layer. Figure 4.8(a) shows the results with the RL module trained without considering intermediate conflicts/LoSs. According to Figure 4.6, the module mainly uses two layers per set only. Thus, aircraft move to layers with higher traffic densities. In this case, the module prefers to move aircraft to layers where the time to loss of separation between the ownship and the surrounding aircraft is greater than 120 seconds. Figure 4.8(b), shows the results with the RL module trained considering intermediate conflicts/LoSs. Here, the traffic density is expected to be lower as the module prioritised segmentation per the three layers per set (see Figure 4.7). In this case, the module will still occasionally move to

a layer even if the time to LoS with the follower aircraft is below 60 seconds. This is likely due to the fact that, with fewer aircraft per layer, the follower aircraft has 'more space' to decelerate to avoid an LoS with the ownship, analogously to what occurs on a highway. This shows the relevance of looking into the effect of a merging action in the complete layer—aircraft consecutively breaking down to avoid conflicts may result in back-end conflicts.
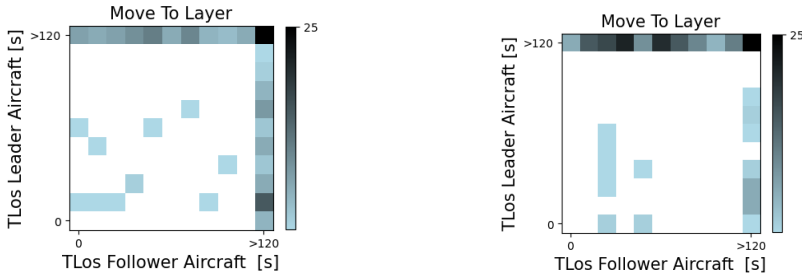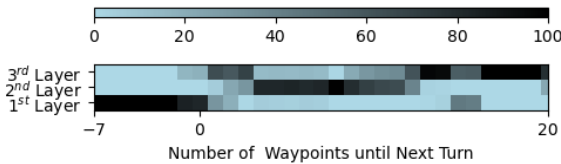

(a) Module trained considering all conflicts/LoSs.

(b) Module trained not considering intermediate conflicts/LoSs.

Figure 4.8: Time to loss of separation between the ownship and the leader and follower aircraft for the actions performed by the decision-making module.

Figure 4.9 shows the likelihood of aircraft being set on each layer according to the number of waypoints until their next turn. Negative waypoint values mean that the aircraft will descend to a different layer set, and positive values indicate a climb. The modules place aircraft that will climb in the third layer and aircraft that will descend in the lowest cruising layer (i.e., the 2nd layer in Figure 4.9(a) and the 1st layer in Figure 4.9(b)). This is an optimal choice, as aircraft are already closer to their next layer set in altitude.
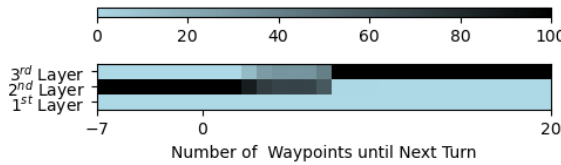

(a) Module trained considering all conflicts/LoSs.

(b) Module trained **not** considering intermediate conflicts/LoSs.

Figure 4.9: Likelihood (in percentage) of aircraft being set on each layer according to the number of waypoints until their next turn.

Figure 4.10 displays the actions chosen by the control-execution module. During training, this module is called upon when a layer change decision command is output based on the baseline navigation rules (see Section 4.5.4). The module opts for a move to the next layer about 95% of the time. Note that when the module decides to delay merging towards the target layer by selecting to stay in the current layer instead, the module will again be called after 5 seconds to check upon the viability of a merging manoeuvre. This process is repeated until the ownship moves into the target layer. Performing a 'move to the next layer' action 100% of the time would be the same as not having a control-execution module; the layer-changing command is always performed immediately. Thus, the main focus is when this module decides to 'delay' the merge, which hopefully decreases conflict/LoSs during vertical transitions. Figure 4.10 shows that the actions adopted by the module lead to a reduction in conflicts/LoSs.
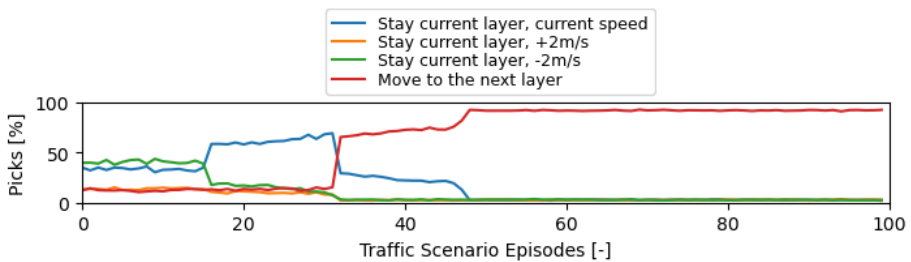
(a) Evolution of the balance between possible actions throughout episodes during training.
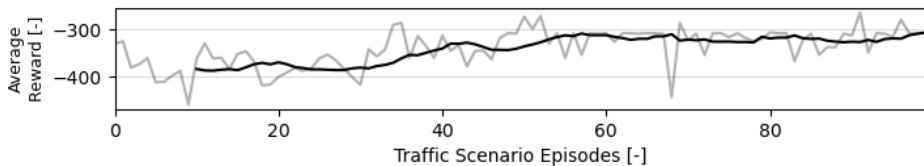

(b) Evolution of the average reward per action throughout episodes during training.

Figure 4.10: Evolution of the actions and rewards during the training of the control-execution module. Roughly 1.4 M actions were performed.

Figure 4.11 displays the environment status during each action of the control-execution module. Figure 4.11(a) shows the time to LoS to the leader and follower aircraft on the ownship's current layer. Figure 4.11(b) maps the time to LoS to the leader and follower aircraft on the target layer. First, the main motivator of whether to move to the next layer appears to be the distance between the leader and follower aircraft in the next layer. Nevertheless, on some occasions, the module will still move aircraft to the next layer when the follower aircraft is in close proximity (see darker points on the top left of the 'Move to Next Layer' action in Figure 4.11(b)). Other variables (such as TLoS to the neighbouring aircraft in the current layer, the ownship's speed, and the number of waypoints to the final target layer) also affect this decision. However, how these values combine for this decision is not clear when looking at them individually. Second, although small, there is a preference for accelerating when the follower aircraft is closer (see darker points on the

top left of the 'Stay Current Layer, +2 m/s' actions) and for decelerating when the leader is near (see darker points on the right side of the 'Stay Current Layer, −2 m/s' actions). Finally, similarly to the decision-making module, the control-execution module seems to prioritise a larger distance to the leader than to the follower aircraft.
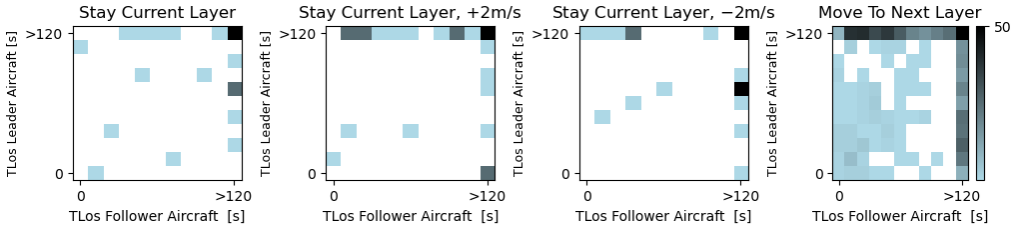
(a) Time to loss of separation between the ownship and the leader and follower aircraft in the current layer for each possible action.
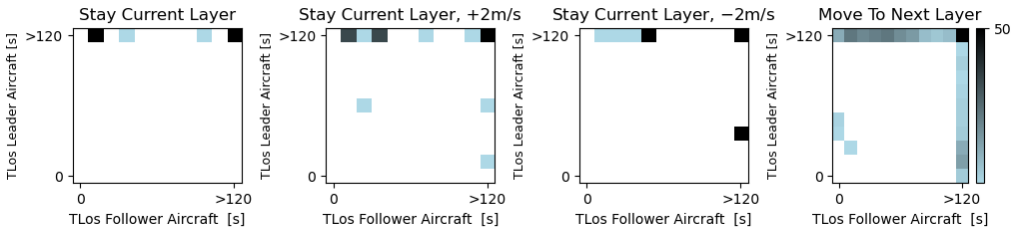
(b) Time to loss of separation between the ownship and the leader and follower aircraft in the target layer for each possible action.

Figure 4.11: Time to loss of separation between the ownship and the leader and follower aircraft.

4.7.2. TESTING OF THE REINFORCEMENT LEARNING MODULES

The RL modules are tested with a total of 225 traffic scenarios; 75 scenarios in each one of the traffic densities (i.e., low, medium, and high). These vary in the number of turns and initial direction(s), as previously described in Section 4.5.1. The RL modules were previously trained within a medium traffic density; it is interesting to see how they behave at lower and higher traffic densities. All testing episodes are different from the one in which the modules were trained. For each traffic scenario (i.e., combination of specific traffic density, initial direction(s), and number of turns), three repetitions with different flight trajectories are performed. Each traffic scenario ran for one hour. In all graphics, the 'baseline' comparison data corresponds to the traffic scenarios run with the analytical rules previously described in Section 4.5.4. Speed and vertical conflict resolution are performed during testing. When traffic scenarios have different trends, line graphs are used to show the results for all scenarios. When the trend is similar to all scenarios, boxplot graphs display the results for all traffic scenarios in each traffic density in favour of simplicity.

SAFETY ANALYSIS

Figure 4.12 shows the mean total number of pairwise conflicts. Both RL modules are tested independently and together; all decreased the total number of conflicts when compared to the baseline navigation rules. Figure 4.12(a) displays the difference between training the decision-making module with and without considering intermediate conflicts/LoSs. Although there was a small difference, the module trained without considering intermediate movements achieved a greater reduction in conflicts. Not considering intermediate conflicts has a negative impact locally, as merging actions will suffer more conflicts/LoSs. However, globally, the fact that the module focuses on efficient segmentation throughout the entire airspace becomes the most beneficial factor. This segmentation is especially relevant at higher traffic densities. Thus, at these densities, better traffic segmentation may outweigh reserving free space for vertical deconflicting manoeuvres.
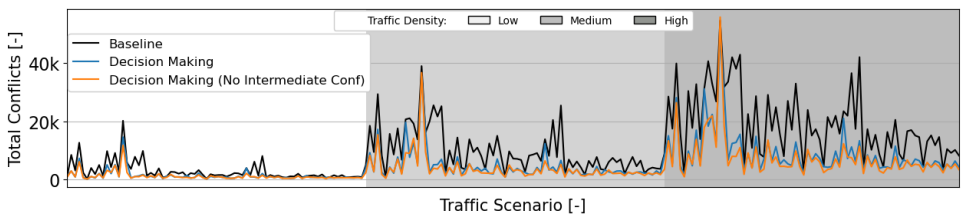
(a) Mean total number of pairwise conflicts during testing of the decision-making module. A comparison is made between the RL module when trained considering all conflicts/LoSs (in blue), and when no intermediate conflicts are considered (in orange).
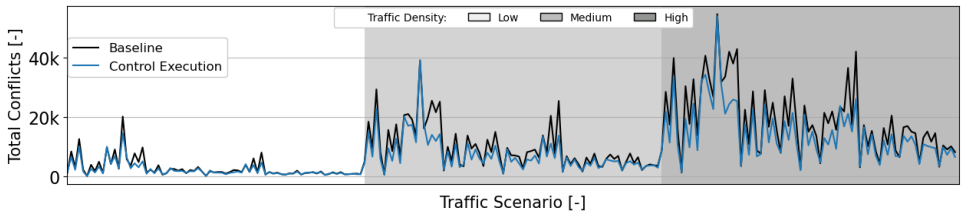
(b) Mean total number of pairwise conflicts during testing of the control-execution module.
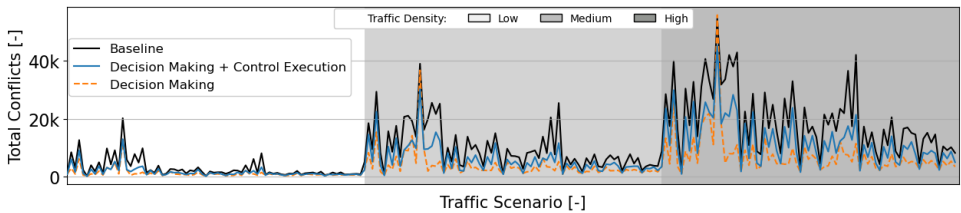
(c) Mean total number of pairwise conflicts during testing of the two RL modules together.

Figure 4.12: Mean total number of pairwise conflicts during the testing of the RL modules. All traffic densities have 75 traffic scenarios, with initial direction(s) and number of turns as defined in Section 4.5.1.

The test results for the control-execution module are shown in Figure 4.12(b). When used independently, this module receives layer change commands from the baseline rules. Its 'delay' actions (i.e., when it chooses to stay in the current layer instead of merging into the target layer) are able to reduce the total number of conflicts for all traffic scenarios and densities. Naturally, the module is only called when there are heading turns, so there is no impact in traffic scenarios with only straight flight routes. The impact is greater in traffic scenarios with a high number of turns. Initial hypotheses considered that the module would lose its effectiveness at high traffic densities due to more intruders and smaller distance gaps between aircraft. However, its influence is especially noticeable in these densities, where it can prevent a large number of conflicts/LoSs occurring due to merging manoeuvres within these small gaps.

Finally, Figure 4.12(c) displays the testing of both modules together. In this case, the decision-making module outputs a layer change command, which is received by the control-execution module. We employ the decision-making module trained without considering intermediate conflicts due to the best testing results. The combination of both modules has fewer conflicts than the baseline navigation rules. However, the combination of both modules increases the total number of conflicts for some traffic scenarios when compared to using the decision-making module alone with immediate merging upon the layer change command. The following safety graphs will show the results of decision-making next to the combination of both modules to analyse the reason for the worsening of the total number of conflicts.
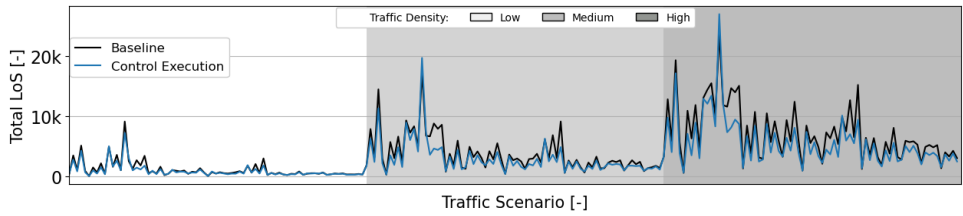
Figure 4.13 shows the mean total number of LoSs. The two modules were able to reduce the number of LoSs for all traffic scenarios and densities when compared to the baseline navigation rules. The total number of LoSs is not a direct result of the total number of conflicts (see Figure 4.12). However, reducing the number of conflicts has a positive influence on the number of LoSs.

Similarly to the total number of conflicts in Figure 4.13(c), for some traffic scenarios, adding the control-execution module results in an increase in the total number of LoSs compared to employing the decision-making module alone and having immediate merging manoeuvres. The 'delays' caused by the control-execution module increase the total number of conflicts/LoSs, especially for traffic scenarios with a single origin (i.e., all aircraft start their route in the same direction). Per Figures 4.12(c) and 4.13(c), this effect worsens as the traffic density increases. The fact that aircraft all have the same origin means that delaying the vertical manoeuvres also delays the dispersion of this localised high traffic density per the rest of the available airspace. Reducing this traffic concentration as fast as possible has a greater effect globally in reducing the total number of conflicts and LoSs. Although the control-execution module reduces the conflicts/LoSs resulting from merging manoeuvres, in these specific traffic scenarios, these are negligible compared with the instability effect resulting from having such a high number of aircraft travelling in the same layers for a long period of time.

However, there is one traffic scenario where adding the control-execution reduced the total number of conflicts/LoSs when compared to using the decision-making module alone, and it does it at medium and high traffic densities. Here, all aircraft start their flight heading north and will perform four turns to the left during their flight. This traffic scenario has the highest number of conflicts/LoSs of all 75 scenarios, meaning that this
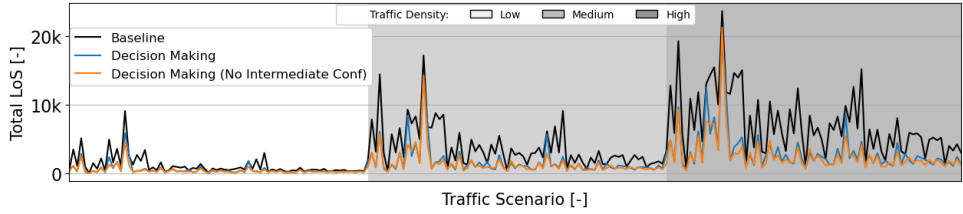
(a) Mean total number of losses of separation during testing of the decision-making module.

(b) Mean total number of losses of separation during testing of the control-execution module.
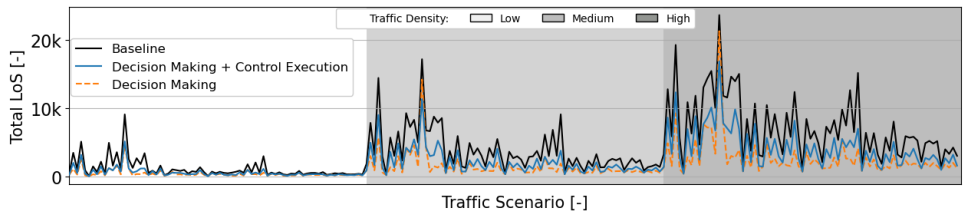
(c) Mean total number of losses of separation during testing of the two RL modules together.

Figure 4.13: Mean total number of losses of separation (LoSs) during the testing of the RL modules. All traffic densities have 75 traffic scenarios, with initial direction(s) and number of turns as defined in Section 4.5.1.

particular direction with a high amount of turns tends to be unsafe in this operational environment. This shows that a control-execution module, as hypothesised, is essential when the merging conflicts/LoSs are the main source of risk. This is the case for flight routes with multiple turns at higher traffic densities, where likely distance gaps between aircraft are smaller. The module reduces merging conflicts while, unfortunately, delaying the dispersion of aircraft clusters in the process, increasing cruising conflicts. Its value is thus highly connected to the traffic scenario.

Figure 4.14 displays the amount of time each aircraft spends in conflict with other aircraft. While in conflict, aircraft will follow the new state computed by the CR method. Aircraft return to their pre-defined route state once detected that they are no longer in a conflict situation with intruders. The final solution, using both RL modules, was able to reduce the time in conflict for all traffic scenarios and densities when compared to the baseline rules. Note that the total number of conflicts (see Figure 4.12) and the total time in conflict do not have a direct correlation. Fewer pairwise conflicts do not necessarily mean less time in conflict per aircraft and vice versa.
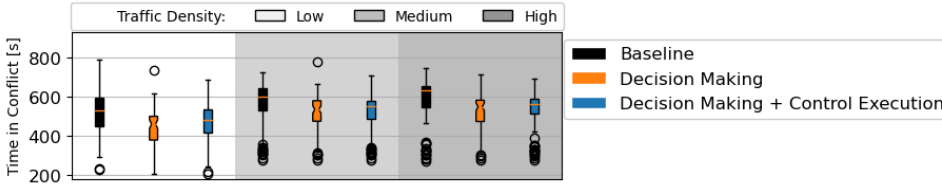
Figure 4.14: Total time in conflict per aircraft. All traffic densities have 75 traffic scenarios, with initial direction(s) and the number of turns following the order defined in Section 4.5.1.

Figure 4.15 displays the intrusion severity. For most traffic scenarios, there is no relevant discrepancy between the efficiency of the combination of the two RL modules and the baseline rules. However, there is a difference between these and the average intrusion severity when employing the decision-making module solely. This is expected: the decision-making module does not take intermediate conflicts/LoSs into account. Thus, it does not try to reduce proximity with other aircraft in intermediate layers, leading to severe intrusions. As a result, adding the control-execution module could be a trade-off between the total number of LoSs (see Figure 4.13(c)) and their severity.

Figure 4.15: Mean intrusion severity rate. All traffic densities have 75 traffic scenarios, with initial direction(s) and number of turns following the order defined in Section 4.5.1.

Figure 4.16 depicts the relative speed between two aircraft in an LoS situation. Higher relative speeds indicate speed heterogeneity, which increases complexity in the airspace. With the baseline rules, transition layers are in place to minimise the effect of high relative speeds from aircraft exiting and entering a cruising layer; aircraft only decelerate/turn/accelerate within the third layer. These layers are safer for this state change, as they are expected to be (almost) devoid of aircraft. Although the RL solution does not leave a layer 'free', such does not result in a considerable increase in relative speed between aircraft. In some traffic scenarios, it even achieved a slight improvement. Optimal segmentation is also beneficial in reducing relative speeds. With fewer conflicts and less time in conflict (see Figures 4.12(c) and 4.14, respectively), aircraft spend a higher amount of time at the ideal cruising speed. Frequent speed variations for conflict resolution may increase speed heterogeneity. Finally, employing the decision-making RL module alone shows some peaks of low relative speeds. These also correspond to traffic scenarios where all aircraft are initially flying in the same direction. Once again, these differences in performance result from a delay in the dispersion of these clusters of aircraft.

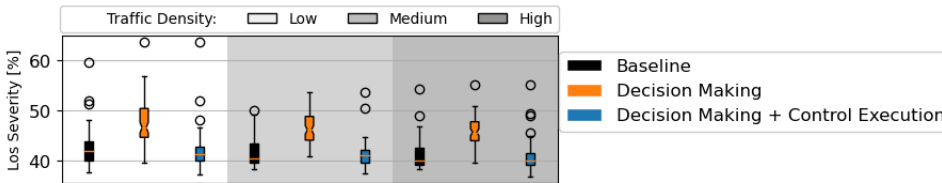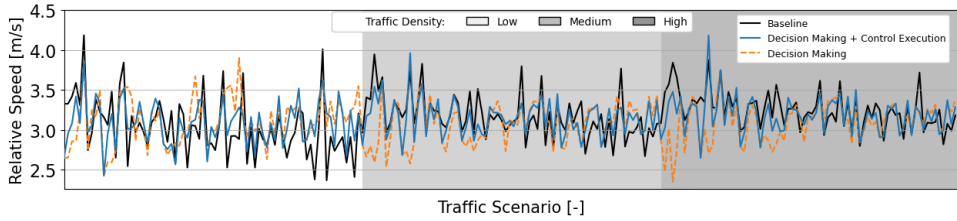Figure 4.16: Mean relative speed between pairs of aircraft in loss of separation. All traffic densities have 75 traffic scenarios, with initial direction(s) and number of turns following the order defined in Section 4.5.1.

STABILITY ANALYSIS

Figure 4.17 shows the mean DEP value. The RL solution shows considerably better stability than the baseline navigation rules. This is likely due to better segmentation of aircraft; the greater distance between aircraft reduces the chance of secondary conflicts when aircraft alter their state.

Figure 4.17: Domino effect parameter values. All traffic densities have 75 traffic scenarios, with initial direction(s) and number of turns following the order defined in Section 4.5.1.

EFFICIENCY ANALYSIS

Figure 4.18 shows the average length of the 3D flight path per aircraft. The differences in length of the flown trajectory originate mainly from: (1) the different vertical distances between traffic layers that the aircraft occupy throughout their path, and (2) the different number of vertical manoeuvres to avoid conflicts. Employing both RL modules shows a slight reduction in flight path length for some of the traffic scenarios when compared to the baseline navigation rules. A bigger reduction in flight path can potentially be achieved when efficiency is also added to the reward formulation. However, this may have the same effect as considering intermediate conflicts in the decision-making module: an optimal cruising layer may be disregarded in favour of a smaller vertical deviation.

Figure 4.19 displays the average flight time per aircraft. For most traffic scenarios, employing the RL solution achieved a faster flight than with the baseline navigation rules. The difference in flight time increases along with traffic density. This results not only from shorter flight paths (see Figure 4.18) but also from aircraft spending less time in conflict (see Figure 4.14). Often, conflict resolution manoeuvres lead to aircraft adopting lower deconflicting speeds, which increase flight time.

Figure 4.18: Flight path length per aircraft. All traffic densities have 75 traffic scenarios, with initial direction(s) and number of turns following the order defined in Section 4.5.1.
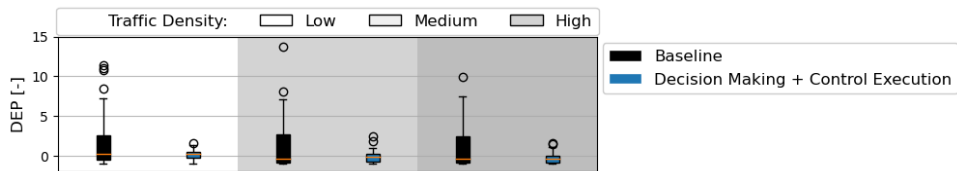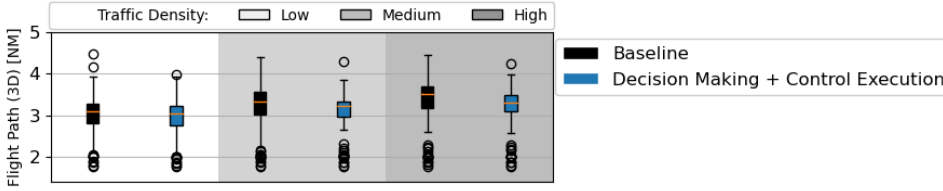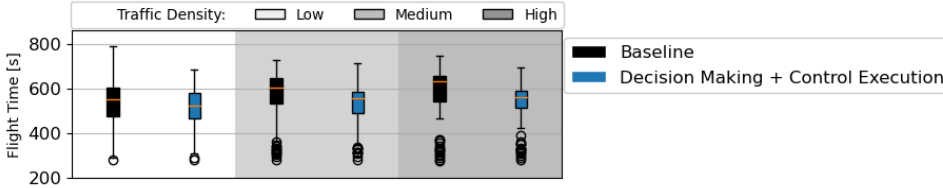

Figure 4.19: Flight time per aircraft. All traffic densities have 75 traffic scenarios, with initial direction(s) and number of turns following the order defined in Section 4.5.1.

4.8. DISCUSSION

Using reinforcement learning to improve the layer change decision proved more successful, safety- and efficiency-wise, than using manually defined baseline rules, both for different traffic scenarios and different traffic densities. First, a decision-making module was used to output layer change commands based on the planned route of the ownship and the position of neighbouring aircraft. The decision-making module proved better at reducing the global number of conflicts and LoSs when conflicts/LoSs resulting from crossing intermediate layers between the initial and the target layer are not taken into consideration. In this way, the module can focus on optimising the cruising phase, which is beneficial for the global traffic scenario, as aircraft try to maximise the cruising phase on their planned route. Second, a control-execution module improves safety during merging manoeuvres by controlling the longitudinal and vertical movements of the merging aircraft. It reduces the negative local impact of the layer change decisions when the ownship crosses multiple layers. By delaying the merging manoeuvre until there is an adequate distance gap between the ownship and the leader and follower aircraft, the intrusion severity during vertical manoeuvres is reduced.

The optimal actions found by an RL method can be used to improve the rules of current navigation analytical methods. Looking at the choices made by the two modules previously described, the following guidelines can be defined:

1. At high traffic densities, a high degree of segmentation during the cruising phase is an effective strategy to decrease conflict and losses of minimum separation.
2. With multiple layers, the separation of aircraft per layer should be performed in relation to how close aircraft are to the next turn. Aircraft closer to a turn should be placed in the outward layers, as they will move out sooner.
3. Delaying a merging manoeuvre may result in a trade-off between reducing the

number of LoSs or keeping LoS severity to a minimum. At high traffic densities, when the gaps between neighbouring aircraft are minimal, most aircraft will likely delay their vertical movement. However, these decisions together result in the local traffic density staying the same, with no improvement in the local situation. Whereas in a method without delays, although the first dispersed aircraft are expected to encounter several conflicts/LoSs, their movement reduces local traffic density. Consequently, the distance gaps between the aircraft increase, facilitating the next vertical manoeuvres.

The unexpected finding that adding the control-executing module to the decision-making module increased the total number of conflicts/LoSs raises questions regarding the observability and reward formulation of the trained RL modules. The control-execution module still has a positive effect locally by reducing the intrusion severity during vertical manoeuvres. However, it also has a negative global effect, delaying the dispersion of aircraft per the available airspace. These elements are further discussed in the following sections.

4.8.1. OBSERVABILITY OF THE REINFORCEMENT LEARNING MODULES

Both RL modules presented in this work have partial observability: the agent has information only on its surroundings, making the observations correlated with its geographical position. However, the results obtained show that the information available is not sufficient for the agent to fully understand the repercussions of its actions. First, given that most of the flight routes favour spending most of their flight cruising, for global safety, it is more beneficial to optimise the cruising phase than to decrease the number of conflicts/LoSs resulting from crossing multiple vertical layers. The latter may result in the module preferring to move to a nearer vertical layer instead of one further away that is potentially safer to cruise in. However, this is not clear to the module as its observability/reward is restricted to the layer change action. Second, delaying the merging manoeuvre until there is a safe distance gap in the target layer prevents the high severity LoSs that the ownship would otherwise suffer, but it also delays the reduction in the local traffic density. Depending on the number of aircraft involved, such may cause instability.

A possible solution would be to increase the amount of information to which each agent has access. For example, the decision-making module can be extended to have more information on the ownship's flight route, hopefully resulting in a more informed decision between prioritisation of cruising and/or turning phases. The reward must also then reflect the safety during the following cruising phase so that the total impact of the layer change manoeuvre plus the cruising phase on the selected layer can be evaluated. However, safety in the cruising phase is also dictated by the other aircraft that join the layer after the ownship. Therefore, it is not clear whether it is possible for the module to correctly evaluate the impact of cruising in a layer. In turn, the control-execution module can be improved to take into account the instability of the surroundings in the form of the local traffic density and relative distance between all aircraft. Increasing the information that each agent has access to requires the exploration of larger state sizes, which also heavily increases the training time and complexity of the state-actions formulations. These are balanced considerations that should be present in future research.

4.8.2. REWARD FORMULATION

The efficacy of a reinforcement learning method is highly dependent on the reward values. The reward formulation used here was based on the number of conflicts and LoSs, as these are considered the main elements of safety. However, the reduction in the absolute value of these elements may have a negative impact on LoS severity (Figure 4.15). This can be a simple action as, for example, the ownship moving away from one intruder and becoming closer to a second intruder in the process. This may result in two not so severe LoSs, versus one severe LoS. This raises the question of whether to prioritise: (1) a low number of LoSs, or (2) a low LoS severity even at the cost of a higher number of LoSs.

Future implementations may benefit from including intrusion severity in the training of RL methods. However, a trade-off must be established between these two aspects. For example, in this work, one LoS was valued as 10 conflicts. The same would need to be established with LoS severity: (1) what are low and high severity intrusions?; (2) how many low severity intrusions count for one high severity intrusion? These seem arbitrary decisions, but they heavily influence the decisions made by the RL modules, and are also dependent on the traffic scenarios and simulation environment.

4.9. CONCLUSIONS

This chapter focused on mitigating the impact of vertical deviations in a layered airspace. Previous hand-crafted rules have limited impact. Notwithstanding the arduous work of experts on the development of these rules, these do not cover the great multitude of different relative geometries between merging, follower, and leader aircraft during a merging manoeuvre. This work took inspiration from extensive research with road vehicles in the area of lane change decisions, where reinforcement learning (RL) techniques have surpassed the performance of hand-crafted rules. We translated these methods into an aviation environment, where they are used for vertical layer change decisions.

This work compared the behaviour of a reinforcement learning-based solution for layer-changing decisions versus employing manually defined navigation rules. Two RL modules were used: a decision-making module, which outputs layer change commands, and a control-execution module, which controls the aircraft longitudinally and vertically to ensure a safe merging manoeuvre. Both modules, working independently and together, reduced the total number of conflicts/LoSs when compared to manually defined baseline rules. The benefit of this approach was especially noticeable at high traffic densities and with routes with a high number of turns. However, it was also shown that delaying a merging manoeuvre, while the gap between the aircraft is yet not sufficient for a safe manoeuvre, also delays the dispersion of aircraft clusters in the process, negatively affecting global safety. Future work should look into the local and global effects, as an action that protects the ownship may increase the risk of conflicts for other neighbouring aircraft.

There is still a long way to go before these RL methods can be implemented in a real-world scenario. However, the behaviour of the methods can already provide guidelines for the implementation of navigation rules. Optimal segmentation during the cruising phase, setting aircraft closer to turns in outward layers, and delaying merging actions so as to limit intrusion severity, can be used to improve the current analytical layer navigation rules. For future improvement of the performance of the RL methods, the reward formulation can be extended to include other safety factors, such as intrusion severity. Finally, this

work can also be extended to more heterogeneous operational environments in terms of differences in performance limits, as well as preference for efficiency over safety.

4

5

USING REINFORCEMENT LEARNING TO IMPROVE AIRSPACE STRUCTURING IN AN URBAN ENVIRONMENT

The results of the Chapters 3 and 4 show that separation of traffic into different altitude layers by employing heading-altitude rules greatly increases safety. Chapter 3 concluded that reinforcing learning (RL) could also be used to improve the structure of the operational environment, catering to the expected traffic scenario. This chapter explores this hypothesis.

In Section 5.5, RL techniques are used to determine the heading range per layer in accordance with the current and expected traffic scenario. Multiple traffic demand scenarios are simulated. Subsequently, the structure-traffic scenario relationship is inferred from the effect of traffic demand variations on a number of airspace performance metrics.

Cover-to-cover readers may chose to skip Sections 5.3, 5.4.2, 5.5.7 which describe a layered airspace, the theoretical background of a Deep Deterministic Policy Gradient (DDPG) algorithm, and the Modified Voltage Potential method, respectively. These are very similar to their counterparts in previous Chapters 3 and 4.

This chapter is based on the following publication:

1. M. Ribeiro, J. Ellerbroek, and J.Hoekstra, Using Reinforcement Learning to Improve Airspace Structuring in an Urban Environment, Aerospace 9 (2022)

ABSTRACT

Current predictions on future drone operations estimate that the traffic density will be orders of magnitude higher than any observed in manned aviation. Such densities redirect the focus towards elements that can decrease conflict rate and severity, with special emphasis on airspace structures, an element that has been overlooked within distributed environments in the past. This work delves into the impact of different airspace structures in multiple traffic scenarios, and how appropriate structures can increase the safety of future drone operations in urban airspace. First, reinforcement learning is used to define optimal heading range distributions with a layered airspace concept. Second, transition layers are reserved to facilitate the vertical deviation between cruising layers and conflict resolution. The effects of traffic density, non-linear routes, and vertical deviation between layers are tested in an open-source airspace simulation platform. Results show that optimal structuring catered to the current traffic scenario improves airspace usage by correctly segmenting aircraft according to their flight routes. The number of conflicts and losses of minimum separation was reduced versus using a single, uniform airspace structure for all traffic scenarios, thus enabling higher airspace capacity.

5

5.1. INTRODUCTION

The European drones outlook study [197] estimates that as many as 400.000 drones will be operating in the airspace by 2050. The use of machine learning in tactical conflict detection and resolution (CD&R) could potentially support advanced and scalable access to the airspace for a large number of drone (U-space) services. The present work aids this research by developing a reinforcement learning module that selects the optimal airspace structure for the current traffic. The objective is to decrease the conflict severity and rate for unmanned aviation operations in urban environments. A conflict is a predicted future loss of minimum separation (LoS). A loss of minimum separation (or intrusion) occurs when two aircraft are closer to each other than the minimum separation distance. The paramount objective of Air Traffic Control (ATC) is to prevent intrusions.

Airspace structure plays a positive role in airspace capacity. Within centralised ATC, structuring consists of separating the airspace into different sectors. Each air traffic controller (ATCo) is responsible for one sector. The number of aircraft in each sector is limited to how many aircraft each ATCo can control simultaneously [207]. However, it is yet not clear how to optimally structure a distributed airspace. The Metropolis Project explored different types of airspace structures for manned flights in a dense urban area, using distributed separation assurance [142]. Results showed that a 'layers' concept, where the available airspace is segmented vertically, increases airspace capacity by reducing the number of conflicts and losses of minimum separation. This concept was further developed recently for unmanned aviation [172], where all directions within an urban infrastructure were divided per the available vertical layers. This research focused on a single, uniform structure and analyzed its effect. The present work builds upon the latter by exploring optimized structures catered to the expected traffic scenario.

Research related to road vehicles explored reinforcement learning (RL) to improve lane configuration [208, 209]. Dynamic lane configurations reduced the average travel

time in congested road networks when compared to a fixed, traditional lane-direction configuration [210]. Fixed configurations assume pre-known, static traffic patterns. However, in the real world, traffic may change considerably; one single configuration is not necessarily optimal for all traffic situations [211]. Urban air traffic has several similarities with road traffic that justify exploring machine learning techniques successfully applied in the latter [212, 213]. First, unmanned aviation is set to follow road infrastructure [214]. Thus, the effects of the environment topology on traffic agglomeration are similar in both cases. Collisions are prevented by maintaining a minimum distance between vehicles, comparable to aviation. However, there are remarkable differences between drones and road vehicles. The latter can become stationary, but not all drones can hover [215]. Additionally, in aviation, minimum separation distances are typically larger. These challenges will be further examined in this work.

This study uses the open-source, ATC simulation tool BlueSky [25] to simulate operations in an urban environment. Aircraft follow pre-planned routes around urban infrastructure (thus, preventing collisions with static obstacles). Conflicts between aircraft are resolved with conflict resolution (CR) with implicit coordination. This work resortes to CR method Modified Voltage Potential (MVP) [15], which has proved effective in reducing losses of separation with minimal state deviation [162]. Normally, most conflict detection and resolution (CD&R) methods favor heading deviations as preferred by air traffic controllers. However, in an urban environment, such deviations could result in collisions with the surrounding infrastructure. We favor a speed and altitude-based conflict resolution approach, guaranteeing that the frontiers with the surrounding urban infrastructure are always respected. Finally, the deep deterministic policy gradient (DDPG) [163] method was used to determine optimal directions per layer within a layered airspace concept.

5.2. RELATED WORK

ATM is a critical domain, with safety as the top priority, which explains the slow progress in the use of machine learning (ML) approaches in the ATM domain when compared to other research fields [216]. Here, we focus on the application of ML for airspace design. The body of work in this area is narrow; ML approaches are often limited to assessing the complexity in an airspace sector. Brito [217] used supervised learning to predict air traffic demand in airspace sectors, enhancing the predictability of airspace sector demand versus a baseline demand estimation model, which mimics the current practice. Li [218] employed an unsupervised learning approach for the airspace complexity evaluation; results showed that it outperformed state-of-the-art methods in terms of airspace complexity evaluation accuracy. Finally, Wieland [219] showed that ML approaches can help determine the importance of each complexity feature in predicting airspace capacity.

Regarding airspace structuring, existent ML methods are more directed at manned aviation, focusing on airspace sectors. Xue [220] approached dynamic vector resectorization with Voronoi diagrams and genetic algorithms. Results show that these are capable of determining the dominant traffic flow, which is one of the main concerns in sector design. Kulkarni [221] used dynamic programming to partition airspace based on the ATCos workload, showing that this could be a viable tool. Finally, Tang [222] proposed an agent-based method to dynamically partition the airspace, to accommodate the traffic growth while satisfying efficiency metrics. The trained method showed promising results

both in balancing the ATCos workload and the average sector flight time.

To the best of the authors' knowledge, this is the first work that approaches airspace structuring for unmanned aviation environments. The latter entails a very specific challenge: these types of operations entail a much higher number of heading deviations (i.e., turns during the flight route) than manned aviation, where aircraft employ (as much as possible) direct routes from the start to the endpoint. We employ an urban environment with the objective of 'forcing' turns to analyse whether the RL method can adapt to these changes. The RL method is responsible for defining the 'directions' allowed at each layer, following the topology of the urban environment. Note that the RL method herein employed could also be applied to an unconstrained, layered airspace. In this case, the method should be used to define the heading ranges allowed in each vertical layer.

5.3. LAYERED URBAN AIRSPACE DESIGN

The usage of drones in an urban environment entails several challenges. Separation with the urban infrastructure must be guaranteed at all times. Most of the current tactical CD&R methods are directed at manned aviation, aimed at detecting other flying traffic at cruise altitude. A method directed at dynamic obstacles cannot automatically be translated to defend against static obstacles. In most existing research on tactical conflict resolution, static obstacles are predominantly defined as (sparse) objects to fly around, as opposed to a multitude of objects that dominate the available space to operate [168]. This work considers that aircraft follow a pre-defined safe route around all static obstacles. Waypoints are set at the center of the roads, from which aircraft do not deviate.

Conflict resolution is not as efficient as it would be in non-constrained airspace, as aircraft cannot modify their headings to avoid conflicts. Near head-on conflicts are practically impossible to resolve without heading deviation. The focus must then be on conflict prevention. Airspace structures directly reduce conflict probability by decreasing the likelihood of aircraft meeting during their flights. The Metropolis Project has shown that a layered airspace structure considerably reduces the rate of conflicts [175]. Two effects contribute to this reduction. First, the total traffic density is segmented into groups of aircraft allocated at different altitude layers. Second, these groups are divided per aircraft heading, enforcing a degree of alignment between the aircraft, which decreases the likelihood of conflicts in each layer.

Previous research [13, 172–174] investigated the layered concept in urban environments. However, only evenly distributed heading ranges per layer (as exemplified in Figure 5.1) have been researched. However, this is only optimal when the heading distribution of the traffic is uniformly distributed as well. In reality, this is often not the case. Flights may be performed predominantly in specific directions, following the topology of the bigger avenues in the urban environment. Aircraft may be expected to heavily move towards areas with higher population densities, or to a few specific storage points when employed for delivery purposes. Additionally, the directions of flight may change often as aircraft redirect at intersections to avoid collisions with static obstacles.

Aircraft will not be equally distributed over the available airspace when the structure of the airspace does not align with the current heading distribution. One layer will have a higher traffic density than the others when aircraft predominantly adopt a certain direction. In the worst-case scenario, the segmentation factor will be lost, canceling

out the benefit of having a layered structure. Thus, the airspace structure should be set as a function of the current traffic scenario to prevent conflicts and reduce travel time. Moreover, given the fast-changing nature of the traffic, an automated control is preferable to guarantee fast response times and higher structure variability. In this work, we propose a reinforcement learning approach responsible for defining the heading range per traffic layer as a function of the expected traffic scenario. The objective is for this automated agent to focus on dividing aircraft per layer according to the real distribution, making full use of the available airspace.
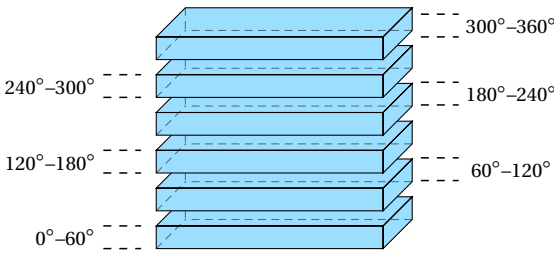

Figure 5.1: Evenly distributed airspace structure; the total heading range (360°) is divided per the available traffic layers.

5.4. AIRSPACE STRUCTURE WITH REINFORCEMENT LEARNING

5.4.1. AGENT
We employ an RL agent whose objective is to optimise the airspace structure in function of the traffic scenario. We assume that the agent has full information on the future traffic density and trajectories. In a real-world application, this agent might be seen as a central component, responsible for defining the structure of the operational airspace.

5.4.2. LEARNING ALGORITHM
An RL method consists of an agent interacting with an environment E in discrete timesteps. At each timestep, the agent receives the current state s of the environment and performs an action a in accordance with which it receives a reward r_t. An agent's behavior is defined by a policy, π, which maps states to actions. The goal is to learn a policy that maximizes the reward. Many RL algorithms have been researched in terms of defining the expected reward following action a. In this work, we used the deep deterministic policy gradient (DDPG), defined in [163].

Policy gradient algorithms first evaluate the policy and then follow the policy gradient to maximise the performance. DDPG is a deterministic actor–critic policy gradient algorithm, designed to handle continuous and high-dimensional state and action spaces. It has proven to outperform other RL algorithms in environments with stable dynamics [164]. Additionally, DDPG has been successfully implemented in the aviation environment [223–225], proving that it can adapt to aircraft dynamics. However, DDPG can become unstable, being particularly sensitive to reward scale settings [188, 189]. As a result, rewards must be carefully defined.

  DDPG is an instance of the actor–critic model. The deterministic actor receives a state from the environment and outputs an action. The critic maps each state–action pair, informing the actor how to adjust towards outputting the best actions. Furthermore, the DDPG method employs target networks and a replay buffer. The target networks are mostly useful to stabilise function approximation when learning for the critic and actor networks. The replay buffer stores multiple past experiences, from which mini-batch samples are used to update the actor and critic. The pseudo-code for DDPG is displayed in Algorithm 5.1. Additionally, noise is added to promote exploration of the environment; an Ornstein–Uhlenbeck process [191] is used in parallel with the authors of the DDPG method. Table 5.1 presents the hyperparameters employed in this work. We resort to 2 hidden layer-neural networks with 120 neurons in each layer.

Algorithm 5.1 Deep deterministic policy gradient.

Initialize critic $Q(s|a^\mu)$ and actor $\mu(s|\theta^\mu)$ networks, replay buffer R, and action exploration
for all episodes **do**
  while episode not ended **do**
    Select action a_t according to the current state s_t from the environment and the current *actor* network
    Perform action a_t in the environment and receive a reward r_t and new state s_{t+1}
    Store transition (s_t, a_t, r_t, s_{t+1}) in the replay buffer R
    Sample a random mini-batch of N transitions from R
    Update *critic* by minimizing the loss
    Update the *actor* policy using the sample policy gradient
    Update the *target* networks
  end while
  Reset the environment
end for

Table 5.1: Hyperparameters of the employed RL method used in this work.

Parameter	Value
TAU	0.001
Learning rate actor (LRA)	0.0001
Learning rate critic (LRC)	0.001
EPSILON	0.1
GAMMA	0.99
Buffer size	1 M
Minibatch size	256
# hidden layer-neural networks	2
# Neurons	120 in each layer
Activation functions	Rectified linear unit (ReLU) in the hidden layers, Softmax in the last layer

5.4.3. State

The state input into the RL method must contain the necessary data for the RL agent to successfully determine an optimal heading division per traffic layer. We consider that such a decision requires information on the traffic demand, flight routes, and their evolution over time. However, representing correct traffic flow evolution is non-trivial

and can assume various shapes. Moreover, with RL, a simplified representation of the environment is often needed to optimise the training of the neural network. Representing the complete flight routes for all aircraft would greatly increase the size of the state formulation and with it the number of possible states and state–action combinations. As the size of the problem's solution space grows exponentially with the number of states, it may reach a point where the training time becomes unrealistic.

To enable a fixed array size, representing a simplified version of the environment, a maximum number of four possible directions are considered within the simulated urban environment: East, South, West, and North. Then, 'snapshots' are taken of the predicted future traffic scenario at different points in time. Each point in time is defined by four variables, with each variable representing the number of aircraft in each of the four possible directions. Figure 5.2 represents the complete state array. A total of four 'snapshops' are taken, each one further in time by five minutes. For example, E_1 represents the number of aircraft traveling East, 5 minutes past the start of the traffic scenario. Naturally, having more 'snapshots' provides more information regarding the environment but at the cost of adding more complexity to the method.

Additionally, for simplification, a fixed number of vertical layers is assumed. Six traffic layers are defined. The six final elements of the state array (L_1 to L_6), are used to indicate the current number of aircraft in each traffic layer. It is considered that the structure is set before the aircraft initiated their flights. Thus, the airspace is empty at the beginning of each episode, and the six final positions equal 0 in the initial state. However, at the end of the episode, as the RL method is informed about the next state, this information becomes relevant. Ideally, the RL agent should opt for a structure that homogeneously divides aircraft across the available airspace (segmentation effect). Additionally, this state formulation could potentially be used in a situation where the traffic volume at the beginning of the episode is not zero, as it is capable of transmitting such information.

E_0	S_0	W_0	N_0	E_1	S_1	W_1	N_1	E_2	S_2	W_2	N_2	E_3	S_3	W_3	N_3	L_1	L_2	L_3	L_4	L_5	L_6

Traffic in 0 min Traffic in 5 min Traffic in 10 min Traffic in 15 min Layer Occupancy
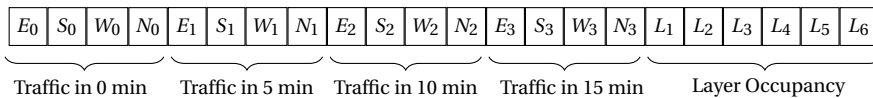
Figure 5.2: State formulation of the reinforcement learning agent. The first 16 positions represent the expected traffic intensity per direction (East, South, West, and North) at the expected point in time. The last 6 position represents the current number of aircraft in each traffic layer.

5.4.4. Action

The RL agent determines the action to be performed for the current state. The incoming state values are transformed through each layer of the neural network, in accordance with the neuron weights and the activation function in each layer. The activation function takes in the output values from the previous layer and converts them into a form that can be taken as the input for the next layer. The output of the final layer must be turned into values that can be used to define the 'direction' in each traffic layer. A softmax activation function is employed in the last layer; the output values are used to define which direction was allowed at each traffic layer. The dimension of the action array is set to 24 (4 directions × 6 layers). Figure 5.3 shows how the necessary information is extracted from the action

array. For example, the first 4 positions of the array correspond to the 4 directions possible in the first layer; the direction with the highest integer value was picked.

E_0	S_0	W_0	N_0	E_1	S_1	W_1	N_1	E_2	S_2	W_2	N_2	E_3	S_3	W_3	N_3	E_0	S_0	W_0	N_0	E_1	S_1	W_1	N_1

1st Layer 2nd Layer 3rd Layer 4th Layer 5th Layer 6th Layer
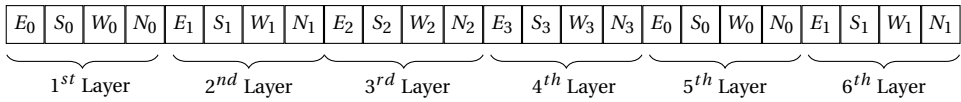
Figure 5.3: Action array output by the reinforcement learning method. Each successive 4 positions represent a traffic layer. The highest integer indicates the direction (East, South, West, or North) to be allowed in the respective layer.

Thus there are two main components upon which the RL agent decides:

- The number of layers for each direction: the RL agent may decide to select more layers for a direction adopted by the majority of aircraft. However, an important safeguard was implemented upon the airspace structure output by the RL agent. To make sure that all directions were allowed in the airspace, a final check was applied to the structure. If all possible directions were not yet allowed, the last layer was overwritten to allow for the missing directions. Note that it may occur that more than one flight direction is allowed in this layer.
- The order of the layers: the RL agent decides which directions are in adjacent layers. For a fixed structure, it is good practice to allow the left or right turning by just climbing or descending one layer. However, on purpose, the agent is free to choose the order of directions. It will be evaluated whether the structure output by the RL agent includes an understanding of perpendicular directions.

5.4.5. REWARD

The RL method should prioritise safety, with the paramount factor being the number of conflicts/LoSs. However, it is unclear, at this state, which element will result in a more optimal convergence: (1) the total number of conflicts, or (2) the total number of losses of minimum separation. As a result, the following reward formulations will be tested and compared:

1. The RL method receives a -1 for each conflict.
2. The RL method receives a -1 for each loss of minimum separation.

A loss of separation is detected when two aircraft are closer to each other than the minimum separation distance. A conflict is a predicted future loss of minimum separation. More details on the state-based conflict detection used in this work are given in Section 5.5.6.

Note that a considerable limitation of this reward formulation is the fact that it does not take into consideration efficiency, more specifically, (1) extra energy consumption resulting from drones traversing between layers far away, and (2) extra energy consumption due to the vertical conflict resolution manoeuvres. Urban air mobility vehicles are limited energy-wise. Thus, these manoeuvres can hinder the paths and travel times of these vehicles. Nevertheless, this work is the first approach intended to study whether RL methods can successfully set an airspace structure adapted to the traffic scenario; thus, we opted for a simple reward formulation focusing only on safety. Notwithstanding, it

may be considered that safety has an indirect positive effect on efficiency: decreasing both the total number of conflicts or LoSs directly reduces the number of vertical conflict resolution manoeuvres. Future work should consider efficiency elements as well. Nevertheless, weights of safety and efficiency should be carefully considered. Safety should not be jeopardised in favor of faster or shorter flights.

5.5. EXPERIMENT: SAFETY-OPTIMISED AIRSPACE STRUCTURE

The following subsections define the properties of the performed experiment. The latter aims at using RL to define the heading range at each vertical traffic layer within layered urban airspace. Note that the experiment involves a training and a testing phase. First, the RL method was trained continuously with a set of traffic scenarios. Second, it was tested with unknown traffic scenarios. Performances with these new scenarios are directly compared to a baseline that employed evenly distributed heading ranges per layer.

5.5.1. SIMULATED ENVIRONMENT

We first define the simulation area. This is an urban setting built using the Open Street Map networks (OSMnx) python library [167], an open-source tool for street network analysis. We used an excerpt from the San Francisco Area, representing an orthogonal street layout with an area of $1.708\,\text{NM}^2$, as depicted in Figure 5.4. The OSMnx library returned a set of nodes from which a network of roads could be defined.

In this area, roads and intersections were defined by vertices and nodes, respectively. Two adjacent nodes represent the edges of a road. Aircraft can only travel from one node to another when these are connected. With the intention of reducing complexity, each node was considered to have (upmost) four connecting roads, as shown in Figure 5.5. Only existing roads were considered. Additionally, we assume that each road was unidirectional, with only one lane. We did not make any assumption regarding the width of the road, which would have been needed if more directions were considered.
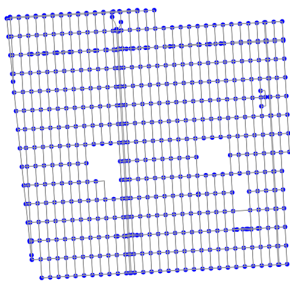
Figure 5.4: The urban environment used in these experiments. The data was retrieved from the OSMnx python library [167].
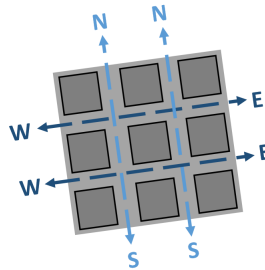

Figure 5.5: Possible directions in each one of six available traffic layers: W (west), N (north), E (east), and S (south).

5.5.2. TRANSITION LAYERS

In conventional aviation, temporary altitude layers are often used as a level-off at an intermediate flight level along a climb or descent to avoid conflicts [205]. In our urban

airspace, we applied the same concept: we include (low-speed) transition layers in the airspace to be used only by aircraft that were transitioning between traffic layers. Aircraft perform the heading turns in these transition layers, preventing conflicts resulting from heterogeneous speed situations when an aircraft decelerated just before a turn. Transition layers are expected to be (almost) depleted of aircraft at any point in time, reducing the likelihood of aircraft meeting in conflict. Moreover, we consider that aircraft fly along the middle of the road. Since we also make no assumptions about the width of a street, aircraft were also not allowed to use heading changes for conflict resolution. This means that aircraft can only resort to speed and altitude changes to avoid conflicts. However, a vertical space needs to be reserved for vertical conflict resolution, preventing aircraft from entering adjacent traffic layers. Thus, additional vertical layers are allocated for this purpose. Figure 5.6 depicts the different layers used in the experimental scenario.
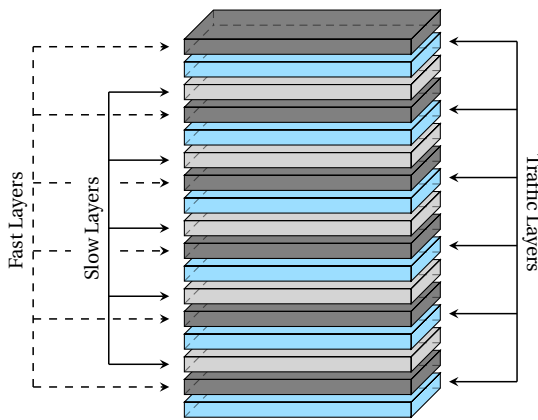

Figure 5.6: Different altitude layers used in this work.

Three different layer types were considered, each dedicated to different actions:

- Six traffic layers (blue): the main layers used by cruising traffic.
- Six slow transition layers (light grey): are used for transitioning between traffic layers. This is a necessary mid-step prior to the aircraft entering different traffic layers. First, the aircraft exit the current traffic layer without modifying their speed, to not create conflict with other cruising aircraft, and they move towards the slow layer. Here, the aircraft decrease their speed to reach the speed required to comply with the turn radius. After turning, the aircraft start accelerating towards the desired cruising speed/moving to the destination traffic layer.
- Six fast transition layers (dark grey): are used to perform vertical conflict resolution when necessary. The overtaking aircraft resolve the conflict by moving into the fast layer; aircraft being overtaken have the right of way. Once the conflict is resolved, the aircraft move back into the traffic layer to guarantee that the fast layers are (mostly) depleted of other traffic when the aircraft need to perform vertical resolution.

All layers were set with a height of 15 ft. There is a margin of 5 ft between the layers to prevent false conflicts.

5.5.3. FLIGHT ROUTES

Aircraft spawn locations (origins) are placed in alternating orders on the edge of the simulation area, with a minimum spacing equal to the minimum separation distance, to avoid conflicts between spawn aircraft and aircraft arriving at their destinations. Multiple traffic layers are used; aircraft are spawned at a layer that allowed for the initial heading. Aircraft climb almost vertically. Finally, an aircraft is deleted from the simulation once it leaves the simulation area. To prevent aircraft from being removed incorrectly when traveling through an edge road, aircraft are set to move out of the map once they finished their route and are removed once they moved away from an edge node.

Each aircraft has several waypoints it must pass through. These are always nodes from the map and are calculated based on the defined initial direction, number, and direction of turns, as displayed in Table 5.2. There are a total of 75 traffic scenarios (15 initial heading distribution × 5 turns) per traffic density. During the creation of the simulation scenarios, the total flight time of the already created aircraft is accounted for so that the desired instantaneous traffic densities are respected. All aircraft start at the corresponding end of the map, allowing for a linear route towards their initial directions (e.g., an aircraft with an initial direction of the East starts at the West end of the map). If there are no turns, the aircraft will travel in their initial directions throughout the complete route. A turn to the right from an aircraft with the initial direction East indicates that the aircraft will turn South during its route. A turn to the left would result in this aircraft turning North.

Table 5.2: Flight routes are defined as per the initial direction and the number of turns. The aircraft initial distribution defines, for each scenario, the percentage of flights starting in each initial direction. A total of 15 scenarios with different initial distributions were used. Each scenario was performed five times, with a different number and direction of turns.

	Traffic Scenario:	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12	#13	#14	#15
% A/C Initial Hdg Distribution	East (E):	100	0	0	0	50	50	50	0	0	0	33	33	33	0	25
	South (S):	0	100	0	0	50	0	0	50	50	0	33	33	0	33	25
	West (W):	0	0	100	0	0	50	0	50	0	50	33	0	33	33	25
	North (N):	0	0	0	100	0	0	50	0	50	50	0	33	33	33	25
Flight Path With Turns:	All traffic scenarios are repeated with:								•No Turns (0) •2 Turns to the Right (2R) •4 Turns to the Right (4R) •2 Turns to the Left (2L) •4 Turns to the Left (4L)							

During the training of the RL method, one set of 75 traffic scenarios with medium traffic density was used. During testing, three different sets of each traffic density (low, medium, and high traffic density) is run. Thus, testing was conducted for three different trajectories for each combination of initial direction and the number of turns. This variability of traffic scenarios is aimed at testing the performance of the RL method in multiple situations. Using different heading distributions tests the capacity of the RL method to successfully segment different traffic scenarios over the available airspace. Using a different number of turns tests the ability of the method to protect against successive changes in the heading distribution.

5.5.4. APPARATUS AND AIRCRAFT MODEL

The open-air traffic simulator BlueSky [25] is used to test the efficacy of dynamic airspace structuring. The performance characteristics of the DJI Mavic Pro are used to simulate all vehicles. Here, speed and mass were retrieved from the manufacturer's data, and common conservative values were assumed for the turn rate (max: 15°/s), acceleration, and breaking (1.0 kts/s).

5.5.5. MINIMUM SEPARATION

The appropriate minimum safe separation distance depends on the operational environment and type of aircraft involved. For unmanned aviation, there are no established separation distance standards yet. We opt for 50 m for horizontal separation, as commonly used in research [59]. For vertical separation, 15 ft was assumed, based on the dimension of the vertical layers.

5.5.6. CONFLICT DETECTION

This study employs state-based conflict detection, which assumes the linear propagation of the current state of all aircraft involved. Thus, the time to the closest point of approach (CPA), in seconds, is calculated as:

$$t_{CPA} = -\frac{\vec{d}_{rel} \cdot \vec{v}_{rel}}{\vec{v}_{rel}}, \tag{5.1}$$

where $\vec{d}_{rel}$ is the Cartesian distance vector between the involved aircraft (in meters) and $\vec{v}_{rel}$ is the vector difference between the velocity vectors of the involved aircraft (in meters per second). The distance between aircraft at CPA (in meters) is calculated as:

$$d_{CPA} = \sqrt{\vec{d}_{rel}^2 - t_{CPA}^2 \cdot \vec{v}_{rel}^2}. \tag{5.2}$$

When the separation distance is calculated to be smaller than the specified minimal horizontal spacing, a time interval can be calculated in which separation will be lost if no action is taken:

$$t_{in}, t_{out} = t_{CPA} \pm \frac{\sqrt{R_{PZ}^2 - d_{CPA}^2}}{\vec{v}_{rel}}. \tag{5.3}$$

These equations will be used to detect conflicts, which are said to occur when $d_{CPA} < R_{PZ}$, and $t_{in} \leq t_{lookahead}$, where R_{PZ} is the radius of the protected zone or the minimum horizontal separation and $t_{lookahead}$ is the specified look-ahead time. A look-ahead time of 30 seconds was used for conflict detection and resolution.

5.5.7. CONFLICT RESOLUTION

To guarantee safety in between static obstacles (e.g., buildings, trees), movement within the horizontal plane was severely limited. For conflict resolution, we look at the remaining degrees of freedom, namely speed and altitude variations. Within an urban environment, we may consider two main conflict geometries: (1) conflicts with aircraft traveling along the same road; (2) conflicts at intersections. Within the first case, aircraft fly in the

same direction; intruders are positioned directly in front or behind the ownship. These conflicts can be treated as pairwise conflicts, with a simple resolution, where each aircraft respects a minimum distance to the aircraft in front. The second type of conflict is more complicated. Crossing traffic flows, or merging aircraft, leads to multi-aircraft conflicts for which simple rules no longer suffice. For these conflicts, we resort to the velocity obstacle theory [148, 149], which translates the two-dimensional problem of crossing flows into speed constraints, identifying which velocities result in conflicts.

Figure 5.7 exemplifies the construction of a velocity obstacle (VO). Ownship (A) is in conflict with an intruder (B). A collision cone (CC) can be defined as the triangular area between the lines tangential to the intruder's protected zone (PZ). A and B are in conflict when the relative velocity between these two aircraft is inside the CC. A VO is defined as a collision cone translated by the intruder's velocity; thus, expressing the separation constraints to the absolute velocity space of the ownship. This VO represents the set of ownship velocities that lead to a loss of separation with the intruder. R represents the radius of the PZ. $P_A(t_0)$ and $P_B(t_0)$ denote the initial positions of the ownship and the intruder, respectively. $P_B(t_c)$ identifies the intruder's position at the moment of collision. Each intruder in the vicinity of an ownship results in a separate VO.
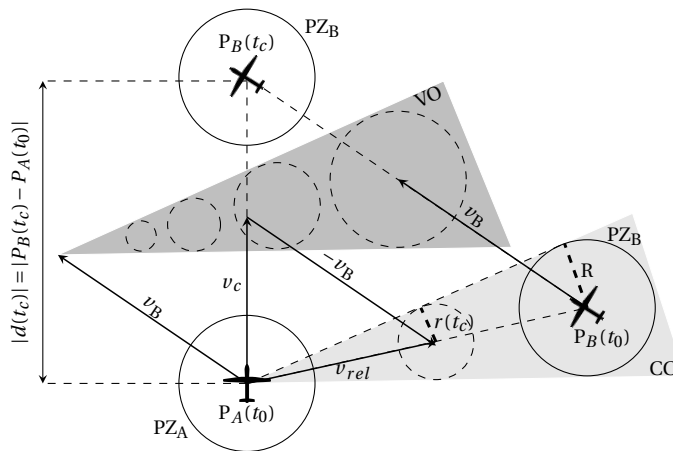

Figure 5.7: Representation of a velocity obstacle (VO) imposed by intruder B, and the relationship between a circular velocity vector set and the protected zone (PZ) [94]. By adding the intruder's velocity, the collision cone (CC) is translated, forming the intruder's VO.

The geometric resolution of the MVP method, as defined by Hoekstra [2, 15], is displayed in Figure 5.8. When a conflict is detected, MVP uses the predicted future positions of both ownship and intruder at the closest point of approach (CPA). These calculated positions 'repel' each other, and this 'repelling force' is converted to a displacement of the predicted position at CPA. The resolution vector is calculated as the vector starting at the future position of the ownship and ending at the edge of the intruder's protected zone, in the direction of the minimum distance vector. Thus, this displacement is the shortest way out of the intruder's protected zone. Dividing the resolution vector by the time left to CPA yields a new speed, which can be added to the ownship's current speed vector, resulting

in a new advised speed vector. From the latter, a new advised heading and speed can be retrieved. The same principle is used in the vertical situation, resulting in an advised vertical speed. In a multi-conflict situation, the final resolution vector is determined by summing the repulsive forces with all intruders. As it is assumed that both aircraft in a conflict will take (opposite) measures to evade the other, MVP is implicitly coordinated.
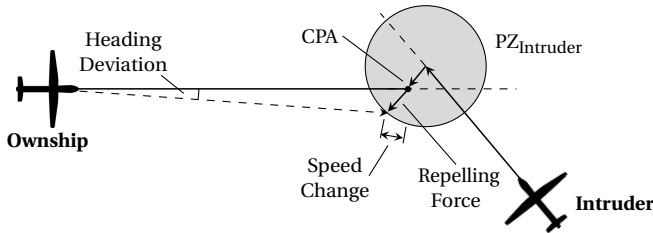

Figure 5.8: Modified voltage potential (MVP) geometric resolution. Adapted from [15].

5.5.8. INDEPENDENT VARIABLES

During training, reward formulation and conflict resolution are introduced as independent variables to observe how each influenced the training of the RL agent. During testing, different traffic densities are introduced to analyse how the RL method performs at traffic densities in which it was not trained. Additionally, airspace structures output by the RL method are compared with a commonly used fixed, uniform airspace structure. More details are given below.

REWARD FORMULATION

Two different reward formulations are tested and compared in terms of training efficacy: (1) -1 per each conflict; (2) -1 per each LoS.

CONFLICT RESOLUTION

The effect of conflict resolution on safety results is tested by directly comparing the efficacy of an RL agent trained in an environment without conflict resolution (CR-OFF), with another RL agent trained in an environment where MVP was used to generate conflict resolution manoeuvres through speed and altitude variation (CR-ON).

TRAFFIC DENSITY

Traffic density ranges from low to high according to Table 5.3. The instantaneous aircraft defines the number of aircraft expected at any given time during the measurement period. At high densities, aircraft spent more than 10% of their flight times avoiding conflicts [193]. The RL agent responsible for setting the airspace structure was trained at a medium traffic density and then tested with low, medium, and high traffic densities. In this way, it is possible to assess the efficacy of an agent performing at a traffic density different from that in which it was trained.

Table 5.3: Traffic volume used in the experimental simulations. The number of spawned aircraft correspond to 20 min of simulation time; the range results from different flight paths as the necessary time to traverse the environment is dependent on the initial direction(s) and the number of turns.

	Low	Medium	High
Traffic density [ac/10,000 NM^2]	292,740	585,408	878,112
Number of instantaneous aircraft [-]	50	100	150
Number of spawned aircraft [-]	80–397	159–794	236–1189

AIRSPACE STRUCTURE

The airspace-structured output by the RL agent must be compared to a baseline-fixed structure ([W,N, E, S, W,N]), to verify that there is a significant improvement in having dynamic structuring catered to each traffic scenario vs one pre-defined structure. The latter is the structure defined in Table 5.4, which obtained good results in previous research [206]. This baseline structure adopts one direction per vertical layer. It is possible to cross into a perpendicular road by climbing or descending to the next layer. The latter is the main benefit of this structure, as it reduces the number of necessary vertical deviations.

Table 5.4: Quadrant rules per altitude layer.

Altitude
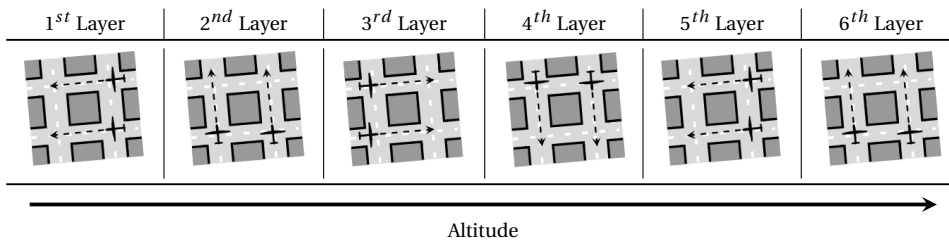
5.5.9. DEPENDENT VARIABLES

Three different categories of measures, safety, stability, and efficiency, are used to evaluate the effects of the different operating rules in the simulation environment.

SAFETY ANALYSIS

Safety is defined in terms of the number and duration of conflicts and losses of separation. Fewer conflicts and losses are considered safer. Additionally, losses of separation are distinguished based on their severity according to how close the aircraft are to each other:

$$LoS_{sev} = \frac{R - d_{CPA}}{R}. \tag{5.4}$$

A low separation severity is preferred.

STABILITY ANALYSIS

Stability refers to the tendency for tactical conflict resolution manoeuvres to create secondary conflicts. In the literature, this effect has been measured using the domino effect parameter (DEP) [151]:

$$DEP = \frac{n_{cfl}^{ON} - n_{cfl}^{OFF}}{n_{cfl}^{OFF}}, \tag{5.5}$$

where n_{cfl}^{ON} and n_{cfl}^{OFF} represent the number of conflicts with CD&R ON and OFF, respectively. A higher DEP value indicates a more destabilising method, creating more conflict chain reactions.

EFFICIENCY ANALYSIS

Efficiency is evaluated in terms of the distance travelled and the duration of the flight. There is a preference for methods that do not considerably increase the path travelled and/or the duration of the flight.

5.6. EXPERIMENT: HYPOTHESES

5.6.1. SIMULATED TRAFFIC SCENARIOS

A set of 75 different scenarios was simulated for each traffic density (low, medium, and high traffic densities). During training, only the medium traffic density is employed; during testing, the three different traffic densities are employed. Within the different scenarios, different initial directions and number of turns throughout the flight routes were set. This is an attempt to introduce a varying number of aircraft per direction and a different number of vertical deviations. Given that the traffic density is constant throughout each scenario, it is hypothesised that a smaller number of different initial traffic directions leads to a higher number of conflicts and LoSs, due to the fact that all aircraft would be travelling in the same vertical layers, on the same 'roads'. When more initial directions are in place, existing traffic is distributed among the airspace to a greater extent, reducing the probability of aircraft meeting in conflict. Additionally, it is hypothesised that a higher number of turns is harder to optimise, as turns are not explicitly represented in the state formulation.

However, looking only at the number of initial directions and turns is not enough to immediately identify the total number of conflicts and LoSs at the end of the simulation. Safety also depends on the trajectories taken and the topology of the environment. The latter may make some directions more prone to conflicts than others; the position of static obstacles may lead to certain locations turning into conflict 'hotspots'. The latter will be analysed with the experimental results.

5.6.2. DYNAMIC AIRSPACE STRUCTURING

It is hypothesised that having a dynamic airspace structure that caters to the expected traffic scenario results in fewer conflicts and LoSs compared to having one fixed structure, which is not optimal for all different traffic cases. For an unbiased comparison, we employ a fixed structure that is expected to perform reasonably well in a wide range of different traffic scenarios. The structure (W, N, E, S, W, N) was chosen; the latter has been proven to be successful in previous research [206]. Naturally, it could even be that there are specific traffic scenarios for which this baseline structure is more efficient and may outperform the structure output by the RL method. This is relevant for comparison to assess which structuring characteristics lead to improved safety.

5.6.3. Training of the Reinforcement Learning Method

The use of conflict resolution during the training of the RL method was hypothesised to be optimal, as it is a better representation of the testing environment, where aircraft attempt to avoid each other. Additionally, having CR during training would allow optimisation to focus on the conflicts that a geometric conflict resolution algorithm cannot resolve, instead of focusing on conflicts with small severity. The latter may be the majority, but are easily resolved through conflict resolution. However, without conflict resolution, the RL agent can focus on conflict prevention; having fewer conflicts may result in fewer multi-conflicts situations.

Furthermore, the main objective of the RL agent is to reduce the LoS number, as this is the paramount value considered for safety. However, LoSs are sparse compared to conflicts, which may limit the optimal convergence of the RL method. The LoS number may not be sufficient for the RL to gather enough information to provide a proper understanding of the environment. Looking at conflicts results in more information for the RL agent, as these occur in a larger number. Thus, the latter was hypothesised to warrant more optimal training. It is assumed that, although the total number of conflicts is not directly proportional to the number of LoSs [162], fewer conflicts lead to fewer LoSs.

Finally, testing of the RL agent included traffic densities similar and different to those of the training conditions. The agent was expected to perform better in the traffic density in which it was trained. However, applying the agent to different densities allows one to assess how the efficiency of airspace structures varies with the traffic density. It is hypothesised that the agent may be the least effective at densities higher than the one in which it was trained, as the complexity of the emergent behaviour, and of the consequent solution, increases proportionally to the density.

## 5.7. Experiment: Results

5.7.1. Training of the Reinforcement Learning Agent

The RL agent responsible for setting the airspace structure is trained at a medium traffic density; 75 different traffic scenarios are repeatedly tested. These vary in the number of turns and initial direction(s), as previously described in Section 5.5.3. Each scenario execution corresponded to an episode, which, during the training phase, ran for 20 min. In total, 100.000 episodes are run. Thus, the set of (different) 75 episodes is repeated roughly 1330 times. Four (2 × 2) different RL agents are trained and compared directly to confirm the hypotheses set in Section 5.6.3; two agents are trained in an environment with CR (CR-ON), and two others without CR (CR-OFF). The two agents in each environment will be used to compare the effectiveness of training based on LoSs and conflicts.

Safety Analysis

Figure 5.9 displays the evolution of the total number of pairwise conflicts and LoSs during training without CR. Each point represents the average conflicts or LoSs for the 75 traffic scenario cycles. The shaded and solid lines represent all values and the moving average over the previous 50 values, respectively. For reference, the high variability at the beginning of the training is due to the impact of exploration noise. This noise was intentionally set to be stronger at the initial cycles to promote exploration. Its impact was reduced throughout training. We can see that although the number of conflicts and the

number of LoSs are strongly correlated, a LoS-based reward results in convergence to an optimal value, whereas training based on conflicts did not. The former converged to a minimum number of conflicts and LoSs after approximately 200 cycles of the 75 training traffic scenarios ($200 \times 75 = 15\,k$ episodes in total). Focusing on reducing the number of LoSs also reduced the number of conflicts. In comparison, training based on conflicts did not lead to the finding of an optimal value during a run of 100.000 episodes. There is no clear trend of decrement in conflicts throughout training. One possible reason is that the large magnitude of the total number of conflicts may have had a negative effect on performance. It could be that decreasing the reward per conflict, or normalising the reward value, as is often done in practise to boost performance, could reduce the training time. However, such an investigation was deemed not relevant given the better success with a LoS-based reward.
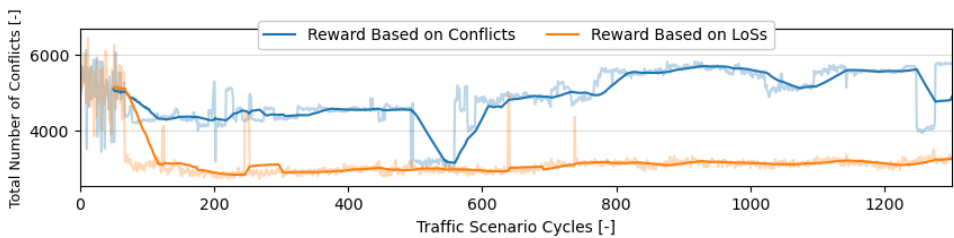
(a) Evolution of the total number of pairwise conflicts (CR-OFF).
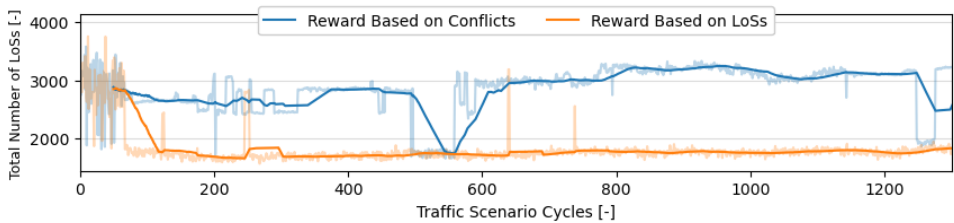
(b) Evolution of the total number of LoSs (CR-OFF).

Figure 5.9: Evolution during training of two RL agents—one trained based on the number of conflicts (in blue), and the other on the number of LoSs (in orange). Conflict resolution was not applied in this environment.
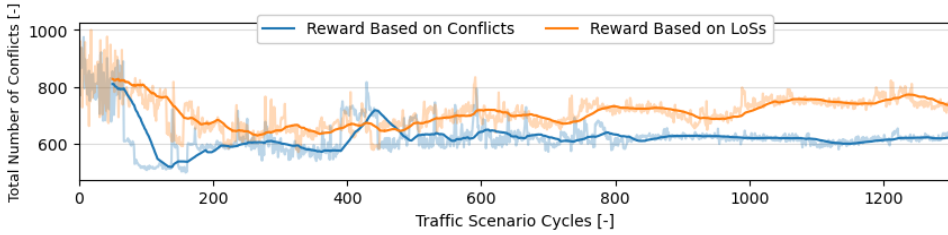
Figure 5.10 shows the evolution of the total number of conflicts and LoSs during training with CR. The differences here are not as great as with the previous RL methods trained without CR. However, contrary to the latter, the agent optimised based on the number of conflicts achieved fewer conflicts and LoSs. We consider this to be a direct consequence of the number of LoS occurrences in the environment. As hypothesised, in an environment with fewer LoSs, the number of conflicts is a better reward formulation, as its higher value provides more information to the RL agent. However, without conflict resolution, the number of LoSs and conflicts is higher. Thus, the LoS provides enough information and, being the paramount safety value, should be used.

The most effective RL agents, 'CR-OFF, LoS'—the agent trained based on LoS in an environment without CR, and 'CR-ON, conf'—the agent trained based on conflicts in

(a) Evolution of the total number of pairwise conflicts (CR-ON).

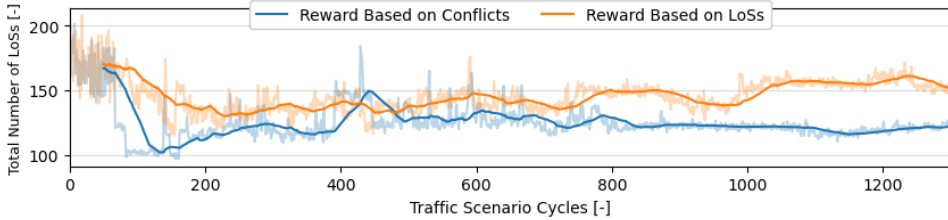(b) Evolution of the total number of LoSs (CR-ON).

Figure 5.10: Evolution during training of two RL agents, one trained based on the number of conflicts (in blue) and the other on the number of LoSs (in orange). Conflict resolution was applied in this environment.

an environment with CR, must now be directly compared within the same conditions. Figure 5.11 shows an example of the airspace structures produced by the two agents. Each row corresponds to one traffic scenario. For example, the first row identifies the traffic scenario in which all aircraft initiated their flights directed East, and no turns were made during the flight. The last row identifies the traffic scenario where aircraft initiated their flights directed East, South, West, or North (with equal distribution); each aircraft made four turns to the left during their flights. The structure outputs by the 'CR-OFF, LoS' and 'CR-ON, conf' agents are displayed in the left and right columns, respectively.

In Figure 5.11, symbol ($\leftarrow$) identifies the airspace structure most commonly used for each RL agent. Agent 'CR-OFF, LoS' used 28 different structures, with structure E,N,S,W,E,N being used in 29 of the traffic scenarios. This structure is employed more often when aircraft are more dispersed throughout the environment, i.e., when more different initial directions are employed. As expected, the more uniform the traffic scenario is, the more the RL agent tends to pick a structure where all directions have similar priority. In comparison, the 'CR-ON, conf' agent used 29 different structures, with structure E,E,E,E,W,(N,S) employed on 10 of the training traffic scenarios. This selection shows a different structure approach than the 'CR-OFF, LoS' agent. The latter opted for a uniform structure that performed relatively well for most traffic scenarios; the 'CR-ON, conf' agent preferred a structure that heavily focused on two directions, East and West, as per Figure 5.11. This is considered to be a direct result of applying conflict resolution. CR resolves many of the conflicts that the 'CR-OFF, LoS' agent prevents with a structure that promotes even segmentation of aircraft throughout the airspace. Thus, the 'CR-ON, conf' agent focuses on conflict 'hotspots' at which CR is ineffective. Given that the most used
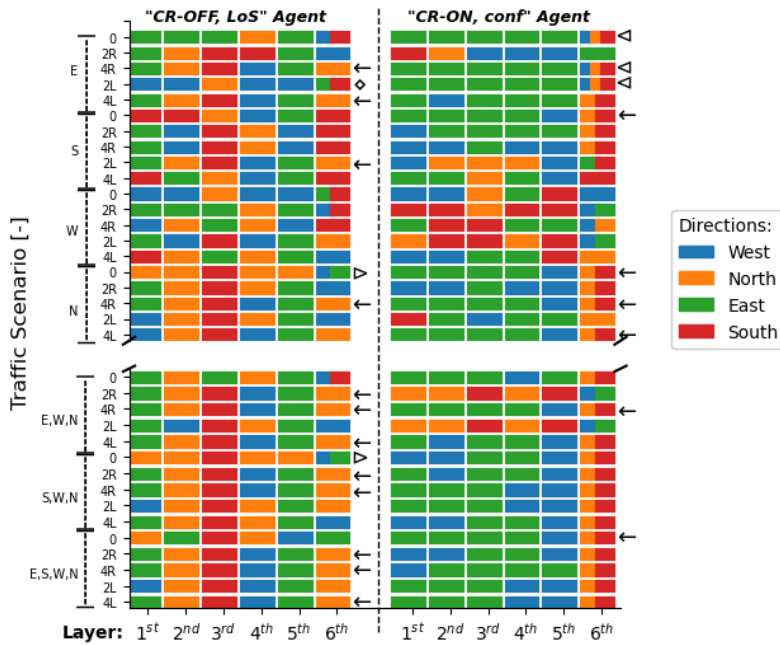
Figure 5.11: Example of airspace structure output by the two best-performing RL agents. On the left: the RL agent trained without CR, based on the number of LoSs. On the right: the RL agent trained with CR, based on the number of conflicts.

structures strongly prioritise the West and East directions, this indicates that the topology of the environment leads to most of the 'hotspots' occurring when aircraft travel in the directions West–East and vice versa.

The efficacy of the segmentation performed by the RL agent is more clearly evaluated when aircraft travel predominantly in one direction. Here, the structure should be optimised to adapt most vertical layers to this direction. For example, the 'CR-OFF, LoS' agent outputs structure N,N,S,N,N,(W,E) (highlighted in Figure 5.11 with the symbol (▷)) for traffic scenarios with: (1) the initial direction North without turns; (2) initial directions South, West, and North without turns. In both situations, the RL method found that guaranteeing a minimum of conflicts between aircraft travelling North had the best impact on reducing LoSs. The 'CR-ON, conf' agent prioritised, for example, structure E,E,E,E,E,(W,N,S) (highlighted in Figure 5.11 with the symbol (◁)) for most traffic scenarios where all aircraft initiated flights heading East. It should be noted that more than one direction in the last layer means that a safeguard was implemented to ensure that all directions were allowed in the final structure, even though the RL agent did not opt to do so. In these cases, this decision was understandable, as aircraft do not follow all directions, and there is a chance that, without the safeguard, the structure would have been even more optimal. Moreover, it is interesting that, often, with multiple directions, the agents chose to focus on one instead of trying to evenly distribute all directions. It seems that, at the current traffic density, strongly optimising one direction results in fewer LoSs and conflicts than trying to equalise all.

Regarding turns (and consequent vertical deviations to move to a traffic layer where the new direction is allowed), the RL agent is able to gather some information on direction changes through the state formulation. The structure selected by the RL agent for no turns was not repeated for the same initial directions(s) when turns were in place. For example, structure W,W,N,W,W,(E,S) (highlighted in Figure 5.11 with the symbol (◇)) was used for the traffic scenario with an initial direction East and two turns to the left (aircraft will first turn North and then West). Thus, the RL favours the directions to which the aircraft moved after turning. Furthermore, the order of directions per vertical layer also affects the final number of vertical deviations that the aircraft must perform. Allowing aircraft to turn left or right by moving one layer upward or downward is often good practise. However, these structures often employ , East–West and North–South in adjacent layers. The impact of climb/descent on final safety will be further analysed during the testing phase.

Figures 5.12 and 5.13 show the results obtained by directly comparing the final structures output by the 'CR-OFF, LoS' and the 'CR-ON, conf' agents in environments with and without conflict resolution, respectively. As previously hypothesised, the agent trained with CR performed better when CR was applied; the 'CR-ON, conf' agent (in green) had fewer conflicts and LoSs (see Figure 5.13). Analogously, the 'CR-OFF, LoS' agent performed better in an environment without CR. However, while the 'CR-OFF, LoS' agent still performed reasonably well in an environment with conflict resolution (often resulting in fewer conflicts and LoSs than the baseline, fixed structure in orange), the 'CR-ON, conf' agent had the worst performing structures when no conflict resolution was applied. This was expected given the structures chosen by this agent (see Figure 5.11). While the 'CR-OFF, LoS' agent selected structures that evenly distributed the existent traffic per the available airspace (which favoured the efficacy of any CR algorithm), the structure output by the 'CR-ON, conf' agent seemed to work directly on the behaviour of the CR algorithm. The agent prioritised directions where the CR algorithm seemed to be unable to resolve conflict 'hotspots'. However, this added pressure in other directions. Although the CR algorithm appeared to be able to resolve conflicts in these directions, without conflict resolution, these directions become concentrations of conflicts.

Furthermore, from the previous results, some conclusions can be drawn regarding the safety impact of singular and multiple directions and the number of turns in the environment:

- Within traffic scenarios starting with a single direction, East and West stand out, resulting in considerably more conflicts. This justifies the emphasis of the 'CR-ON, conf' agent on these directions. Moreover, as expected, when aircraft are initially distributed through more directions, the consequent segmentation results in fewer conflicts and LoSs.

- It was hypothesised that increasing the number of turns would lead to a higher number of conflicts and LoSs. Turns lead to vertical deviations between cruising layers, and having aircraft enter and leave these layers leads to conflict situations [206]. However, within the experimental results, more turns sometimes result in fewer conflicts and LoSs. This is considered a result of the additional segmentation created by the vertical deviations. Aircraft become more distributed throughout the available airspace, as now they also move within the transition layers. This effect appears to have had a positive impact on safety.
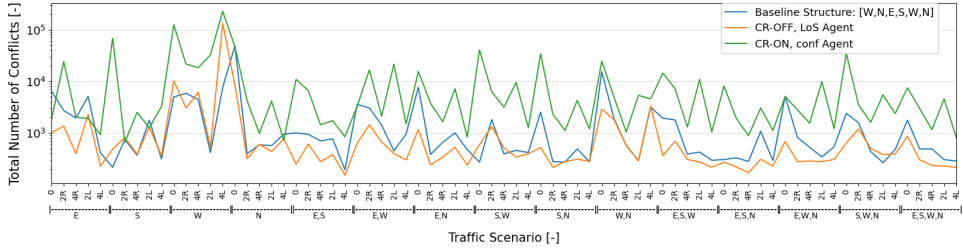
(a) Total number of pairwise conflicts.
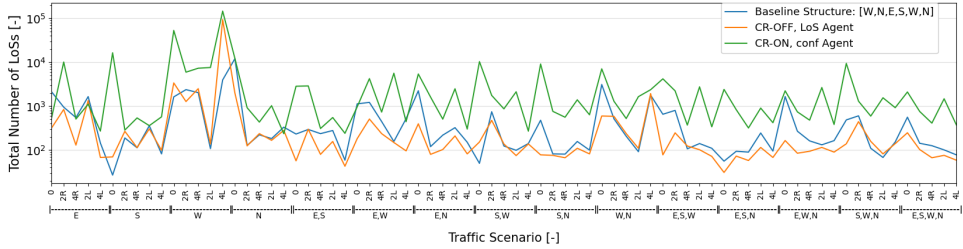
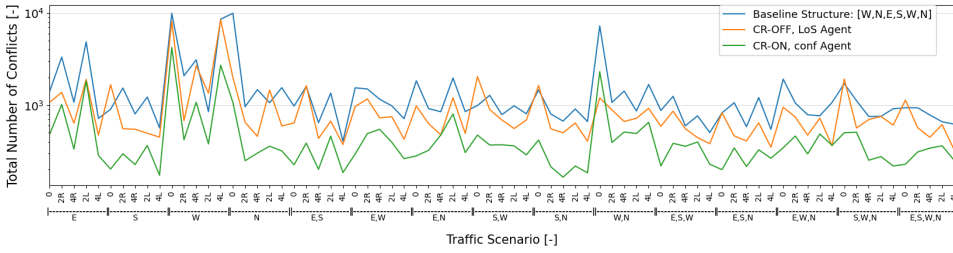(b) Total number of losses of minimum separation.

Figure 5.12: Final comparison of the best RL agents during training in an environment without conflict resolution. The results are directly compared using a baseline, fixed structure. All traffic densities have 75 traffic scenarios, with initial direction(s) and number of turns as defined in Section 5.5.3.

5.7.2. TESTING OF THE REINFORCEMENT LEARNING AGENT

Due to the training results, the 'CR-ON, conf' agent was picked for the forthcoming testing verification with additional traffic scenarios. This RL agent was tested with a total of 225 traffic scenarios; 75 scenarios in each traffic density (i.e., low, medium, and high). The RL agent was previously trained within a medium traffic density; it is interesting to see how it behaved at lower and higher traffic densities. All testing episodes were different from those in which the RL agent trained. For each traffic scenario (i.e., the combination of specific traffic density, initial direction(s), and the number of turns), three repetitions with different flight trajectories were performed. Each traffic scenario lasted one hour. However, note that the state formulation was not modified; it still covered only the first 20 minutes of the traffic scenario. The duration of the traffic scenario was increased to analyse the effect of having a scenario longer than the state contemplated. Additionally, a longer run allowed for a more complete analysis of the impact of employing the structure output by the RL agent vs. a fixed, uniform one. Finally, testing was performed in an environment where aircraft can change speeds and altitudes to avoid conflicts.

SAFETY ANALYSIS

Figure 5.14 shows the mean total number of pairwise conflicts. The RL method reduced the number of conflicts for all traffic scenarios and densities when compared to having a fixed airspace structure. Contrary to the hypothesis, the RL agent did not perform worse at high traffic densities. The airspace structures, which led to an optimal number of conflicts at a medium traffic density, were also applicable to higher traffic densities.

(a) Total number of pairwise conflicts.
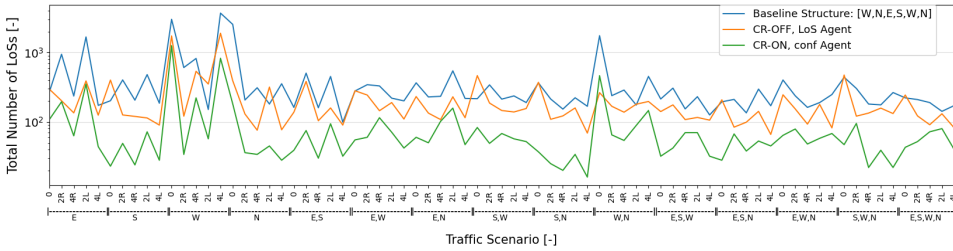
(b) Total number of losses of minimum separation.

Figure 5.13: Final comparison of the best RL agents during training in an environment with conflict resolution. The results are directly compared using a baseline, fixed structure. All traffic densities have 75 traffic scenarios, with initial direction(s) and number of turns as defined in Section 5.5.3.
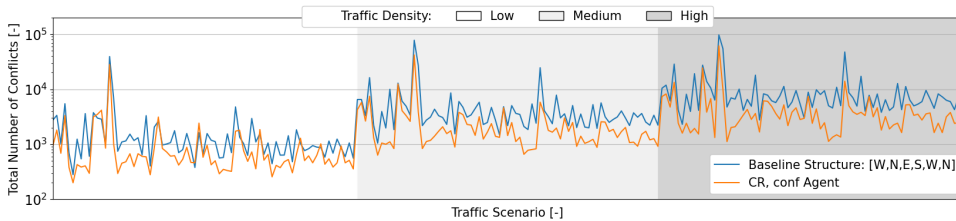
Figure 5.14: Mean total number of pairwise conflicts during testing of the RL agent. All traffic densities have 75 traffic scenarios, with initial direction(s) and number of turns as defined in Section 5.5.3.

Figure 5.15 shows the amount of time spent with a deconflicting state decided by the CR method, rather than following its preferred state. This does not include the time to recovery when aircraft are no longer in conflict and are redirected to their next waypoints. The RL method was able to reduce the time in conflict for all traffic scenarios and densities compared to having a fixed airspace structure. Although the RL reduced both the number of conflicts and the total time in conflict, these do not have a direct correlation. Fewer pairwise conflicts do not necessarily mean less time in conflict per aircraft and vice versa.

Figure 5.16 shows the mean total number of LoSs. The RL method was able to reduce the number of LoSs for all traffic scenarios and densities when compared to having a fixed airspace structure. Although the focus of the RL agent was on reducing conflicts, fewer conflicts led to fewer LoSs. Similarly to the total number of pairwise conflicts (see
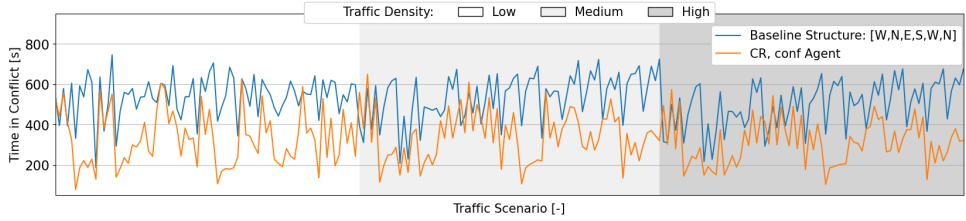
Figure 5.15: Total time in conflict per aircraft. All traffic densities have 75 traffic scenarios, with initial direction(s) and number of turns as defined in Section 5.5.3.

Figure 5.14), there was no decrease in efficacy for higher traffic densities. Interestingly, compared to the fixed structure, the improvement obtained with the RL agent appeared to be stronger in the high traffic density than in the low one.
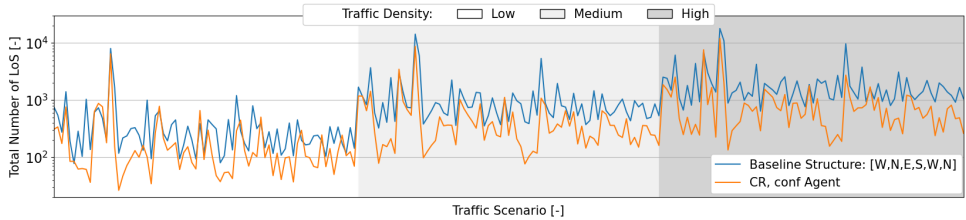
Figure 5.16: Mean total number of losses of separation. All traffic densities have 75 traffic scenarios, with initial direction(s) and number of turns as defined in Section 5.5.3.

Figure 5.17 displays the intrusion severity. For most traffic scenarios, there was no relevant discrepancy between the fixed uniform structure and the structure output by the RL agent. However, with the former, there were outliers in which the mean intrusion severity reached higher values. With a more efficient segmentation, aircraft were better able to maintain a safer distance and were not as close. Finally, no direct correlation was observed between intrusion severity and traffic density.

Figure 5.17: Mean intrusion severity rate. All traffic densities have 75 traffic scenarios, with initial direction(s) and number of turns as defined in Section 5.5.3.

Figure 5.18 presents the relative speed between aircraft in an LoS situation. Higher relative speeds indicate speed heterogeneity that increases complexity in the airspace. Transition layers were in place to minimise the effect of high relative speeds from aircraft

exiting and entering a cruising layer; aircraft only decelerate, turn, and accelerate within the slow layers. Slow layers are considered safer for this state change, as they are expected to be (almost) depleted of aircraft. This might not be the case when multiple aircraft initiate vertical deviations simultaneously. Additionally, a high relative speed can occur in a fast layer. Aircraft performing an resolution manoeuvre in close proximity with different resolution speeds will result in high relative speed conflict situations. On average, the structure output by the RL agent leads to a lower relative speed between aircraft in conflict. However, surprisingly, at lower traffic densities, there are outliers of high relative speeds. This may explain why, in some low-density traffic scenarios, the RL agent was unable to significantly decrease the number of LoSs (see Figure 5.16).
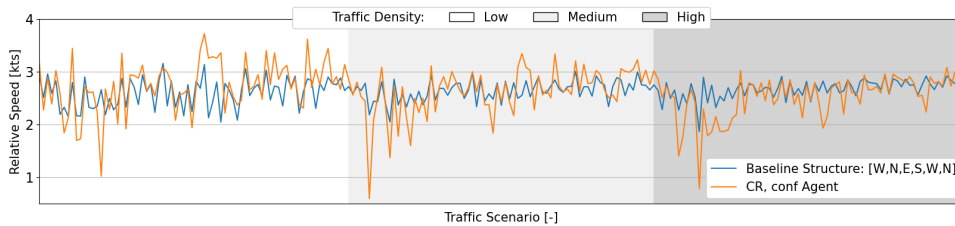
Figure 5.18: Mean relative speed between pairs of aircraft during LoSs with multiple layers. All traffic densities have 75 traffic scenarios, with initial direction(s) and number of turns as defined in Section 5.5.3.

Figures 5.19 and 5.20 show where LoSs occurred for all traffic scenarios tested for the fixed structure and the structures produced by the RL agent, respectively. As shown in Figure 5.16, with the RL agent, there were fewer LoSs. Figure 5.19 shows that, with a uniform structure, most of the LoSs occurred in the transition layers. Figure 5.20 also displays LoSs in the transition layers, but not predominantly. In this case, the last layer stands out as having the most LoSs. This is due to the safeguard implemented on the structure output by the RL agent; if not all directions are included in the structure, the last layer is overwritten to allow for the missing directions. Consequently, this layer may have an agglomeration of aircraft with different headings, leading to a high incidence of LoSs. As per Figure 5.11, the RL agent opted for heavily prioritising certain directions, instead of a more uniform distribution. This approach proves to be more reasonable in the medium traffic density in which the RL agent was trained than in a higher traffic density. In a medium traffic density, including multiple directions in one layer may still result in a number of conflicts that do not cancel out the benefit of prioritising other directions. However, at high traffic densities, a high incidence of traffic in one layer may result in a significant number of conflict chain reactions with a negative impact on safety.

Stability Analysis

Figure 5.21 displays the mean DEP value. A high positive value represents conflict chain reactions, resulting from conflict resolution manoeuvres, causing airspace instability. Previous work on unconstrained airspace showed that applying conflict resolution manoeuvres at high traffic densities tends to create secondary conflicts while reducing LoSs [175]. When free airspace is scarce, having aircraft move laterally and occupy a larger area of airspace often results in more conflicts. However, in this work, as resolution
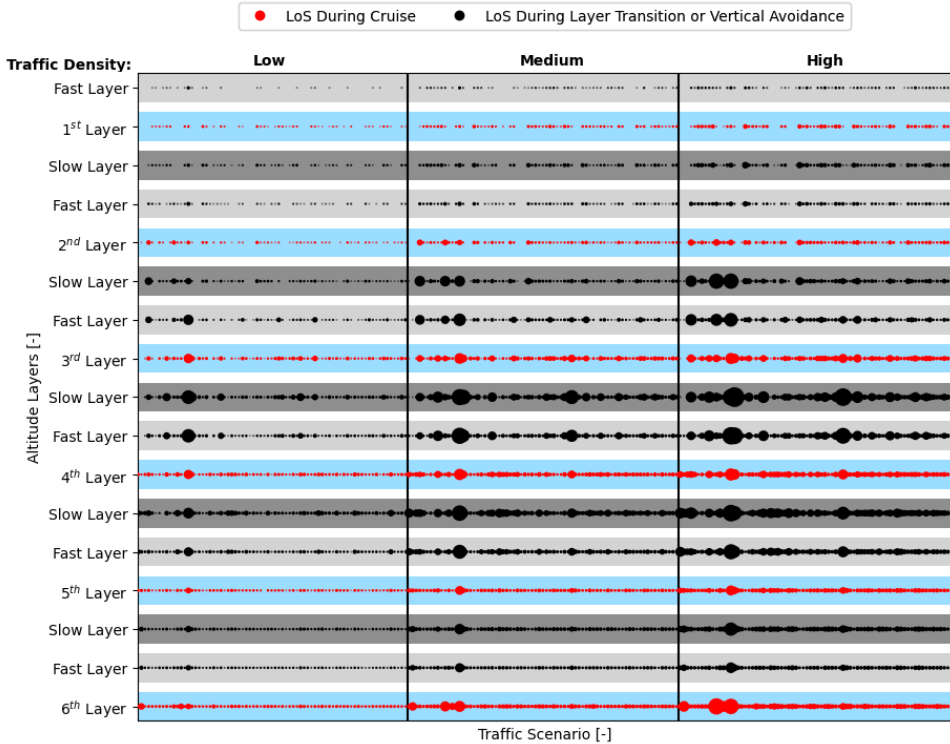
Figure 5.19: Altitudes at which LoSs occur with a baseline structure. The sizes of the points vary between a maximum value of 3128 and a min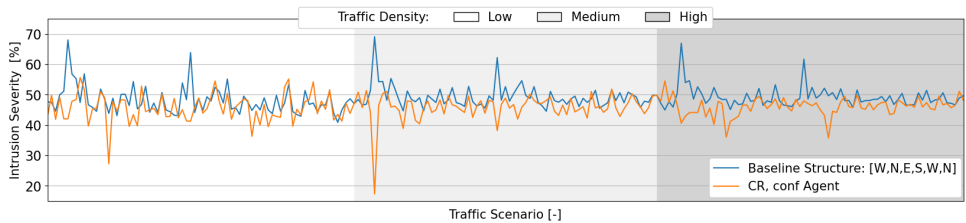imum value of 1 LoS. All traffic densities have 75 traffic scenarios, with initial direction(s) and number of turns as defined in Section 5.5.3.

manoeuvres only move aircraft to a vertical layer dedicated to this purpose, they did not cause secondary conflicts. For most simulated traffic scenarios, employing conflict resolution reduced the number of conflicts compared to a situation without CR.

Figure 5.21 shows peaks very close to -1 and 1, showing how the effect on stability of applying conflict resolution must be correlated with the traffic scenario and flight routes. Additionally, the RL method selects a different structure for each traffic scenario. Some structures may put stress on some traffic layers, which may create conflict 'hotspots' with aircraft continuously resolving and creating conflicts. Interestingly, the highest peaks (i.e., traffic scenarios in which conflict resolution induced instability) were more frequent at lower traffic densities. Negative peaks, where conflict resolution strongly reduced the number of conflicts, occurred more often at higher traffic densities. From these results, it can be derived that the greatest benefit of conflict resolution was the decrease in conflict 'hotspots' resulting from the high incidence of traffic on the same 'road'. Although vertical conflict resolution can be expected to result in secondary conflicts, due to uncertainty regarding the intruder's manoeuvres, it reduces the number of aircraft cruising at the traffic layer by moving some aircraft to the 'fast' layer. At higher traffic densities, the latter effect significantly reduces the number of conflicts. At low traffic densities, conflict
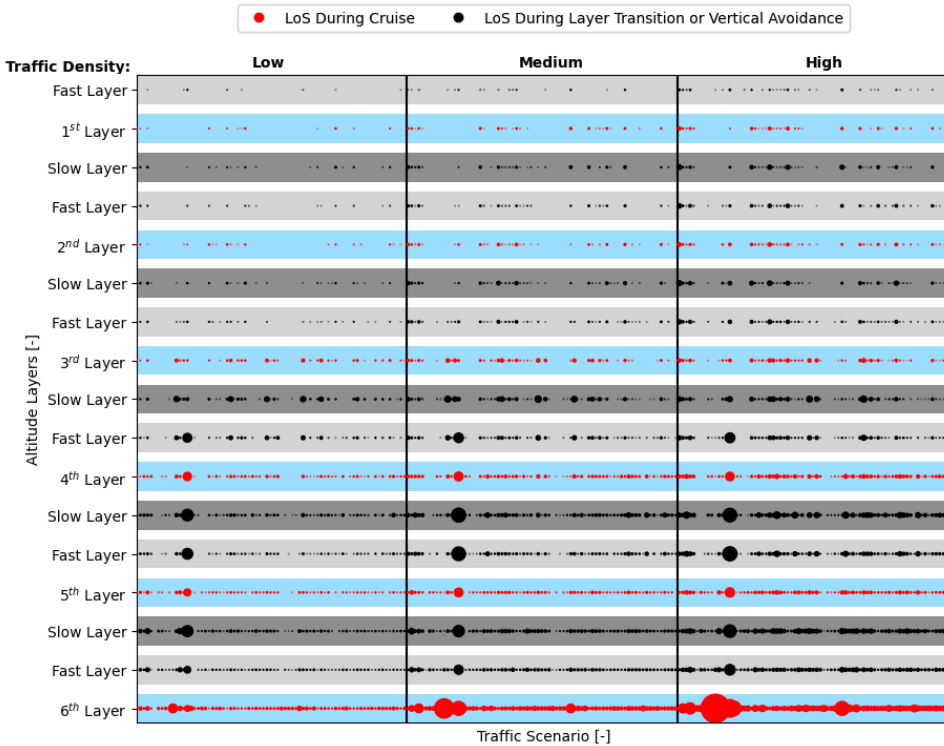
Figure 5.20: Altitudes at which LoSs occurred with the structure produced by the RL agent. The sizes of the points vary between a maximum value of 7519 and a minimum value of 1 LoS. All traffic densities have 75 traffic scenarios, with initial direction(s) and number of turns as defined in Section 5.5.3.

'hotspots' are not as common and, therefore, secondary conflicts due to vertical deviations increase in the total number of conflicts.
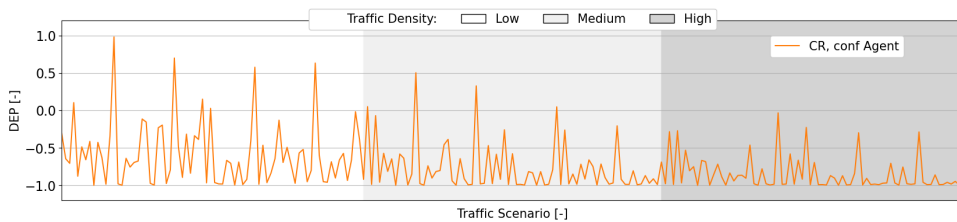
Figure 5.21: Domino effect parameter values. All traffic densities have 75 traffic scenarios, with initial direction(s) and number of turns as defined in Section 5.5.3.

EFFICIENCY ANALYSIS

Figure 5.22 shows the average length of the 3D flight path per aircraft. The differences in flight paths between different structures originate mainly from: (1) different vertical

distances between traffic layers that aircraft occupy throughout their paths, and (2) different numbers of vertical manoeuvres to avoid conflicts. The RL method shows a reduction in the flight path lengths for some of the traffic scenarios when compared to having a fixed and uniform airspace structure; however, this behaviour is not consistent across all traffic scenarios.
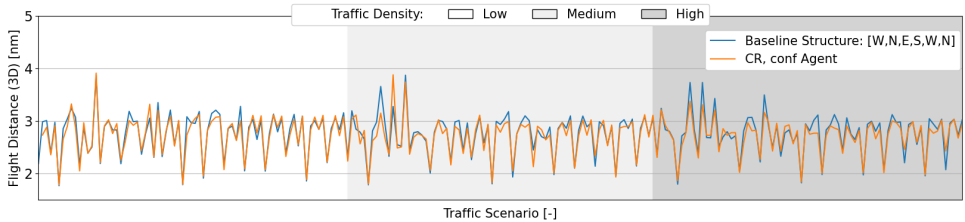
Figure 5.22: Flight path per aircraft. All traffic densities have 75 traffic scenarios, with initial direction(s) and number of turns as defined in Section 5.5.3.

Figure 5.23 shows the average flight time per aircraft. There is no clear improvement in flight time when the RL method is employed. Furthermore, the flight path and time are not directly proportional (see Figure 5.22). A shorter flight path does not necessarily mean a shorter flight time, as sometimes speed changes resulting from conflict resolution manoeuvres also affect flight time.
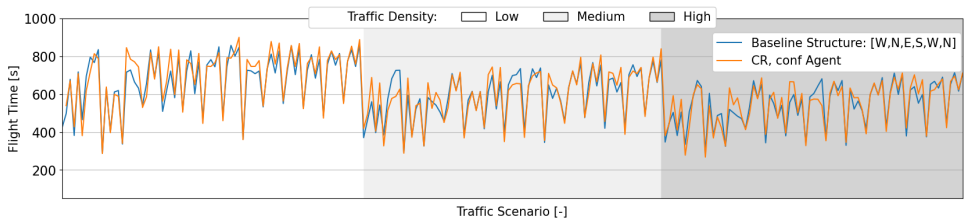
Figure 5.23: Flight time per aircraft. All traffic densities have 75 traffic scenarios, with initial direction(s) and number of turns as defined in Section 5.5.3.

5.8. Discussion

Using reinforcement learning to find an airspace structure that caters to the traffic scenario has a positive effect by reducing the total number of conflicts and losses of minimum separation compared to using a uniform, fixed heading distribution per vertical layer. The latter is optimal for a uniform traffic distribution. However, this is hardly the case in an urban environment where aircraft must respect the topology of static obstacles (e.g., buildings, trees). Adapting the airspace to the operational traffic scenario allows for maximising the efficiency with which the available airspace is utilised. When an inadequate structure is employed, the vertical distribution of traffic will be uneven, reducing the intrinsic safety provided by the layered design.

However, there are still questions regarding this implementation. First, the final structure output by the RL agent seems to be directly correlated with the behaviour of the conflict resolution algorithm. Structures lose efficacy severely when applied in an environment without conflict resolution. Similarly, it is likely that the structures will be less than optimal when different conflict resolution rules are implemented. Which structures benefit capacity is entirely dependent on the conditions of the operational environment. Second, it is not yet clear how the safety of operations can be guaranteed during configuration changes. Traffic scenarios will naturally vary substantially throughout the day; therefore, the airspace structure should also. In this work, the change from one structure to another was not analysed. It was assumed that such transitions would involve several vertical deviations to allow the cruising aircraft to adapt to the new structure. Increasing the number of vertical deviations can increase the number of conflicts. Therefore, it is likely that, during a direct change in the airspace structure, the RL agent must take into account the previous structure to reduce the number of vertical deviations. The following sub-sections dwell further into these subjects.

5.8.1. EFFICACY OF REINFORCEMENT LEARNING

As initially hypothesised, the structure output by the RL agent is highly dependent on whether conflict resolution is applied or not. Without conflict resolution, the airspace structure is optimised to efficiently segment the existing traffic throughout the available airspace. With conflict resolution, structures focus on increasing segmentation for the directions where most conflicts remain after conflict resolution is applied. The structures depend on the topology of the environment and the conflict resolution strategies that are applied. Under different conditions, these structures may not be as optimal. In conclusion, as is the case with most reinforcement learning research, the RL method performs better during testing when trained in a similar environment. The conflict resolution and navigation rules with which the RL agent is trained should be as similar to the real environment as possible.

Furthermore, the reward formulation strongly influences the performance of the reinforcement learning agent. It is often considered that the reward should specify *what* the agent should be doing, but not *how* it should be doing it [226]. The reward should be based on the number of LoSs as this is the paramount value for safety. However, in an environment with conflict resolution, it is often the case that the number of LoSs is not sufficient to provide enough information for proper training. Conflict resolution is often able to resolve most LoSs, and the remaining ones may not be preventable with the airspace structure alone. Thus, the RL agent will not be able to find a clear path through optimisation. Based on the test results, with conflict resolution, the number of conflicts proved to be a more efficient reward formulation. Naturally, this is only valid because it is fair to assume that fewer conflicts will lead to fewer LoSs. Interestingly, the opposite was true for training without conflict resolution, where a LoS-based reward formulation resulted in faster and more optimised training. In this case, the airspace structure had a direct impact on the number of LoSs as these were not resolved by a conflict resolution algorithm. Therefore, the reward formulation should be carefully tuned to the environment.

5.8.2. Conflict Resolution

Previous work on layered airspace structures in urban environments focused on speed-only conflict resolution [172, 206]. However, this was found to be insufficient to prevent conflicts at high traffic densities. As was the case with this work, conflict resolution through heading variation is often not possible. To do so would require knowing the width of every 'road' in order to decide where aircraft can resolve conflicts laterally. Additionally, in a multi-conflict situation, the lateral resolution could potentially cause aircraft to push each other into the surrounding urban infrastructure. Therefore, the remaining degree of freedom is the vertical dimension. By reserving vertical space for upward vertical resolution manoeuvres, we are able to reduce the total number of conflicts and losses of minimum separation. This is due to increasing the amount of manoeuvres aircraft may perform to resolve conflicts, as well as temporally increasing segmentation as some aircraft temporarily move to the layer reserved for vertical resolution.

The results of the current study show the importance of having vertical space specifically reserved for vertical conflict resolution. The vertical manoeuvre will effectively resolve the conflict if: (1) the aircraft moves towards a flight level that is not already densely populated (i.e., moving vertically does not result in secondary conflicts), and/or (2) small relative speeds with aircraft present at the altitude the ownship moves into. The former is achieved by reserving the layer for vertical resolutions only. Aircraft return to the main traffic layer once the conflict has been resolved. The latter is guaranteed as the MVP employs a 'shortest-way-out' solution. The variation will always be as minimal as possible from the aircraft's current state to resolve the conflict. As a result, the relative speed between aircraft travelling in the 'fast' layer will be relatively small, as they opt to travel as close as possible to the desired cruising speed. Thus, the relative speed with other aircraft in the 'fast' layer is not as great as with aircraft in the 'slow' layer, which is purposely used for turns that must be performed at a limited speed necessary to comply with the turn radius.

Another point of concern for the success of vertical deviation is the uncertainty regarding intruder manoeuvres. If the intruders also initiate a similar vertical manoeuvre, the conflict will probably not be resolved. Future research can reduce uncertainty by: (1) applying priority rules defining which aircraft has the right of way; (2) sharing intent information, making aircraft aware of the intruder's future trajectory. However, prior to using intent information, the risks of its implementation must be considered. First, data transmission and processing delays will affect aircraft reaction times, decreasing the effectiveness of resolving short-term conflicts. Second, the aircraft must have the necessary equipment to receive and transmit data if they want to take advantage of this safety. Consequently, the safety of each aircraft also depends on how many of its intruders have this system.

Finally, the effectiveness of resolution manoeuvres depends on the speed and acceleration of the operating aircraft. Aircraft with different performance limits will resolve a different number of conflicts. Additionally, a different number of vertical layers or different safety margins for minimum separation will affect climbing and descending times, which may affect the number of conflicts and losses of minimum separation during vertical manoeuvres. In this work, a 'fast' layer per traffic layer was used for conflict resolution. More layers dedicated to vertical resolution may improve safety, but it would

also increase the number of vertical layers aircraft must traverse. These choices are highly dependent on the operating environment and the aircraft involved.

5.8.3. Advice for Future Work
The following are advised for further research and improvements:

- Exploring more powerful states and reward formulations. For the state formulation, four 'snapshots' of the evolution of the traffic were considered. However, in fast-changing traffic scenarios, the RL agent may need more snapshots to fully understand the progression of traffic over time. Additionally, only safety factors were considered as reward. Future implementations may also benefit from including efficiency elements such as flight path and flight time.
- In this work, the last traffic layer was used to allow directions for which the RL did not allocate space. However, this layer may become a 'hotspot' for conflicts when more than one direction is set. Other possibilities could be researched (e.g., distributing aircraft travelling within 'missing directions' over layers with small heading differences).

5.9. Conclusions
This chapter examined adapting a layered airspace design to the operational traffic scenario through the use of reinforcement learning. The structures produced by an RL agent optimised the use of airspace by segmenting aircraft efficiently throughout the available airspace by taking into account their flight plans. The results showed a reduced number of conflicts and losses of minimum separation when compared to a uniform, fixed structure, which assumed a uniform traffic scenario (as has been the case with previous research). Furthermore, the introduction of layers reserved for vertical resolution manoeuvres further improved the efficacy of conflict resolution.

The application of RL with different environments and rewards showed how optimal structuring is directly related to the behaviour of the aircraft. In an environment where aircraft actively try to resolve conflicts, focusing on prioritising layers for specific directions reduced the total number of conflicts and LoSs. Without conflict resolution, the RL method preferred structures in which aircraft were uniformly distributed throughout the available airspace. Additionally, rewards should be carefully tuned. Safety-wise, focus may be placed on reducing the total number of conflicts and/or LoSs. Prioritisation of one of these two elements, or the weights given to each, must be set according to the number of occurrences during the operation.

However, there are some considerations before this method can be implemented in a real-world scenario. Future work should look into transitions between different structures, and the impacts on safety that may arise from the necessary vertical deviations in order for aircraft to adapt to the new structure. Finally, this work can be extended to more heterogeneous operational environments, in terms of differences in performance limits, as well as preference for efficiency over safety.

PART II:
DIRECT APPLICATION OF REINFORCEMENT LEARNING IN CONFLICT RESOLUTION

6

DISTRIBUTED CONFLICT RESOLUTION WITH REINFORCEMENT LEARNING

This Chapter marks the beginning of Part II of this thesis, where reinforcement learning (RL) is used directly to resolve conflicts. This is different from Part I, where RL was used to reduce the conflict rate and severity within the environment.

Section 6.2 defines an RL method that takes the local observations of each agent and is responsible for distributed conflict resolution. The RL method is tested with different action formulations and different traffic densities to evaluate its performance under complex multi-actor conflict geometries. The results obtained are directly compared to a state-of-the-art distributed CR algorithm.

Cover-to-cover readers will find several differences in the simulated environment in this Chapter when compared to Chapters 3 to 5. Previous chapters focus on investigating whether RL techniques can be used to reduce conflict and severity in an urban environment. This chapter no longer employs an urban environment. The main objective is to investigate whether RL can be used to successfully decide upon the conflict resolution manoeuvre. Geofences are removed not to hinder the training of the RL method. Finally, the readers may choose to skip Section 6.3.5 which describes the Modified Voltage Potential (MVP) method, already described in Chapter 2.

This chapter is based on the following publications:

1. M. Ribeiro, J. Ellerbroek, and J. Hoekstra, Distributed Conflict Resolution at High Traffic Densities with Reinforcement Learning, Aerospace 9 (2022)

ABSTRACT

Future operations involving drones are expected to result in traffic densities that are orders of magnitude higher than any observed in manned aviation. Current geometric conflict resolution (CR) methods have proven to be very efficient at relatively moderate densities. However, at higher densities, performance is hindered by the unpredictable emergent behaviour of surrounding aircraft. Reinforcement learning (RL) techniques are often capable of identifying emerging patterns through training in the environment. Although research has started introducing RL to resolve conflicts and ensure separation between aircraft, it is not clear how to employ it with a higher number of aircraft, and whether it can compare to or even surpass the performance of current CR geometric methods. This work employs an RL method for distributed conflict resolution; the method is completely responsible for ensuring minimum separation of all aircraft during operation. Two different action formulations are tested: (1) where the RL method controls heading, and speed variation; (2) where the RL method controls heading, speed, and altitude variation. The final safety values are directly compared to a state-of-the-art distributed CR algorithm, the Modified Voltage Potential (MVP) method. Although, in general, the RL method is not as efficient as MVP in reducing the total number of losses of minimum separation, its actions help identify favourable patterns to avoid conflicts. The RL method has a more preventive behaviour, defending in advance against nearby surrounding aircraft not yet in conflict, and head-on conflicts while intruders are still far away.

6

6.1. INTRODUCTION

Should the predictions become reality, the aviation field will have to prepare for the introduction of a large number of mass-market drones. Up to 400,000 drones are estimated to provide services in European airspace by 2050 [197]. At least 150,000 are expected to operate in an urban environment for multiple delivery purposes. This is expected to result in traffic densities that are orders of magnitude higher than those observed in manned aviation. As a result, automation of separation assurance in unmanned aviation is a priority, as drones must be capable of conflict detection and resolution (CD&R) without human intervention. Both the FAA [11] and the ICAO [26] have ruled that a UAS must have Sense and Avoid capability in order to be allowed in civil airspace.

Operations with high traffic densities also increase the likelihood of aircraft encountering so called 'multi-actor' conflict situations, where an aircraft is in a state of conflict with multiple other aircraft at the same time. In a pairwise conflict, conflict resolution (CR) methods, or rules, can be implemented so that aircraft work together towards preventing a loss of minimum separation (i.e., implicit coordination). However, these rules alone cannot predict the traffic patterns that emerge from successive conflict resolution manoeuvres and the consequent knock-on effects. As a result, these methods can no longer predict the characteristics that lead to optimal behaviour at these higher densities.

Through continuous improvement, reinforcement learning (RL) can potentially identify trends and patterns in this otherwise unpredictable emergent behaviour. RL can adjust to this emergent behaviour, and develop a large set of rules and weights for different conflict geometries, from the knowledge of the environment captured during training.

In this specific context, a high traffic operating scenario is essentially a multi-agent problem, with emergent behaviour and complexity that arise as a result of aircraft interacting. The knowledge gathered from the actions performed by RL methods to increase safety, can be used to improve current conflict resolution methods and support the decision-making process of air traffic controllers. In this work, we pose the following questions:

1. Can RL methods, by adapting to the global patterns emerging from multi-actor conflicts and knock-on effects, find geometry-specific resolution manoeuvres capable of improving safety compared to current geometric CR methods?
2. Is it possible to derive conflict resolution rules, from the actions performed by the RL method, that improve the performance of current geometric CR methods?

The answer to these questions must take into account the limitations of reinforcement learning approaches. The complexity of training, and consequently the time required for the method to reach an optimal performance, is directly proportional to the number of state-action combinations. Consequently, environments are often discretised to a level such that only a small amount of information is available to the RL method. Additionally, to reach an optimal solution within an acceptable amount of time, degrees of freedom are often limited, as exemplified in previous work such as Pham [227]. Finally, aircraft must prioritise global safety as well as their own. Global safety can only be achieved through coordinated actions. Nevertheless, any level of coordination between agents is non-trivial. A common approach to incite coordination is to resort to multi-agent RL, as exemplified by Isufaj [228]. However, this limits the number of aircraft, as the RL method must learn different policies per aircraft. In summary, it may be that the limitations set in an RL method, to limit its convergence time, may also limit its ability to generalise towards unseen conflict geometries and/or traffic densities. The result would be an RL method with limited rules/solutions, which represents the main issue with geometric CR methods.

In this work, an operational unmanned airspace scenario is implemented with the open-source, multi-agent ATC simulation tool BlueSky [25]. We use the Soft Actor–Critic (SAC) algorithm, as created by UC Berkely [229], for the RL method responsible for conflict resolution. Several versions of the method are tested to determine the best action formulation: (1) action and speed variation only; and (2) action, speed, and altitude variation. Additionally, we consider all aircraft homogeneous and test whether a global safety reward can lead to coordinated movements. The final efficacy and efficiency of the RL method are directly compared to a state-of-the-art geometric distributed CR algorithm, the Modified Voltage Potential (MVP) [15], which resolves conflicts with a minimum path deviation. Finally, the rules that can be used to improve the behaviour of current geometric CR methods are derived from the actions performed by the RL method.

6.2. Conflict Resolution with Reinf. Learning

This section defines the parameters of the RL method responsible for conflict resolution, which guarantees minimum separation between all aircraft. When applying RL to mitigate undesirable emergent patterns resulting from multi-actor conflicts and knock-on effects, several questions follow:

1. What information does the RL method need to successfully resolve conflicts?
2. Which degrees of freedom should the RL method control to perform effective

conflict resolution manoeuvres?

Additionally, two problems arise when using RL in cooperative multi-aircraft situations. First, with each action, the next state depends not only on the action performed by the ownship, but on the combination of that action with the actions performed simultaneously by the surrounding aircraft. From the point of view of each agent, the environment is non-stationary and, as training progresses, modifies in a way that cannot be explained by the agent's behaviour alone. Second, a certain action may be favourable to the ownship but may have negative results on the surrounding aircraft. The latter may, for example, have to perform bigger deviations from their nominal path to avoid a loss of minimum separation with the ownship. In the following subsections, the parameters chosen to tackle these challenges will be discussed.

Finally, to answer the second question (i.e., which degrees of freedom should the RL method control), two different action formulations will be tested and compared directly. For the larger action formulation (i.e, with altitude variation on top of heading, and speed variation), extra information is also added to the state formulation, so that the method knows which values to employ for the altitude variation. Tables 6.1 and 6.2 identify the parameters in the state and action formulations, respectively.

6.2.1. AGENT

This work employs an RL agent responsible for guaranteeing a minimum separation distance between the aircraft at all times. The RL method performs actions based on the information specific to each aircraft, namely its current state, distance, and relative heading to the nearest surrounding aircraft. During training, rewards are based on the global number of losses suffered by all aircraft in the environment.

6.2.2. LEARNING ALGORITHM

An RL method consists of an agent interacting with an environment in discrete timesteps. At each timestep, the agent receives the current state of the environment and performs an action in accordance, for which it receives a reward. An agent's behaviour is defined by a policy that maps states to actions. The goal is to learn a policy that maximises the expected cumulative reward over time. Defining the reward is one of the biggest problems affecting the performance of RL methods. The reward tells the agent *what* to do, not *how* to do it [226]. Nevertheless, the agent should complete the task in the most desirable way. However, it can be that it finds undesirable ways to satisfy the objective, even if the algorithm was implemented flawlessly. Finally, the defined reward also influences convergence speed, and the likelihood of the agent becoming stuck in local optima.

This work uses the Soft Actor–Critic (SAC) as defined in [229]. SAC is an off-policy, actor–critic deep RL algorithm. It employs two different deep neural networks to approximate an action-value function and a state-value function. The actor maps the current state based on the action that it estimates to be optimal, while the critic evaluates the action by calculating the value function. The main feature of SAC is its maximum entropy framework: the actor aims to maximise the expected reward while also maximising entropy. This results in an exploration/exploitation trade-off. The agent is explicitly pushed towards the exploration of new policies while at the same time avoiding being stuck in sub-optimal behaviour.

6.2.3. Action Formulation

The RL agent determines the action to be performed for the current state. The incoming state values are transformed through each layer of the neural network, in accordance to the neurons' weights and the activation function in each layer. The activation function takes as input the output values from the previous layer and converts them into a form that can be taken as input to the next layer. The output of the final layer must be converted into values that can be used to define the elements of the state of the aircraft that the RL agent controls. In this study, all actions are computed using a *tanh* activation function. The *tanh* function outputs values between −1 and +1, which can prevent the output value of the policy network from being too large and causing great state changes per action [230].

The output of the *tanh* function is translated to a variation of the current state of the ownship, as identified in Table 6.1. Note that a variation in heading of −15° and +15° indicates a turn of 15° to the left and 15° to the right, respectively. With regard to speed, the ownship can reduce or increase its speed up to 5 m/s every timestep. A timestep of 1 second is employed. Finally, the vertical speed can decrease or increase every action to a maximum of 2 m/s. These values were empirically tuned. Different values may be used depending on the operating environment. However, the following should be taken into account:

- A greater range of state variation increases the number of different aircraft states that the RL agent may set with each action. Thus, also increasing the number of possible actions and, in turn, convergence time. Additionally, small variations in the agent's actions will have a greater impact on the aircraft's state. This requires the agent to learn a high level of precision.
- Consider the acceleration limits of the aircraft models involved. At each timestep, there is a maximum state variation that an aircraft may achieve. With great state variations, the reward received by the RL method may not be based on the state output by the method. Instead, it will be a result of the maximum variation that the aircraft was able to achieve within the available time. This may make it harder for the RL method to correctly relate actions to expected rewards.

Two different action formulations are tested: (1) the RL method controls heading, and speed variation; or (2) heading, speed, and altitude variation. The two action formulations allow for defining the best usage of the RL method. On the one hand, increasing the size of the action formulation may decrease the optimality of the actions performed by the RL method. The latter must pick from a much larger set of state-action combinations. On the other hand, the RL method also has more control over aircraft and thus may influence the environment to a greater extent.

Table 6.1: Action formulation for the RL method. First, the RL method will be tested controlling only heading and speed variation. Second, it will also control vertical speed variation on top of the former elements.

Action	Limits	Units	Dimension
Heading Variation	[−1,+1] transforms to [−15,+15]	°	1
Speed Variation	[−1,+1] transforms to [−5,+5]	m/s	1
Only when the RL method can also vary altitude:			
Vertical Speed Variation	[−1,+1] transforms to [−2,+2]	m/s	1

6.2.4. STATE FORMULATION

The state input into the RL method must contain the data required for the RL agent to successfully resolve conflicts. Such a decision requires information regarding the current state of the ownship, and the relative position and speed of the surrounding aircraft. However, representing all aircraft in the airspace is impractical. We limit the state information to the closest aircraft, in terms of physical distance. We consider the four closest aircraft. This decision is a balance between giving enough information to the method, so that it can make a based decision while keeping the state formulation to a minimum size. The size of the problem's solution grows exponentially with the number of possible state permutations. The size of the state formulation must be limited to ensure that the method trains in an acceptable time. However, it may be that considering only the four closest aircraft is not ideal for every operational scenario; such must be decided on a case-by-case basis.

As defined in Table 6.2, the RL method is informed of the ownship's current heading, bearing to target, and current speed. Regarding surrounding aircraft, it has knowledge of their current distance to the ownship, relative heading, distance at the closest point of approach (CPA), and time to CPA. Figure 6.1 depicts the data used to defined the relation between ownship and surrounding aircraft in the horizontal plane. When the agent also controls altitude variation, it additionally receives information on the ownship's current altitude and relative altitude to the closest aircraft.

Table 6.2: State formulation for the RL method. Note that the current altitude and relative altitude to aircraft, are only added to the state formulation when the RL method controls altitude on top of heading and speed variation.

Element	Dimension
Current heading	1
Relative bearing to target	1
Current speed	1
Current distance to #surrounding aircraft	#surrounding aircraft
Distance at CPA with #surrounding aircraft	#surrounding aircraft
Time to CPA with #surrounding aircraft	#surrounding aircraft
Relative heading to #surrounding aircraft	#surrounding aircraft
Only when the RL method can also vary altitude:	
Current altitude	1
Relative altitude to #surrounding aircraft	#surrounding aircraft

6.2.5. REWARD FORMULATION

The reward given to the RL agent is primarily based on safety. However, within safety, several factors may be considered. The paramount objective is to lead the agent to favour deconflicting actions that reduce the likelihood for LoSs. Thus, the reward is set based on the number of LoSs. Moreover, to favour coordinated manoeuvres which improve global safety, the reward given for each action is based on the number of LoSs suffered by all aircraft in the previous time step. A value of -1 is added to the reward for every LoS that occurred in the environment since the action was initiated, i.e., in the last timestep. A negative factor of this reward approach is that the reward to an action will be affected by unrelated LoSs, suffered by other (far away) aircraft. Such may
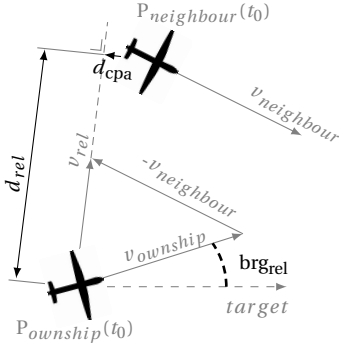
Figure 6.1: Parameters defining the (horizontal) representation of the relationship between the ownship and its closest neighbours. $P_{ownship}(t_0)$ and $P_{neighbour}(t_0)$ denote the ownship and the neighbour's initial position, respectively. $v_{ownship}$ is the observed aircraft velocity vector, $v_{neighbour}$ is the neighbour's velocity vector, and v_{rel} is the relative velocity vector. d_{rel} is the distance vector, and d_{CPA} indicates the distance at the closest point of approach (CPA). brg_{rel} represents the relative bearing to target.

increase convergence time, or even affect the capacity of the RL method to converge towards optimal values. The consequences of this reward implementation will be further examined in the discussion of the results.

6.3. EXPERIMENT: CONFLICT RESO. WITH REINF. LEARNING

The following sections define the properties of the performed experiment. The latter uses RL to perform optimal deconflicting manoeuvres at high traffic densities. The experiment involves a training and a testing phase. First, the RL method is trained continuously with a set of 16 known traffic scenarios. For reference, without conflict resolution, each training episode has on average about 1000 conflicts in 20 minutes running time. The evolution of the amount of LoSs and conflicts, for every training episode, is directly compared with the average number of LoSs and conflicts when running these 16 scenarios with the MVP. Additionally, the final optimal actions of the RL method for every conflict situation are directly compared with the ones that MVP would perform for those exact situations. Each training scenario runs for 20 minutes. Second, the RL method is tested with unknown traffic scenarios at the same and different traffic densities that it was trained in. The safety, stability, and efficiency results of the method are directly compared to running the same scenarios with the MVP method. Each testing scenario runs for 30 minutes.

6.3.1. FLIGHT ROUTES
The measurement area is a square-shaped with an area of 144 NM^2. The aircraft spawn locations (origins) are placed on the edges of this area, with a minimum spacing equal to the minimum separation distance, to avoid conflicts between recently spawned aircraft and aircraft arriving at their destination. All aircraft fly a straight route towards their destination, at the same altitude level. Three waypoints are added between origin and target points which aircraft must pass through in order. Ideally, aircraft would only operate within the measurement area, thereby ensuring a constant density of aircraft

within that area. However, aircraft may temporarily leave the measurement area during the resolution of a conflict and should not be deleted in this case. Therefore, a second, larger area encompassing the measurement area is considered: the experiment area. As a result, aircraft in a conflict situation close to their origin or destination are not deleted incorrectly from the simulation. Ultimately, an aircraft is removed from the simulation once it leaves the experiment area. We assume a no-boundary setting, with sufficient flight space around the measurement area, to avoid edge effects from influencing the results.

6.3.2. APPARATUS AND AIRCRAFT MODEL
The open air traffic simulator BlueSky [25] is used in order to test the efficiency of RL in resolving conflicts. The performance characteristics of the DJI Mavic Pro were used to simulate all vehicles. Here, speed and mass were retrieved from the manufacturer's data, and common conservative values were assumed for turn rate (max: $15°/s$) and acceleration/breaking ($1.0\,\text{kts/s}$).

6.3.3. MINIMUM SEPARATION
The value of the minimum safe separation distance may depend on the density of air traffic and the region of the airspace. For unmanned aviation, there are no established separation distance standards yet, although $50\,\text{m}$ for horizontal separation is a value commonly used in research [59], and will therefore be used in these experiments. For vertical separation, $15\,\text{ft}$ was assumed.

6.3.4. CALCULATION OF CLOSEST POINT OF APPROACH (CPA)
This work assumes linear propagation of the current state of all aircraft involved to calculate the CPA between two aircraft. Using this approach, the time to CPA (in seconds) is calculated as:

$$t_{CPA} = -\frac{\vec{d}_{rel} \cdot \vec{v}_{rel}}{\vec{v}_{rel}^2}, \tag{6.1}$$

where $\vec{d}_{rel}$ is the Cartesian distance vector between the involved aircraft (in meters), and $\vec{v}_{rel}$ the vector difference between the velocity vectors of the involved aircraft (in meters per second). The distance between aircraft at CPA (in meters) is calculated as:

$$d_{CPA} = \sqrt{\vec{d}_{rel}^2 - t_{CPA}^2 \cdot \vec{v}_{rel}^2}. \tag{6.2}$$

Both t_{CPA} and d_{CPA} are added to the state formulation of the RL method as previously defined in Table 6.2.

6.3.5. CONFLICT RESOLUTION (MODIFIED VOLTAGE POTENTIAL ONLY)
As previously mentioned, the results obtained with the RL method will be directly compared to those obtained with the state-of-the-art CR method MVP. This work employs the method as defined by Hoekstra [15, 231]. An important difference between the RL method and MVP, from the very beginning, is how they select 'intruders'. We apply as little bias on the state formulation of the RL method as possible. Thus, we simply select

aircraft based on their distance to the ownship. It may be considered that adding only the aircraft that the ownship is in conflict with might be more efficient. However, the RL method would then not have a full perception of the consequence of its actions when moving toward a non-conflicting nearby aircraft.

MVP, on the other hand, considers only aircraft that are actually in conflict in its resolution. Conflicts are detected when $d_{CPA} < R_{PZ}$, and $t_{in} \leq t_{lookahead}$, where R_{PZ} is the radius of the protected zone (PZ), or the minimum horizontal separation, and $t_{lookahead}$ is the specified look-ahead time. A look-ahead time of 300 seconds is used for conflict detection and resolution. This value was selected since, empirically, it was found to result in the best behaviour of the MVP method in this specific simulation environment. Note that this is a larger look-ahead time than typically used in unmanned aviation, where values can be even less than 1 minute. Nevertheless, these values are often considered in constrained airspace, as larger look-ahead times would result in the inclusion of false conflicts past the borders of the environment [204]. Additionally, it is likely this large look-ahead time would perform worse in environments with uncertainty regarding intruders' current position and future path. Finally, delays in data transmission and severe meteorologic conditions are often a source of errors in the estimation of future positions.

The behaviour of MVP is displayed in Figure 6.2. MVP uses the predicted future positions of both ownship and intruder CPA. These calculated positions 'repel' each other, and this 'repelling force' is converted to a displacement of the predicted position at CPA. The resolution vector is calculated as the vector starting at the future position of the ownship and ending at the edge of the intruder's protected zone, in the direction of the minimum distance vector. This displacement is thus the shortest way out of the intruder's protected zone. Dividing the resolution vector by the time left to CPA, yields a new speed, which can be added to the ownship's current speed vector resulting in a new advised speed vector. From the latter, a new advised heading and speed can be retrieved. The same principle is used in the vertical situation, resulting in an advised vertical speed. In a multi-conflict situation, the final resolution vector is determined by summing the resolution vectors from all intruders. By taking the shortest way out, each aircraft in a conflict will take (opposite) measures to evade the other in a way that makes MVP implicitly coordinated.
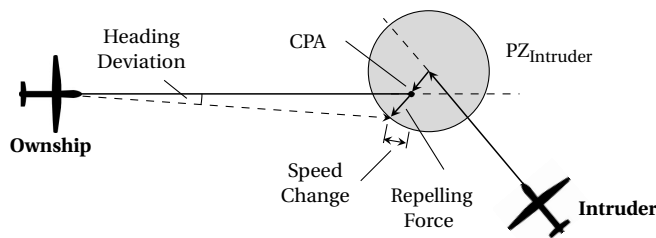

Figure 6.2: Modified Voltage Potential (MVP) geometric resolution. Adapted from [15].

6.3.6. INDEPENDENT VARIABLES
The main independent variable is the method used to resolve conflicts and ensure minimum separation between all aircraft; this is either the RL or the MVP method. Both

the training and testing results of the RL method are directly compared to the results obtained when the same traffic scenarios are run with the MVP method instead. Additionally, different action formulations are employed to analyse how the RL method reacts to different degrees of freedom: (1) the CR method only performs heading and speed variation to resolve conflicts; (2) the CR method uses heading, speed, and altitude variation. The variations of heading, speed, and altitude performed by the RL method during training, are directly compared with the ones that the MVP would perform for the exact same conflict situations.

Finally, during testing, different traffic densities are introduced to analyse how the RL method performs at traffic densities in which it was not trained. These range from low to high according to Table 6.3. At high densities, vehicles spend more than 10% of their flight time avoiding conflicts [193]. The RL agent is trained at a medium traffic density, and is then tested with low, medium, and high traffic densities. In this way, it is possible to assess the efficiency of an agent performing in a traffic density different from that in which it was trained.

Table 6.3: Traffic volume used in the experimental simulations.

Traffic density	Training (20 minutes simulation) Medium	Testing (30 minutes simulation)		
		Low	Medium	High
Number of aircraft per $10\,000\,\text{NM}^2$	40000	20000	40000	60000
Number of instantaneous aircraft	576	288	576	863
Number of spawned aircraft	886	665	1330	1994

6.3.7. Dependent Variables

Three different categories of measures are used to evaluate the effect of the different conflict resolution methods in the simulation environment: safety, stability, and efficiency.

Safety Analysis

Safety is defined in terms of the number and duration of conflicts and losses of minimum separation. The most important factor is a reduction in the total number of LoSs compared to a situation in which no conflict resolution is performed. Additionally, LoSs are distinguished on the basis of their severity according to how close aircraft get to each other, where a low separation severity is preferred. The latter is calculated as follows:

$$LoS_{sev} = \frac{R_{PZ} - d_{CPA}}{R_{PZ}}. \tag{6.3}$$

Stability Analysis

Stability refers to the tendency for tactical conflict resolution manoeuvres to create secondary conflicts. In the literature, this effect has been measured using the Domino Effect Parameter (DEP) [151]:

$$DEP = \frac{n_{cfl}^{ON} - n_{cfl}^{OFF}}{n_{cfl}^{OFF}}, \tag{6.4}$$

where n_{cfl}^{ON} and n_{cfl}^{OFF} represent the number of conflicts with conflict resolution ON and OFF, respectively. A higher DEP value indicates a more destabilising method, which creates more conflict chain reactions.

Efficiency Analysis

Efficiency is evaluated in terms of the distance and duration of the flight. Significantly increasing the path travelled and/or the duration of the flight is considered inefficient.

6.4. Experiment: Hypotheses

It is hypothesised that the RL method will be able to understand the concept of minimum separation and resolve the majority of conflicts. A direct performance comparison between the RL and MVP methods is uncertain at this point. On the one hand, the latter can perform geometric manoeuvres that guarantee conflict resolution with minimum path and state deviation. This is a level of precision that limits the creation of secondary conflicts. It is hypothesised that the RL method will likely perform greater state variations than the MVP method. On the other hand, the RL method is capable of adapting to the global patterns emerging from multi-actor conflicts, and knock-on effects from successive resolution manoeuvres. It can create a much larger set of rules and solutions for resolution of different conflict geometries. Whether this can help the RL method surpass the performance of the MVP method remains to be seen. Nevertheless, it is also hypothesised that the manoeuvres performed by the RL method can provide guidelines to improve the efficacy of the existing distributed, geometric CR algorithms.

Additionally, CR methods are normally able to resolve more conflicts as the degrees of freedom increase. It is expected that the MVP method will resolve more conflicts when it can vary altitude, heading, and speed versus a situation where it can only vary heading and speed. However, as the state formulation increases, so does the set of possible state-action combinations. This often results in longer training times, and not so optimal choices by an RL method. Thus, it is hypothesised that when the RL method only controls heading and speed, its final efficacy in resolving conflicts will be closer to that of the MVP method.

Finally, the RL method will be tested with different traffic densities. It is hypothesised that the method will be most effective at low and medium traffic densities. The method is trained at medium traffic densities. The higher the traffic density, the more complex the conflicts' geometries are to resolve, with each aircraft potentially facing multiple conflicts with multiple simultaneous intruders. Thus, the optimal actions learnt during training may not be sufficient to resolve conflicts with a higher number of intruders. Moreover, the state formulation contains only information regarding the four closest neighbouring aircraft. In a conflict situation with a considerably higher number of near-by aircraft, the method may not have enough information to resolve all conflicts with all these aircraft. As previously mentioned, the limitation of the state and action formulations to improve convergence times, may limit the ability of the RL method to generalise its actions to operational environments with different characteristics.

6.5. Experiment: Results

In the results section, a distinction is made between the training and the testing phases. The former shows the evolution of the RL method while training with a repeating cycle of 16 episodes, at medium traffic density, to investigate how well the RL method learns. In total, 300 episodes are run. During training, each episode runs for 20 minutes. Second, the trained RL method is tested with different traffic scenarios at a low, medium, and high traffic density. For each traffic density, 3 repetitions are run with 3 different route scenarios, for a total of 9 different traffic scenarios. During the testing phase, each scenario runs for 30 minutes. Safety, stability, and efficiency results of the RL method are directly compared to the ones obtained when running the same scenarios the MVP method.

### 6.5.1. Training of the RL Agent for Conflict Resolution

This section presents the results of the training phase of the RL method. The latter is trained with a repeating cycle of 16 episodes at medium traffic density. Its results are directly compared with the average total number of conflicts and LoSs obtained with the MVP method with the same traffic scenarios. Furthermore, the actions performed by the method are examined in order to understand the conflict resolution decisions adopted.

Safety Analysis

Figure 6.3 shows the evolution in safety performance of the RL method in terms of losses of separation (Figure 6.3(a)) and number of conflicts (Figure 6.3(b)) for both action formulations. The values obtained when both the MVP and RL methods control only the heading and speed variations are indicated by 'MVP Method (H + S)' and 'RL method (H + S)', respectively. The values with '(H + S + A)' indicate the performance of the previous methods when these control altitude variation, on top of heading and speed variation. The values presented for the MVP method represent the average values for all 16 training episodes. With and without altitude deviation, the RL method is able to converge towards actions that resolve the great majority of the conflicts. Considering the total number of conflicts in Figure 6.3(b), the RL method is able to resolve 99.7 % and 99.83 % of the conflicts with heading + speed and heading + speed + altitude control, respectively. Contrary to what was hypothesised, the RL method is able to achieve safety results comparable to those of the MVP when it can control more degrees of freedom. This indicates that the method is able to use the increased number of possible actions to resolve conflicts effectively.

With only heading and speed variation, the RL method has a higher total number of LoSs than MVP. However, MVP has fewer conflicts than the RL method. Tactical CR manoeuvres typically create secondary conflicts. Deviating from the nominal path, in order to avoid conflicts, often results in a longer flight path. At high traffic densities, conflict-free airspace is scarce, and when each aircraft requires a larger portion of the airspace it often results in more conflicts. MVP employs a 'shortest-way-out' resolution strategy, limiting the space used by each aircraft, which in turn limits conflict chain reactions. The RL method resolves conflicts with path deviations larger than MVP, resulting in a higher number of secondary conflicts. The latter in turn leads to a higher final count of LoSs.

When altitude variation is also controlled, the RL method is able to reach the same level of efficacy in resolving conflicts as MVP. With the three degrees of freedom, the ac-
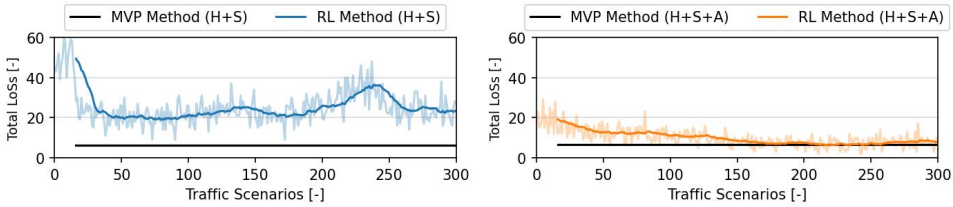
(a) LoS evolution during training.
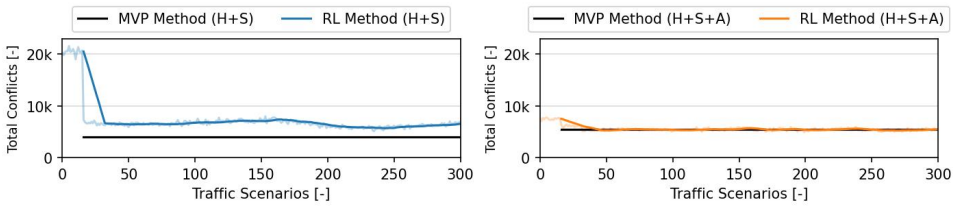
(b) Conflict evolution during training.

Figure 6.3: Evolution of the total number of LoSs and conflicts resolved by the RL method during training.

tions of the RL method do not have to be as precise. With all aircraft travelling at the same altitude, the success of conflict resolution may lie in having heading variations just large enough to resolve conflicts, but not so large that they move the ownship into the flight path of other aircraft. However, vertical deviation is a powerful tool that allows moving away from this one layer of traffic. As the ownship is moving to 'free space', it is less likely that deconflicting actions will result in secondary conflicts. Thus, such precise heading and altitude deviations are not as crucial.

Figure 6.4 shows the difference in the actions carried out by the RL and MVP methods, for the same conflict situations, in all training episodes. In this case, the episodes are run with the conflict resolution decisions by the RL method. Simultaneously, the actions that MVP would output for every conflict situation are recorded, making sure that the actions can be directly compared. The graphs on the left show the difference in actions when both MVP and RL control only heading and speed variation. The top graph indicates the smallest angle difference between the heading solutions produced by the two methods. First, the difference in heading is at most 20° between the two methods. Second, the negative values indicate that the solutions output by the MVP method are, in general, directed more towards the left than the solutions by the RL method. This may be because the RL method has a preference for resolving conflicts by turning aircraft in one direction, in this case right. This preference is not an optimal solution for every conflict geometry, but it is likely a result of the RL method finding a local optimum with this decision. These local optima are often dependent on the method's initialisation, and a product of chance.

Regarding the horizontal speed (middle graph), negative values represent a decrease in the current speed. In general, the MVP employs slightly stronger speed variations to resolve conflicts than the RL method. In comparison, the heading and speed actions produced by the RL method are much similar to those of the MVP method, when both
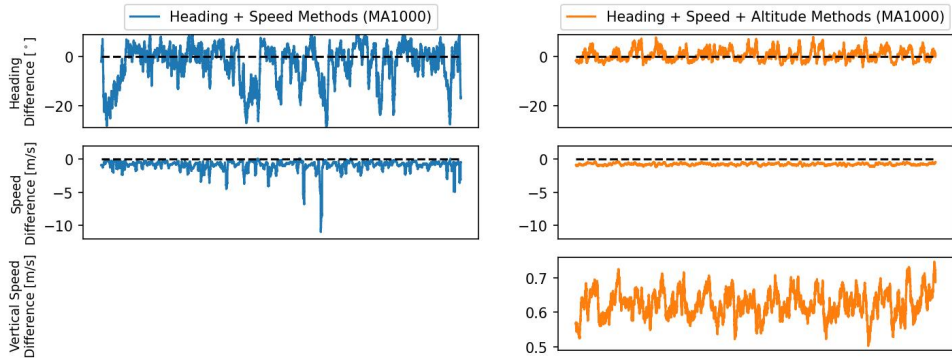
Figure 6.4: Difference between actions performed by the RL and MVP methods for the same conflict situations.

methods can also move aircraft in the vertical dimension. It seems that, as the RL method can vary more degrees of freedom, it has learnt not to vary each degree as strongly to resolve conflicts. Finally, the RL method typically employs slightly stronger climbing actions than the MVP method.

The following sections will further explore the differences between the actions of the RL and the MVP methods.

HEADING VARIATION

Figure 6.5 connects elements of the state formulation with the average heading variation chosen by the RL and MVP methods. Interestingly, the RL method performs larger heading variations whenever the average distance at CPA or time to CPA, is smaller, but not when both are small. At every timestep, contrary to the RL method, the heading variation performed by the MVP is not limited. Thus, the extremes of the heading variation are stronger. On average, MVP performs very small heading variations, which are likely a result of the 'shortest-way-out' resolution strategy. However, MVP still scarcely resorts to large values. Note, however, that the great state changes output by the MVP method are likely not achievable within the observation timestep, due to performance limits. The effective state changes are likely smaller in these cases.

Taking into account all variables, it appears that the current distance to intruders played a bigger role in the RL method's decisions, than the distance at CPA. The method adopts strong heading variations when surrounding aircraft are at a very short distance. This behaviour is not as consistent with a smaller distance at CPA and/or short times to CPA. This may be because the method was able to relate negative rewards with a small distance between aircraft. Additionally, the RL method has information over the closest surrounding aircraft. These are not necessarily always the intruders with the shortest distance at CPA or time to CPA. It could be that, as a result, the method has learnt to prioritise current distance.

Additionally, as expected, the closer the intruders are, the stronger the deconflicting heading variation is. However, the RL method still resorts to strong heading deviations in some situations where the intruders are far away. Taking into account the average relative

heading, these intruders would result in head-on conflicts if they were to continue with their current state. Thus, it seems that the RL method is adopting preventive actions against possible future severe conflicts. Finally, the method shows a strong preference for turning one direction, in this case right, independently of the positions of the intruders. It may have been found that this resulted in some degree of coordination between aircraft.
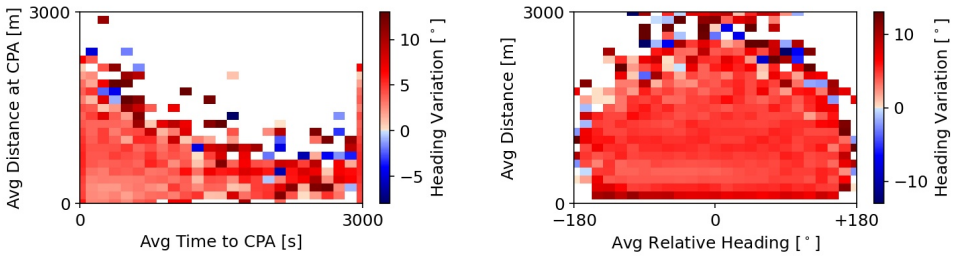
(a) RL method controlling heading, and speed variation.
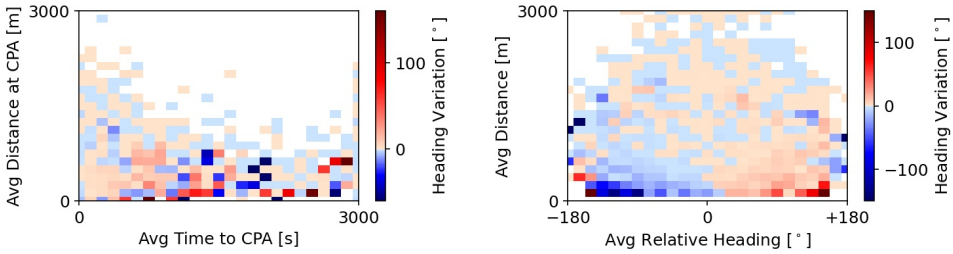
(b) MVP method controlling heading, and speed variation.

Figure 6.5: Heading variation by both the RL and the MVP methods controlling heading and speed variation.

Figure 6.6 presents the same values as Figure 6.5, with the difference being that the methods can now also vary altitude. As seen in Figure 6.4, the RL method performs, on average, smaller heading variations in this case. Furthermore, here, the RL method seems to prefer heading deviations to the left. The RL method is not capable of learning when right or left might be a better option, depending on the conflict geometry. Instead, it learns that a common direction used by all aircraft results in some sort of coordination. The actions produced by the MVP are very similar to those seen in Figure 6.6(b). This is expected; altitude variation is decoupled from heading and speed in the calculation of the resolution manoeuvre by the MVP. Thus, adding altitude variation will not alter MVP's heading and speed deviations. However, the values in Figures 6.5(b) and 6.6(b) are not exactly the same. Figure 6.5(b) shows the actions of the MVP method as it would respond to the conflict situations that occur in the traffic episodes run with the RL method that controls speed and heading variation. In comparison, Figure 6.6(b) has different conflict situations, since the episodes are run with a different RL method that now also controls altitude. Different resolution manoeuvres lead to different secondary conflict situations.

(a) RL method controlling heading, speed, and altitude variation.

(b) MVP method controlling heading, speed, and altitude variation.
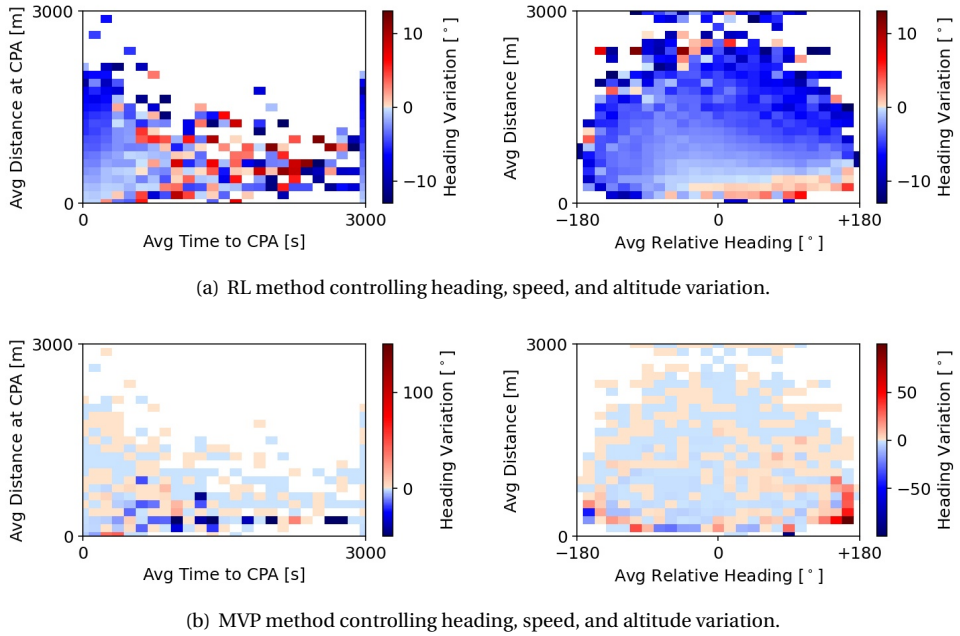
Figure 6.6: Heading by the RL and the MVP methods controlling heading, speed, and altitude variation.

SPEED VARIATION

Figure 6.7 shows the speed variation produced by the RL and MVP methods when these control heading and speed variation. Unlike the MVP method, the RL method often chooses to reduce speed. This explains the higher average speed variations previously seen in Figure 6.4. The RL method opts for a defensive position, where speeds may be reduced to increase the time to CPA. Naturally, this has a negative effect on efficiency, as it increases flight time. However, efficiency is not included in the reward formulation of the method, and thus the method is unaware of this. The RL method only increases speed when surrounding aircraft are very close in proximity, likely in an attempt to rapidly increase the distance between the ownship and these aircraft.

Finally, Figure 6.8 shows the speed variation performed by the RL and MVP methods when these control heading, speed, and altitude variation. When the RL method also controls altitude, it outputs smaller horizontal speed variations.

VERTICAL SPEED VARIATION

Figure 6.9 displays the vertical speed variation performed by the RL and MVP methods when these control the heading, speed, and altitude variation. The RL method learnt to disperse aircraft quite efficiently, using climbing and descent actions almost equally. In practise, the RL method is creating three separated layers of traffic. MVP employs smaller vertical speed variations to resolve conflicts. Here, MVP is again more precise than the RL method, employing only the minimum altitude variation necessary to resolve conflicts. The RL method disperses aircraft more significantly through the airspace. This
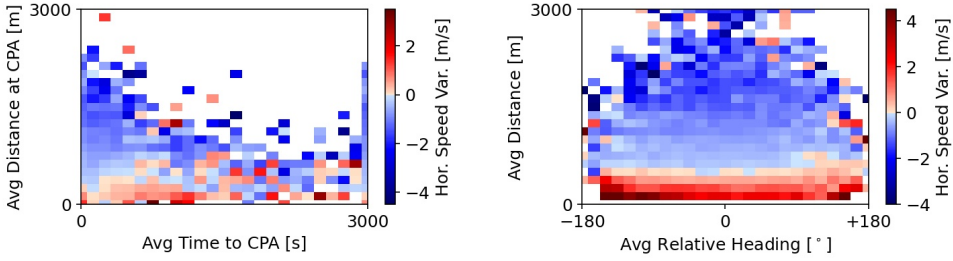
(a) RL method controlling heading, and speed variation.
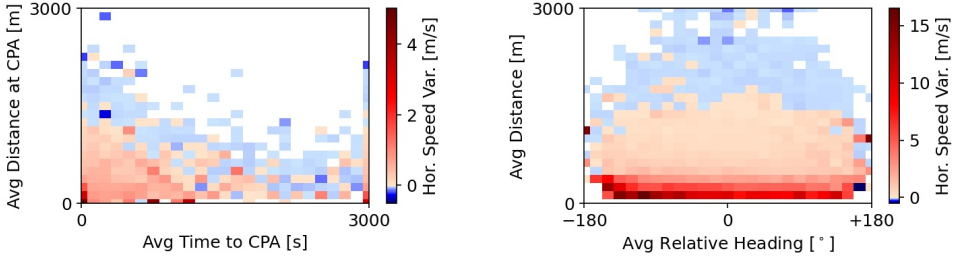
(b) MVP method controlling heading, and speed variation.

Figure 6.7: Speed variation by the RL and the MVP methods controlling heading and speed variation.
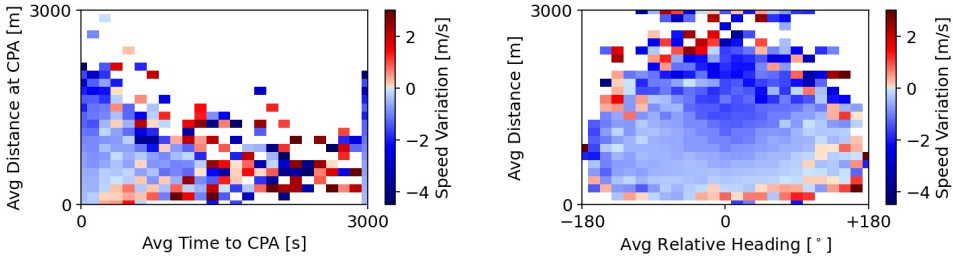
6

(a) RL method controlling heading, speed, and altitude variation.
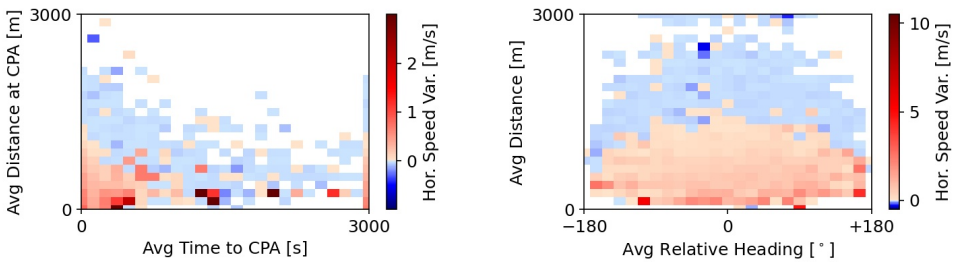
(b) MVP method controlling heading, speed, and altitude variation.

Figure 6.8: Speed variation by the RL and the MVP methods controlling heading, speed, and altitude variation.

spread of traffic is likely to contribute to the reduction in conflicts and LoSs (see Figure 6.3). However, it is likely to negatively affect flight path and time. This will be covered in more detail in the testing results in Section 6.5.2.

(a) RL method controlling heading, speed, and altitude variation.

(b) MVP method controlling heading, speed, and altitude variation.
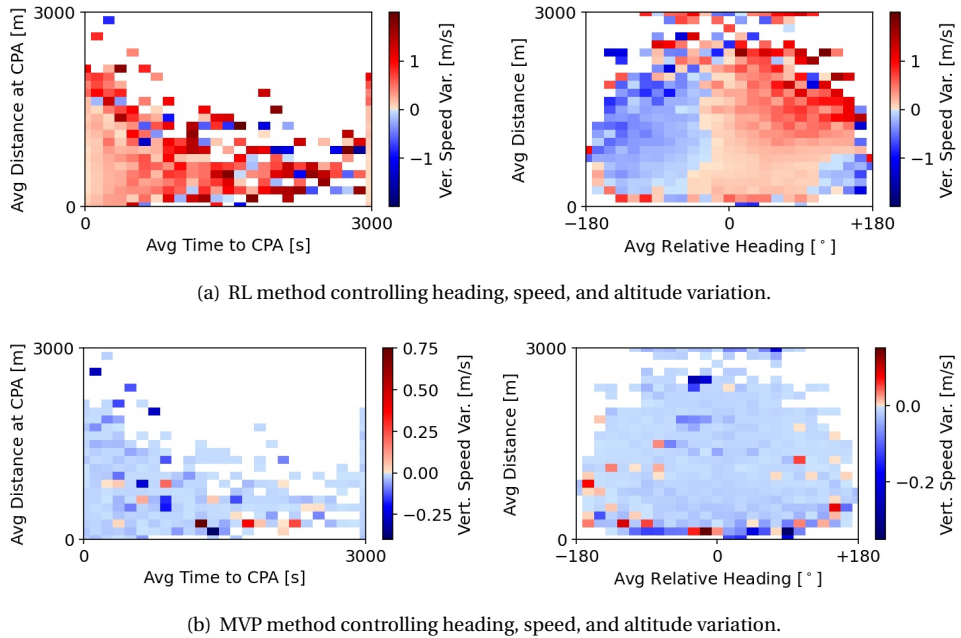
Figure 6.9: Vertical speed by the RL and MVP methods controlling heading, speed, and altitude variation.

6.5.2. TESTING OF THE RL AGENT FOR CONFLICT RESOLUTION

This section presents the testing phase of the RL method. The latter is tested with different traffic scenarios at a low, medium, and high traffic density. For each traffic density, three repetitions are run with three different route scenarios. The results of the RL method, related to safety, stability, and efficiency, are directly compared to those obtained when running the same traffic scenarios the MVP method.

SAFETY ANALYSIS

Figure 6.10 displays the average total number of pairwise conflicts. At low and medium traffic densities, the difference between the total number of conflicts with the RL and MVP methods is small, similar to the training results (see Figure 6.3(b)). With speed and heading deviation, there is a small difference in high traffic density, with the RL method achieving slightly fewer conflicts. Nevertheless, this is not a considerable difference. However, when altitude is also controlled, the RL method is capable of a great reduction in conflicts compared to MVP. This is probably a result of the larger altitude variations, as seen previously in Figure 6.9. These variations move aircraft out of the main traffic layer both by climbing and descending, thus reducing the likelihood of secondary conflicts.
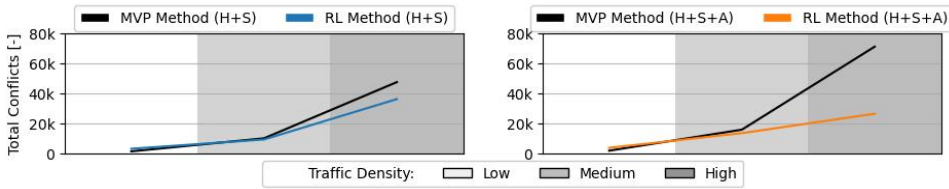
Figure 6.10: Average mean total number of pairwise conflicts during testing of the RL agent.

Figure 6.11 displays the average time in conflict per aircraft. An aircraft enters 'conflict mode' when it adopts a new state computed by the CR method. The aircraft will exit this mode once it is detected that it is past the previously calculated time to CPA (and no other conflict is expected between now and the look-ahead time). At this point, the aircraft will redirect its course to the next waypoint. The time to recovery is not included in the total time in conflict. Similarly to the average total number of conflicts (see Figure 6.10), the total time in conflict is similar in both methods when these control heading and speed variation. In the H + S + A scenarios, with the RL method, aircraft spend less time in conflict at all traffic densities. This is likely the result of the same choices in altitude variation that lead to a reduction in the total number of conflicts.
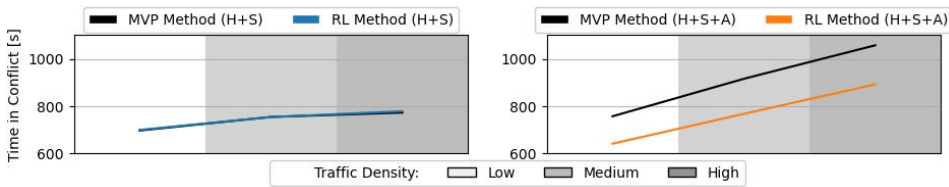
Figure 6.11: Average time in conflict per aircraft during testing of the RL agent.

Figure 6.12 displays the total number of LoS. This is the paramount safety factor that the RL method aims to reduce. The RL method achieves a similar performance level at the medium traffic density independent of the action formulation, and an even stronger reduction in the number of LoSs at low traffic density when the methods control heading and speed variation. This is probably a result of the 'defensive' posture adopted by the RL method with distant intruders and close aircraft, as seen in Figure 6.5(a). Head-on conflicts at relatively large distances, and nearby aircraft that can create imminent conflicts with only a small change in their velocity vector, can both be dangerous. However, this behaviour is not as efficient at higher traffic densities. Here, the CR methods must be more selective in the aircraft to defend against, as considering too many aircraft may result in a solution that does not fully resolve any conflict.

As hypothesised, the performance of the RL method deteriorated at a higher traffic density than that in which it was trained. This may be a result of the small number of surrounding aircraft considered in the state formulation. With a higher number of intruders per conflict geometry, the ownship will be 'blind' to part of the intruders. Limitation of the information in the state formulation limited the capability of the RL method to

generalise the learnt behaviour to conflict geometries with more intruders. Furthermore, different traffic densities may require different resolution strategies: in this case, an RL method must be trained at least at the traffic density at which it will be used.
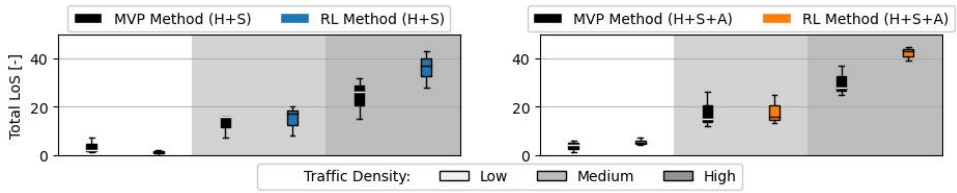
Figure 6.12: Total number of LoS during testing of the RL agent.

Figure 6.13 displays the average LoS severity. With heading and speed control, at low traffic densities, the RL method reduces the LoS severity on top of the total number of LoSs (see Figure 6.12). In all other situations, the average LoS severity is very similar. With all methods, there is a slight increase in the LoS severity as the traffic densities increases.
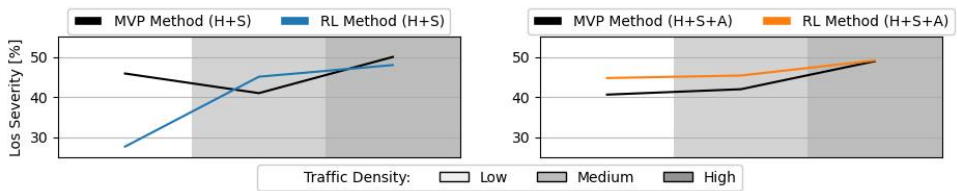
Figure 6.13: Average LoS severity during testing of the RL agent.

STABILITY ANALYSIS

Figure 6.14 shows the average DEP value during the testing of the RL method. A high DEP symbolises a method that tends to create a larger number of secondary conflicts, resulting from large deviations with conflict resolution manoeuvres. With heading and speed control, the increase in the number of conflicts is negligible (as also seen in Figure 6.10). With additional altitude control, the RL method is better at preventing secondary conflicts than the MVP method, which is also in line with the average total number of conflicts.
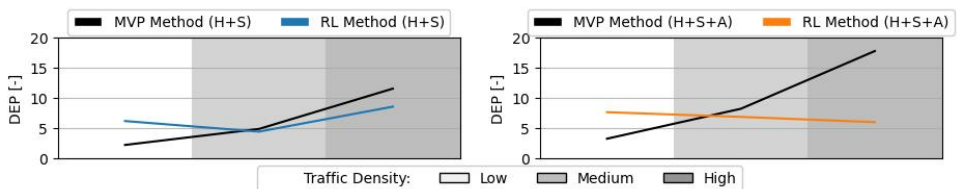
Figure 6.14: Average domino effect parameter (DEP) during testing of the RL agent.

EFFICIENCY ANALYSIS

Figures 6.15 and 6.16 show the average 2D and 3D flight path lengths per aircraft during testing of the RL method, respectively. There is only a noticeable difference between the RL and MVP methods, when altitude is also controlled. In this case, the RL method is able to reduce the increase in flight path resulting from tactical conflict manoeuvres, in comparison to the MVP method. Previously in Section 6.5.1, it was mentioned that the large vertical deviations performed by the RL method could have a negative effect on efficiency, by considerably increasing flight path length. However, it appears as though the decrease in total time in conflict (see Figure 6.11) may have counterbalanced these non-efficient vertical manoeuvres. As aircraft spend more time travelling towards the target and less time following a deconflicting state, the increase to the flight path is reduced.
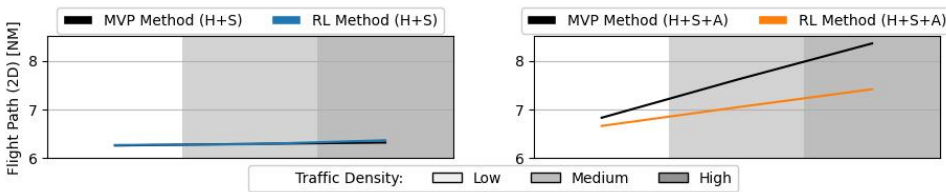
Figure 6.15: Average 2D flight path during testing of the RL agent.
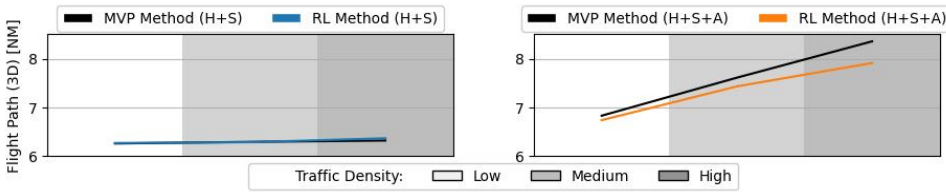
Figure 6.16: Average 3D flight path during testing of the RL agent.

Figure 6.17 shows the average flight time per aircraft during testing of the RL method. The differences between the RL and MVP methods in efficiency with heading and speed control are again negligible. When the methods also performed altitude variation, there is a slight increase in the average flight time per aircraft. Since the method achieves a shorter flight path (see Figures 6.15 and 6.16), the RL methods adopt on average lower horizontal speeds. This is in line with the speed variation decisions performed by the RL method during training (see Figure 6.9(a)), where, contrary to the MVP, it often opts for a decrease in horizontal speed.

6.6. DISCUSSION

This work investigated whether RL methods can mitigate the undesirable global patterns emerging from successive conflict resolution manoeuvres in multi-actor conflicts, and improve safety compared to current geometric CR methods. The results obtained show that, at low traffic densities, RL methods can match, and sometimes even surpass, the performance of these geometric algorithms. The RL method learnt that the high
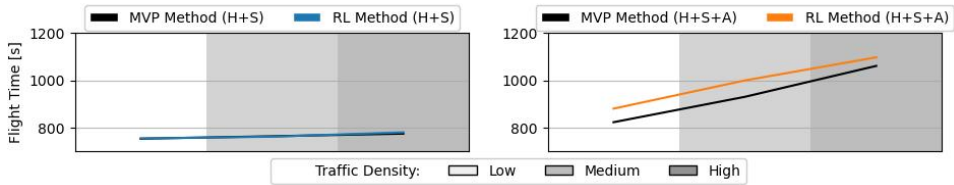
Figure 6.17: Average flight time during testing of the RL agent.

number of conflicting aircraft force aircraft to frequently change their velocity vector. Defending in advance against nearby aircraft, even when not in conflict, has benefits given that a sudden change in the velocity vector can create imminent conflicts. Furthermore, defending in advance against head-on conflicts, even when still at a large distance, may be crucial to prevent future LoSs.

However, performance deteriorates when these methods are exposed to traffic densities higher than those in which they were trained. The actions learnt from a set of conflict geometries, at smaller traffic densities, do not generalise well to conflict geometries with a higher number of intruders. This is a weakness of the approach. The benefit of RL is the potential to make generalisations about emerging patterns. If an RL method is only capable of creating a limited set of rules, its performance will be similar to that of geometric CR methods. Such may be a result of the limited information and variety of training scenarios provided to the RL method. Within the state representation, we considered a limit number of aircraft, which was found to be efficient within specific training scenarios while still resulting in practicable convergence times. However, in conflict geometries with a higher number of intruders, the RL method may not have enough information to perform successful resolution manoeuvres. Thus, limiting state-action combinations may have a negative impact on the ability of RL methods to generalise solutions to different operational environments. However, this limitation is often required to achieve convergence.

Naturally, it may also be that the RL method requires a longer training time, or even a more generalised training environment, than was provided in this work. Further testing is needed, with more information given to the method and more extensive training scenarios. However, it should also be taken into account that doing so greatly increases the necessary training time of the method. Notwithstanding, the actions performed by the RL method in this study can provide guidelines on how to improve the performance of current geometric CR methods. This topic, as well as further analysis of the actions performed by the RL method, is addressed in the following sections.

6.6.1. ACTIONS PERFORMED BY THE REINFORCEMENT LEARNING METHOD
This work proved that an RL method can successfully resolve conflicts and prevent losses in minimum separation. Furthermore, having a global reward improved action coordination between aircraft. The actions of the RL method are the result of the combination of multiple factors. However, the results show that the current distance to aircraft and relative heading have a greater impact on the method's decisions than, for example, the distance at the closest point of approach. This is possibly because the method was

able to establish that small distances between aircraft result in negative rewards. Additionally, the method learnt that head-on conflicts are especially hard to resolve in the short-term. Finally, by observing all actions performed during training, it is clear that the RL method has a preference to always turn one direction when resolving conflicts. Although this is a very simple coordination rule, it is efficient in pairwise conflicts. However, it is likely that a better coordination rule is necessary when more aircraft are involved in a conflict situation. This might explain why the performance of the RL method deteriorated at higher traffic densities.

Moreover, with three degrees of freedom (i.e., heading, speed, and altitude), the RL method achieved fewer conflicts and LoSs than with 2 degrees of freedom (i.e., heading, and speed). This is unexpected, since larger actions formulations are often negative for RL methods, as it increases the possible combinations of state-action that the method must learn and adapt to. However, the improvement in resolution manoeuvres is intrinsically related to best practises for conflict resolution. With more degrees of freedom, the method learnt to perform smaller deviations on each. This reduces the amount of airspace that the ownship occupies during a deconflicting manoeuvre, thus also reducing the likelihood of colliding with other aircraft. Additionally, given the characteristics of the operational environment in this work, where all aircraft travel in one layer, fast singular vertical manoeuvres can be very efficient in resolving conflicts. The RL method took advantage of this fact. It is likely that the RL method would find different optimal options in an environment with different characteristics, e.g., a layered airspace.

6.6.2. RULES FOR CONFLICT RESOLUTION

Reinforcement learning applications are often a 'black-box'; the reasons for their choices are often not clear or predictable. However, if these are to be certified as safety critical systems, we must find ways to make their behaviour interpretable and traceable. Many researchers also defend the idea that we should look at RL methods as a source of best practises. These practises can then be implemented in human-built CR methods, whose actions are predefined and can be trusted upon. With this implementation in mind, and considering the behaviour of the RL method in this work, the following rules for improving geometric CR methods can be derived:

- The RL method often prioritised the current distance to aircraft over the distance at CPA. Nearby aircraft can potentially change course and immediately turn into an (almost) impossible to resolve conflict. Geometric CR methods often simply look at the distance at CPA, and the time to CPA. Nearby aircraft, even if not in conflict, should be defended against.
- The RL method defends against head-on conflicts in advance. Based on the results, the RL method also performs strong state variations when intruders are far away, but (near-)head-on. Short-term head-on conflicts can only be resolved with coordinated sharp heading turns. When the intruder is farther away, smaller heading deviations are needed to resolve the conflict. Geometric CR methods initiate deconflicting actions for all conflict situations in the same manner, i.e, when these are within a pre-defined look-ahead time. However, the decision of when a resolution manoeuvre is initiated should be made with regard to the relative geometry between the ownship and the intruder. In practise, different rules for look-ahead

times could be implemented per relative heading.

In summary, the RL method adopts a more preventive/cautious position towards nearby aircraft, even when not in conflict, and defends in advance against severe conflict geometries. However, given the results obtained, it may be that the previous rules are more efficient at low traffic densities where there is enough space. The previous measures resulted in the method defending against more aircraft, per deconflicting action, than a typical geometric CR method would have. At higher traffic densities, it may be that this behaviour results in too high a number of conflicts that saturates the solution space.

6.6.3. FUTURE WORK

This work should be extended to different operational environments. Comparison of the optimal manoeuvres performed by RL methods trained in different environments provides valuable information. First, it helps identify potential risks in each type of operation. Second, it helps identify an optimal global usage of reinforcement learning to improve safety in aviation. In particular, the following is suggested for future work:

- Different traffic densities may require different resolution strategies, as was also hypothesised in the Metropolis project [13]. In this case, the RL method must learn different responses per complexity of emergent behaviour resulting from increasing traffic densities. RL methods should be tested at different traffic densities, and their actions compared, before they can be implemented in a real-world scenario.
- With only heading and speed control, the RL method employed large heading deviation which led to a large number of secondary conflicts. This may be because all losses of minimum separation are valued the same. The method may benefit from ensuring large distances between aircraft to avoid negative rewards. As a possible solution, less weight could be given to minor, less severe LoSs. Such a scenario could potentially lead the method to adopt smaller state changes, running the risk of scraping the protected zone of intruders over large deconflicting manoeuvres that would place the ownship in the direct path of other aircraft.

6.7. CONCLUSIONS

Reinforcement learning (RL) has the potential to adapt to the detrimental emergent behaviour from multi-actor conflicts, and knock-on effects from successive conflict resolution manoeuvres, which occur as traffic densities increase. An RL method can potentially develop a large set of rules, adapted to different conflict geometries, from knowledge of the environment captured during training. Adding to the success of RL approaches in other scientific areas, this chapter has shown that RL methods can be used to guarantee minimum separation between all aircraft in an unmanned aviation environment.

The RL method developed herein successfully resolved conflicts and reduced the number of losses in minimum separation (LoSs). Moreover, it matched, and at lower traffic densities even surpassed, the performance of a state-of-the-art geometric conflict resolution (CR) algorithm. The advantage of an RL method is the ability to learn safe procedures, beyond the limitations of a fixed set of rules, as implemented with geometric CR methods. However, there are still some weaknesses in this approach. The performance of the RL method deteriorated when exposed to traffic densities higher than those in

which it was trained. The RL method receives limited information from the environment, in order to limit the number of possible state-action combinations. This probably also limited its ability to generalise its solutions to different operational environments.

Additionally, RL methods can inspire the creation of additional rules/guidelines to improve the performance of geometric CR algorithms. How RL methods resolve conflicts in different environments can help identify the risks in every type of operation, as well as possible solutions. In this specific work, the RL method showed that: (1) adopting a more 'preventive' position towards nearby aircraft, even when not in conflict, and (2) defending in advance against difficult conflict geometries, help prevent LoSs at low traffic densities.

The next steps will focus on further exploring state and action formulations in order to increase the efficacy of the RL method at traffic densities higher than the one in which it is trained. Furthermore, related studies should extend the training and testing of conflict resolution RL methods to different operational environments. Different methods, with different amounts of information and control, should be applied and compared. Such will make it possible to evaluate the influence of these parameters on the method's ability to generalise and identify conflict geometry-specific solutions. However, these decisions must be evaluated together with the consequent necessary training time, and resources, for the method to converge towards optimal actions.

6

7

IMPROVING CONFLICT RESOLUTION MANOEUVRES WITH REINFORCEMENT LEARNING

The results of the Chapter 6 show that reinforcement learning (RL) methods can be used to improve the efficacy of current state-of-the-art distributed CR algorithms. These algorithms often resort to pre-defined, fixed values that are used to calculated a deconflicting manoeuvre for every conflict geometry. However, at thigh traffic densities each aircraft will face a multitude of conflict geometries for which the same values might not be optimal.

Section 7.3 defines an RL method responsible for defining the values that an CR algorithm uses for generating a deconflicting manoeuvre for every conflict geometry. Different action formulations are tested to better understand the potential of RL methods. The RL method is tested with different traffic densities to evaluate its performance under complex multi-actor conflict geometries.

Cover-to-cover readers may chose to skip Sections 7.3 and 7.4.5, which describes the theoretical background of a Soft Actor-Critic (SAC) and of the Modified Voltage Potential methods (MVP), respectively. These are very similar to their counterparts in previous Chapter 6.

This chapter is based on the following publications:

1. M. Ribeiro, J. Ellerbroek, and J. Hoekstra, Improving Algorithm Conflict Resolution Manoeuvres With Reinforcement Learning, Aerospace, 2022
2. M. Ribeiro, J. Ellerbroek, and J. Hoekstra, Determining Optimal Conflict Avoidance Manoeuvres At High Densities With Reinforcement Learning, 10th SESAR Innovation Days (2020)

ABSTRACT

Future high traffic densities with drone operations are expected to exceed the number of aircraft that current air traffic control procedures can control simultaneously. Despite extensive research on geometric CR methods, at higher densities, their performance is hindered by the unpredictable emergent behaviour from surrounding aircraft. In response, research has shifted its attention to creating automated tools capable of generating conflict resolution (CR) actions adapted to the environment and not limited by man-made rules. Several works employing reinforcement learning (RL) methods for conflict resolution have been published recently. Although proving that they have potential, at their current development, the results of the practical implementation of these methods do not reach their expected theoretical performance. Consequently, RL applications cannot yet match the efficacy of geometric CR methods. Nevertheless, these applications can improve the set of rules that geometrical CR methods use to generate a CR manoeuvre. This work employs an RL method responsible for deciding the parameters that a geometric CR method uses to generate the CR manoeuvre for each conflict situation. The results show that this hybrid approach, combining the strengths of geometric CR and RL methods, reduces the total number of losses of minimum separation. Additionally, the large range of different optimal solutions found by the RL method shows that the rules of geometric CR method must be expanded, catering for different conflict geometries.

7.1. INTRODUCTION

Recent studies estimate that as many as 400,000 drones will be providing services in the European airspace by 2050 [197]. Several geometric CR methods have been developed to implement the tactical separation function for these operations. These methods are capable of guaranteeing separation between aircraft without human intervention. Nevertheless, at the traffic densities envisioned for drone operations, these methods start to suffer from destabilising emergent patterns. Multiactor conflicts and knock-on effects can lead to global patterns that cannot be predicted based on limited man-made rules. Researchers have started employing reinforcement learning (RL) techniques for conflict resolution, which can defend against this multiagent emergent behaviour [232]. RL can train directly in the environment and adapt directly to the interaction between the agents. In previous work [233] (Chapter 6 of this thesis), we compared an RL approach with the geometric state-of-the-art Modified Voltage Potential (MVP) method [15]. The results showed that the employed RL method was not as efficient as the MVP method in preventing losses of minimum separation (LoSs), at higher traffic densities. Nevertheless, at lower traffic densities, the RL method defended in advance against severe conflicts. The MVP method was not able to resolve these in time, as it initiated the conflict resolution manoeuvre later. This suggests that RL approaches can improve the decisions taken by current geometric CR methods. This hypothesis is explored in this work.

    Geometric CR algorithms are typically implemented using predefined, fixed rules (e.g., predefined look-ahead time, and in which direction to move out of the conflict) that are used for all conflict geometries. However, at high traffic densities, each aircraft will face a multitude of conflict geometries for which the same values might not be optimal. For example, in some situations it may be useful to have a larger look-ahead time to defend

against future conflicts in advance. In other cases, prioritising short-term conflicts may be necessary to avoid false positives from uncertainties accumulating over time. Similarly, fast climbing actions can prevent conflicts with other aircraft when these occupy a single altitude. Additionally, a speed-only state change may be sufficient to resolve the conflict while preventing the ownship from occupying a larger portion of the airspace. All these decisions are dependent of the conflict geometry. Nevertheless, the number of potential geometry variations are too large for experts to create enough rules to cover them all. However, RL approaches are known for their ability to find optimised solutions in systems with a large number of possible states.

We propose a Soft Actor–Critic (SAC) model, as created by UC Berkeley [229], that trains within the airspace environment to find the optimal values for the calculation of a conflict resolution manoeuvre. Specifically, in each conflict situation, the RL method defines the look-ahead time (0 s–600 s) and how many degrees of freedom to employ (i.e, heading, speed, or altitude variation) that the MVP method then uses to generate the CR manoeuvre. The RL method uses the local observations of the aircraft to define these values. This hybrid RL + MVP approach is used for all aircraft involved in the conflict. Finally, experiments are conducted with the open-source, multiagent ATC simulation tool BlueSky [25]. The source code and scenarios files are available online [234].

Section 7.2 introduces the current state-of-the-art research employing RL to improve separation assurance between aircraft. Several works are described, as well as how this chapter adds to this body of work. Section 7.3 outlines the RL algorithm used in this work, as well as the state, action, and reward formulations used by the RL method. Section 7.4 describes the simulation environments of the experiments performed, as well as the conflict detection and resolution (CD&R) methods employed. The hypotheses initially set for the behaviour of hybrid RL + MVP approach are specified in Section 7.5. Section 7.6 displays the results of the training and the testing phases of the RL method. Finally, Sections 7.7 and 7.8 present the discussion and conclusion, respectively.

7.2. RELATED WORK

This chapter adds to the body of work that uses RL to improve CD&R between aircraft. Recently, an overview of the most recent studies in this area was published [232], showing that a variety of different RL approaches had been implemented. In previous work, Soltani [235] used a mixed-integer linear programming (MILP) model to include conflict avoidance in the formulation of taxiing operations planning. Li [236] developed a deep RL method to compute corrections for an existing collision avoidance approach to account for dense airspace. Henry [237] employed Q-Learning to find conflict-free sequencing and merging actions. Pham (2019) [227] used the deep deterministic policy gradient (DDPG) method [163] for conflict resolution in the presence of surrounding traffic and uncertainty. Isufah (2021) [228] proposed a multiagent RL (MA-RL) conflict resolution method suitable for promoting cooperation between aircraft. Brittain (2019) [238] defined an MA-RL method to provide speed advisories to aircraft to avoid conflict in high-density, en route airspace sector. Groot [239] developed an RL method capable of decreasing the number of intrusions during vertical movements. Dalmau [240] used message-passing neural networks (MPNN) to allow aircraft to exchange information through a communication protocol before proposing a joint action that promoted flight efficiency and penalised

conflicts. Isufah (2002) implemented an algorithm based on graph neural networks where cooperative agents could communicate to jointly generate a resolution manoeuvre [241]. Brittain (2022) proposed a scalable autonomous separation assurance MA-RL framework for high-density en route airspace sectors with heterogeneous aircraft objectives [242]. Panoutsakopoulos employed an RL agent for separation assurance of an aircraft with sparse terminal rewards [243]. Finally, Pham (2022) trained an RL algorithm inspired from Q-learning and DDPG algorithms that can serve as an advisory tool [244].

All the aforementioned works show that RL has the potential to improve the set of rules used for conflict resolution. RL trains directly in the environment, and can thus adapt to the emergent behaviour resulting from successive avoidance manoeuvres in multiactor conflict situations. However, RL also has its drawbacks, such as nonconvergence, a high dependence on initial conditions, and long training times. We consider that having an RL method that is completely responsible for the definition of avoidance manoeuvres is (practically) infeasible, as it would have severe issues converging to the desirable behaviour. In this chapter, we hypothesise that the best usage of RL is, instead, to work towards improving the current performance of state-of-the-art CR methods. We develop a hybrid approach, combining the strengths of geometric methods and learning methods, and hopefully mitigating the drawbacks of each of the individual methods. In this approach, rewards are scaled by the efficacy of the conflict resolution manoeuvres, and the starting point is the current efficacy of the CR method.

7.3. IMPROVING CONFLICT RESOLUTION WITH RL

This chapter employs RL to define the values to input into the CR algorithm responsible for calculating CR manoeuvres. RL was chosen due to its ability to understand and compute a full sequence of actions. A consequence of resolving conflicts is often the creation of secondary conflicts when aircraft move into the path of nearby aircraft while changing their state to resolve a conflict. This often leads to consequent CR manoeuvres to resolve these secondary conflicts. Additionally, knock-on effects of intruders avoiding each other may result in unforeseen trajectory changes. The latter increases uncertainty regarding intruders' future movements, decreasing the efficacy of CR manoeuvres. RL techniques are often capable of identifying these emerging patterns through direct training in the environment.

Several studies have used other tools such as supervised learning, where classification and regression can be used to estimate the values to be used to aid CR [151, 245, 246]. These works have shown that these methods can also lead to favourable results. However, these methods do not train directly in the environment, instead resorting to prior knowledge and repeating this knowledge on a large scale. Nevertheless, we assume no previous knowledge and were mainly interested in the knowledge that RL can learn by adapting to the emergent behaviour experienced in the environment.

Section 7.3.1 specifies the theoretical background of the RL algorithm employed in this work as well as the defined hyperparameters. Next, Section 7.3.2 describes the RL agent employed and how it interacts with the environment. Section 7.3.3 details the information that the RL agent receives from the environment, and Section 7.3.4 the actions performed by the agent. Finally, Section 7.3.5 presents the reward formulation used to evolve the RL agent towards finding optimal actions.

7.3.1. LEARNING ALGORITHM

An RL method consists of an agent interacting with an environment in discrete time steps. At each time step, the agent receives the current state of the environment and performs an action accordingly, for which it receives a reward. An agent's behaviour is defined by a policy, π, which maps states to actions. The goal is to learn a policy which maximizes the reward. Many RL algorithms have been researched in terms of defining the expected reward following an action.

This chapter uses the Soft Actor–Critic (SAC) method as defined in [229]. SAC is an off-policy actor–critic deep RL algorithm. It employs two different deep neural networks for approximating action-value functions and state-value functions. The actor maps the current state based on the action it estimates to be optimal, while the critic evaluates the action by calculating the value function. The main feature of SAC is its maximum entropy objective, which has practical advantages. The agent is encouraged to explore more widely, which increases the chances of finding optimal behaviour. Second, when the agent finds multiple options for a near-optimal behaviour, the policy commits equal probability to these actions. Studies have found that this improves learning speed [247].

Table 7.1 presents the hyperparameters employed in this work. We resorted to two-hidden-layer neural networks with 256 neurons in each layer. Both layers used the rectified linear unit (ReLU) activation function.

Table 7.1: Hyperparameters of the employed RL method used in this work.

Parameter	Value
TAU	0.005
Learning rate actor (LRA)	0.0001
Learning rate critic (LRC)	0.001
EPSILON	0.1
GAMMA	0.99
Buffer size	1 M
Minibatch size	256
#Hidden layer-neural networks	2
#Neurons	256 in each layer
Activation functions	Rectified linear unit (*ReLU*) in the hidden layers, *tanh* in the last layer

7.3.2. AGENT

We employ an RL agent responsible for setting the parameters for the calculation of a CR manoeuvre by a geometric CR method. At each time step, the CR method outputs a new deconflicting state for all aircraft in conflict. The new state aims at preventing LoSs with the necessary minimum path deviation. Every time the CR method computes an avoidance manoeuvre, it uses the following parameters decided by the RL method:

1. The look-ahead value (between 0 and 600 s).
2. A selection of which state elements vary (i.e., heading, speed, and/or altitude variation).

The previous values are used by the CR method to generate the CR manoeuvre. The latter is performed by the ownship in order to resolve the conflict and avoid an LoS. Note that

all aircraft involved in a CR perform a deconflicting manoeuvre computed by the hybrid RL + CR method.

A single RL agent is considered in this work. Note that several works previously mentioned in Section 7.2 have considered the application of MA-RL instead for CR. Looking at the existent body of work [232], there is no clear preference for either MA-RL or single RL in current studies. The selection of a single agent or multiagent mainly depends on the problem being tackled. Although, theoretically, MA-RL is expected to better handle the nonstationarity of having multiple agents evolving together, a single RL is used in this work for the following reasons:

- The single agent can be used on any aircraft; it does not limit the number of aircraft. Instead, MA-RL represents the observations of all agents in its state formulation. As the state array has a fixed size, it can only be used with a fixed number of aircraft.
- Through practical experiments, Zhang [248] concluded that MA learning was weaker than a single agent under the same amount of training. It is thus necessary to balance the optimisation objectives of multiagents in an appropriate way. It is reasonable to expect that MA-RL may require more training due to the increasing state and action formulation, as well as the need to correctly identify which actions had more impact on the reward.

Figure 7.1 is a high level representation of how the RL agent interacts with the CR algorithm. As per Figure 7.1, the state input is transformed into the action output through each layer of the neural network. The compositing of the state and action arrays is described in Sections 7.3.3 and 7.3.4, respectively. The variables of the state array have continuous values within the limits presented in Table 7.2. The action formulation contains both continuous and discrete values as shown in Table 7.3.
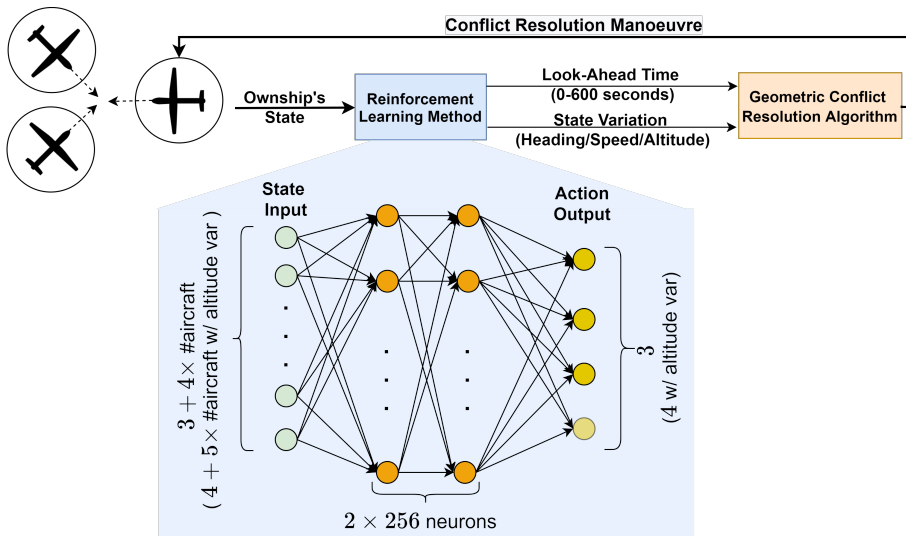
7

Figure 7.1: High level of the hybrid conflict resolution system implemented in this work. Based on the ownship's current state and closest surrounding aircraft, the RL method makes a decision on look ahead time and in which way the state of the ownship will vary. The geometric CR algorithm then uses these values to generate the conflict resolution manoeuvre.

7.3.3. State

The RL method must receive the necessary data for the RL agent to successfully decide which values to input into the CR algorithm for the generation of an effective CR manoeuvre. We took inspiration from the same data that typically distributed, geometric CR methods have access to so as to create a fair comparison. These data included the current state of the ownship aircraft, and the distance, relative heading, and relative altitude of the closest surrounding aircraft. Furthermore, the distance at the closest point of approach (CPA) and the time to the CPA were also considered as this is also information that the CR method has access to. When RL controls altitude variation, on top of heading and speed, it also receives information on the ownship's current altitude and its relative altitude to the closest intruders. Table 7.2 defines the complete state information received by the RL method. Note that the RL method was tested with 2 different implementations of the CR method: (1) the CR method varies heading and speed; (2) the CR method varies heading, speed, and altitude. Thus, the optimal look-ahead times could be related to the level of control that the geometric CR algorithm had over the ownship.

In the state representation, we considered the closest 4 surrounding aircraft. This decision was a balance between giving enough information on the environment, while keeping the state formulation to a minimum size. The size of the problem's solution grows exponentially with the number of possible states permutations. Thus, this must be limited to guarantee that the RL method trains within an unacceptable amount of time. The 4 closest aircraft (in distance) were chosen in order of their proximity, independently of them being in conflict or not. The reason for considering all aircraft was to allow the RL method to make its decision based not only on the conflicting aircraft but also on nearby nonconflicting aircraft, which could create severe conflicts if they modified their state in the direction of the ownship.

Table 7.2: State formulation of the RL method.

Dimension	Element	Limits
1	Current heading	$-180°$–$180°$
1	Relative bearing to next waypoint	$-180°$–$180°$
1	Current speed	$0\,\text{m/s}$–$18\,\text{m/s}$
#surrounding aircraft	Current distance to #surrounding aircraft	$0\,\text{m}$–$3000\,\text{m}$
#surrounding aircraft	Distance at CPA with #surrounding aircraft	$0\,\text{m}$–$3000\,\text{m}$
#surrounding aircraft	Time to CPA with #surrounding aircraft	$0\,\text{s}$–$600\,\text{s}$
#surrounding aircraft	Relative heading to #surrounding aircraft	$-180°$–$180°$
Only when the geometric CR method can also perform altitude variation:		
1	Current altitude	$0\,\text{ft}$–$100\,\text{ft}$
#surrounding aircraft	Relative altitude to #surrounding aircraft	$0\,\text{ft}$–$100\,\text{ft}$

7.3.4. Action

The RL agent determines the action to be performed for the current state. As previously displayed in Figure 7.1, the incoming state values are transformed through each layer of the neural network, in accordance to the neurons' weights and the activation function in each layer. The output of the final layer must be turned into values that can be used to define the elements of the state of the aircraft that the RL agent controls. All actions were

computed using a *tanh* activation function; the RL method thus output values between
−1 and +1. Table 7.3 shows how these values were then translated to the values to be used
by the geometric CR method.

Table 7.3: Action formulation of the RL method.

Dimension	Action	Limits	Units
1	Look-ahead time (for CR only)	[−1, + 1] transforms to [0, 600]	Seconds
1	Heading variation	Yes if ≥ 0, no otherwise	Yes/no
1	Speed variation	Yes if ≥ 0, no otherwise	Yes/no
Only when the geometric CR method can also perform altitude variation:			
1	Vertical speed variation	Yes if ≥ 0, no otherwise	Yes/no

The RL method was tested with the geometric CR method controlling (1) heading and
speed variations, and (2) heading, speed, and altitude variations. In both cases, the RL
method defined the look-ahead value to be used. This was a continuous action. Note
that this was the look-ahead time used for resolving conflicts. The RL method received
information regarding the aircraft surrounding the ownship through the state input.
This was how it 'detected' conflicts. Then, it decided on the look-ahead time used for CR,
as displayed in Figure 7.2. This meant that the method decided which conflicts to consider
in the next avoidance manoeuvre. The method could opt, for example, for prioritising
closer conflicts.
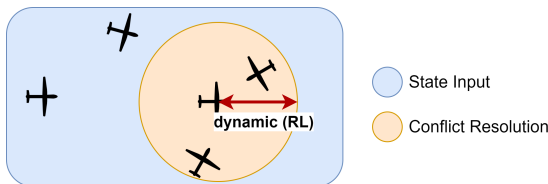
Figure 7.2: The RL method receives information on the aircraft surrounding the ownship through the state input,
and outputs a look-ahead value to be used for conflict resolution.

Furthermore, depending on the degrees of freedom that the geometric CR algorithm
controlled, the RL method defined how the state of the ownship could vary. This selection
was a discrete action. For heading, speed, and altitude variation, if the respective value
on the state array was higher than 0, the resolution by the geometric CR method included
a variation within that degree.

Finally, note that these options took a continuous value (from the RL method's output)
and turned it into a discretised option (≥0 ∨ <0). This could hinder the ability of the
RL method to properly understand how its continuous values were used, and limit the
efficacy of training. Nevertheless, this was preferred in order to (1) have one RL method re-
sponsible for all actions so that the effect of their combination could be directly evaluated
by the method, and (2) to have continuous values for the look-ahead time, which allowed
the RL method to directly include or disregard specific aircraft in the generation of the
avoidance manoeuvre. Future work will explore whether there is an increase in efficiency
by having 2 different RL methods. The first may produce continuous actions to define the

look-ahead time. The second receives this look-ahead value and outputs discrete actions for the selection/deselection of heading, speed, and altitude variation. Nevertheless, here the first RL method was not aware of the decisions of the second method.

7.3.5. REWARD
The RL method was rewarded at the time step following the one where it set the parameters for the CR manoeuvre calculation for an aircraft. The reward for each state (s_t) was based on the number of LoSs, as this was the paramount safety objective:

$$R(s_t) = \begin{cases} -1 & \text{Loss of Separation occurs} \\ 0 & \text{otherwise} \end{cases} \tag{7.1}$$

The RL method was not aware of the MVP method, as the latter only contributed indirectly to the reward by attempting to prevent LoSs. Note that efficiency elements could also be added to the reward to decrease flight path and time. Nevertheless, the weight combination of the different elements must be carefully tuned to establish how one LoS compares to a large path deviation. At this phase of this exploratory work, we therefore opted for a simple reward formulation focusing on the main objective.

Furthermore, a reward can be local, when based on the part of the environment that the agent can directly observe, or global, when the reward is based on the global effect on the environment. There are advantages and disadvantages for both types of rewards. On the one hand, local reward may promote 'selfish' behaviour as each agent attempts to increase its own reward [249]. When solving a task in a distributed manner, if each agent tries to optimise its own reward, it may not lead to a globally optimal solution. On the other hand, the task of attributing a global, shared reward from the environment to the agents' individual actions is often nontrivial since the interactions between the agents and the environment can be highly complex (i.e., the 'credit assignment' problem).

In this work, the primary objective was to reduce the total number of LoSs in the airspace. Thus, a global reward was used, where each agent was rewarded based on the total number of LoSs suffered in the airspace. Note that there are technique to (partially) handle the credit assignment problem that were not used in this work. To the best of the authors' knowledge, these focus mainly in value function decomposition and reward shaping [250–253]. However, value function decomposition is hard to apply in off-policy training and potentially suffers from the risk of unbounded divergence [254]. In our case, we opted for having a simplified reward formulation (see Equation (7.1)). Thus, we did not make use of any strategy directed at reducing the credit assignment problem. As a result, however, it was expected that the training phase would take longer, as the RL agent had to explore the state and action spaces at length to understand how its actions influenced the global reward.

7.4. EXPERIMENT: IMPROVING ALGORITHM CONFLICT RESOLUTION MANOEUVRES WITH REINFORCEMENT LEARNING
This section defines the properties of the performed experiment. The latter aimed at using RL to define the values that a distributed, geometric CR method used to generate

CR manoeuvres. Note that it was divided into two main phases: training and testing. First, the hybrid RL + CR method was trained continuously with a predefined set of 16 traffic scenarios, at a medium traffic density. Each training scenario ran for 20 min. Afterwards, it was tested with unknown traffic scenarios at different traffic densities. Each testing scenario ran for 30 min. Each traffic density was run with 3 different scenarios, containing different routes. During testing, the performance of the hybrid method was directly compared to the performance of the distributed, geometric CR method, with baseline rules that have been commonly used in other works [162].

7.4.1. FLIGHT ROUTES

The experiment area was a square with an area of $144\,\mathrm{NM}^2$. Aircraft were created on the edges of this area, with a minimum spacing equal to the minimum separation distance, to avoid LoSs between spawn aircraft and aircraft arriving at their destination. Aircraft flew a linear route, all at the same altitude. Each linear path was built up of several waypoints. Aircraft were spawned at the same rate as they were deleted from the simulation, in order to maintain the desired traffic density. Naturally, when conflict resolution was applied to the environment, the instantaneous traffic density could be higher than expected as aircraft would take longer to finish their path due to path deviations to avoid LoSs. In order to prevent aircraft from being incorrectly deleted from the simulation when travelling through the edge of the experiment area, or when leaving the area to resolve a conflict, a larger area was set around the experiment area. An aircraft was removed from the simulation once it left this larger area.

7.4.2. APPARATUS AND AIRCRAFT MODEL

An airspace with unmanned traffic scenarios was built using the Open Air Traffic Simulator Bluesky [25]. The performance characteristics of the DJI Mavic Pro were used to simulate all vehicles. Here, speed and mass were retrieved from the manufacturer's data, and common conservative values were assumed for turn rate (max: $15°/\mathrm{s}$) and acceleration/breaking ($1.0\,\mathrm{kts/s}$).

7.4.3. MINIMUM SEPARATION

Minimum safe separation distance may vary based on the traffic density or the structure of the airspace. For unmanned aviation, a single, commonly-used value does not (yet) exist. In this experiment, we chose $50\,\mathrm{m}$ for horizontal separation and $50\,\mathrm{ft}$ for vertical separation.

7.4.4. CONFLICT DETECTION

The experiment will employ state-based conflict detection for all conditions. This assumes linear propagation of the current state of all aircraft involved. Using this approach, the time to CPA (in seconds) is calculated as:

$$t_{CPA} = -\frac{\vec{d}_{rel} \cdot \vec{v}_{rel}}{\vec{v}_{rel}}, \tag{7.2}$$

where $\vec{d}_{rel}$ is the cartesian distance vector between the involved aircraft (in meters), and $\vec{v}_{rel}$ the vector difference between the velocity vectors of the involved aircraft (in meters per second). The distance between aircraft at CPA (in meters) is calculated as:

$$d_{CPA} = \sqrt{\vec{d}_{rel}^2 - t_{CPA}^2 \cdot \vec{v}_{rel}^2}. \tag{7.3}$$

When the separation distance is calculated to be smaller than the specified minimal horizontal spacing, a time interval can be calculated in which separation will be lost if no action is taken:

$$t_{LoS} = t_{CPA} - \frac{\sqrt{R_{PZ}^2 - d_{CPA}^2}}{\vec{v}_{rel}}. \tag{7.4}$$

These equations will be used to detect conflicts, which are said to occur when $d_{CPA} < R_{PZ}$, and $t_{LoS} \leq t_{lookahead}$, where R_{PZ} is the radius of the protected zone, or the minimum horizontal separation, and $t_{lookahead}$ is the specified look-ahead time.

With the baseline CR method, a look-ahead time of 300 seconds is used. This value was selected as, empirically, it was found to be the most efficient common value for the 16 training scenarios within the simulated environment. This is a larger value than commonly used with unmanned aviation. However, smaller values are often considered in constrained airspace to reduce the amount of false conflicts past the borders of the environment [204]. Finally, this large look-ahead time should not be used in environments with uncertainty regarding intruders' current position and future path. Expanding the intruders trajectory far into the future can result in a great amount of false positive conflicts.

7.4.5. CONFLICT RESOLUTION
We use the distributed, geometric CR method MVP. The values used by MVP to calculate conflict avoidance manoeuvres are defined by the RL method. The principle of the geometric resolution of the MVP method, as defined by Hoekstra [2, 15], is displayed in Figure 7.3. MVP uses the predicted future positions of both ownship and intruder at CPA. These calculated positions 'repel' each other, towards a displacement of the predicted position at CPA. The avoidance vector is calculated as the vector starting at the future position of the ownship and ending at the edge of the intruder's protected zone, in the direction of the minimum distance vector. This displacement is thus the shortest way out of the intruder's protected zone. Dividing the avoidance vector by the time left to CPA, yields a new speed, which can be added to the ownship's current speed vector resulting in a new advised speed vector. From the latter, a new advised heading and speed can be retrieved. The same principle is used on the vertical situation, resulting in an advised vertical speed. In a multi-conflict situation, the final avoidance vector is determined by summing the repulsive forces with all intruders. As it is assumed that both aircraft in a conflict will take (opposite) measures to evade the other, MVP is implicitly coordinated.

7.4.6. INDEPENDENT VARIABLES
First, two different implementations of the hybrid RL+MVP method are trained with different action formulations. During testing, different traffic densities are introduced to
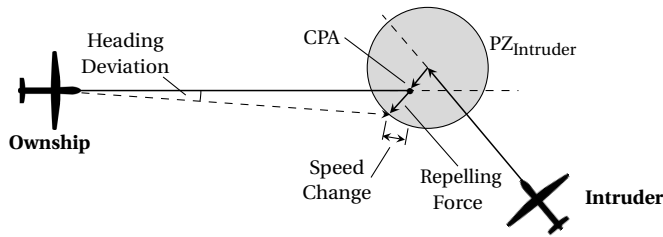
Figure 7.3: Modified Voltage Potential (MVP) geometric resolution. Adapted from [15].

analyse how the RL method performs at traffic densities it was not trained in. Finally, the efficacy of the hybrid RL+MVP is directly compared to that of the baseline MVP method. More detail is given below.

ACTION FORMULATION

Different action formulations were employed (1) when the CR method only performed heading and speed variations to resolve conflicts and (2) when the geometric CR method used heading, speed, and altitude variation. These allow direct analyses of how the decisions of the RL method change depending on the control it has over the state of all aircraft involved in a conflict situation.

TRAFFIC DENSITY

Traffic density varies from low to high according to Table 7.4. The RL agent is trained at a medium traffic density, and is then tested with low, medium, and high traffic densities. Thus, it is possible to assess the efficiency of the agent when performing in a traffic density different from that in which it was trained.

Table 7.4: Traffic volume used in the experimental simulations.

Traffic density	Training (20 minutes simulation) Medium	Testing (30 minutes simulation)		
		Low	Medium	High
Number of aircraft per $10\,000\,\mathrm{NM}^2$	40000	20000	40000	60000
Number of instantaneous aircraft	576	288	576	863
Number of spawned aircraft	886	665	1330	1994

CONFLICT RESOLUTION MANOEUVRES

All testing scenarios were run with (1) the hybrid RL + MVP method and (2) the baseline MVP method. The latter used a look-ahead time of 300 s, and moved all aircraft in all available directions. For example, in the case where MVP could vary heading, speed, and altitude, all of these directions were used to resolve the conflict. All aircraft involved in the conflict situation moved in these directions.

7.4.7. DEPENDENT VARIABLES

Three different categories of measures are used to evaluate the effect of the different operating rules set in the simulation environment: safety, stability, and efficiency.

SAFETY ANALYSIS

Safety is defined in terms of the total number of conflicts and losses of minimum separation. Fewer conflicts and losses of separation are preferred. Additionally, LoS are evaluated based on their severity according to how close aircraft get to each other:

$$LoS_{sev} = \frac{R - d_{CPA}}{R}.$$ (7.5)

Finally, the total time that aircraft spend resolving conflicts is accounted for.

STABILITY ANALYSIS

Stability refers to the tendency for tactical conflict avoidance manoeuvres to create secondary conflicts. Aircraft deviate from their straight, nominal path occupying more space of the environment, increasing the likelihood of running into other aircraft. In literature, this effect has been measured using the Domino Effect Parameter (DEP) [151]:

$$DEP = \frac{n_{cfl}^{ON} - n_{cfl}^{OFF}}{n_{cfl}^{OFF}},$$ (7.6)

where n_{cfl}^{ON} and n_{cfl}^{OFF} represent the number of conflicts with CD&R ON and OFF, respectively. A higher DEP value indicates a more destabilising method, which creates more conflict chain reactions.

EFFICIENCY ANALYSIS

Efficiency is evaluated in terms of the distance travelled and the duration of the flight. Shorter distances and shorter flight duration are preferred.

7.5. EXPERIMENT: HYPOTHESES

The RL method dictated how far in advance the MVP method initiated a deconflicting manoeuvre, and in which direction(s) each aircraft moved to resolve the conflict. The RL method could adjust its resolution to every conflict geometry. As a result, it was hypothesised that using an RL method to decide the values that the MVP method used for the calculation of the conflict resolution manoeuvres would reduce the total number of LoSs. However, it was also hypothesised that the hybrid RL + MVP method could lose efficiency at traffic densities higher than the one in which it had been trained. Conflict geometries with a higher number of involved aircraft could require different responses from those that the RL learnt.

It was hypothesised that the RL method would make use of a range of look-ahead values, as this could be a powerful way to prioritise short-term conflicts and to defend in advance against potential future severe LoSs. In previous work [233], an RL method, directed at conflict resolution, chose to defend against several LoSs (i.e., near head-on) in advance. Thus, it was expected that the RL + MVP method would choose larger look-ahead values than the baseline value of 300 s for these kinds of situations. This could increase the number of conflict resolution manoeuvres performed by the hybrid RL + MVP method. Thus, the hybrid RL + MVP solution was expected to have a higher number of conflicts when it used larger look-ahead values.

Finally, the solutions output by the MVP method were implicitly coordinated in pair-wise conflicts. It was guaranteed that both aircraft would move in opposite directions. There is no such guarantee in a multi-actor conflict. Different aircraft may resolve the conflict by moving in the same direction, making CR manoeuvres ineffective. Nevertheless, in order to reduce LoSs, the RL method had to find some sort of coordination. The RL method could not decide whether the ownship climbed or descended when the altitude was varied; this was calculated by the MVP method. However, it could decide whether the altitude was varied or not. Altitude variation would assuredly move the ownship out of conflict when intruders remained at the same altitude level. Thus, the RL method was expected to employ different combinations of actions, preventing aircraft in a multi-actor conflict from attempting to move out of conflict in the same direction.

7.6. EXPERIMENT: RESULTS

As mentioned above, the results section is divided between the training and testing phases. The first shows the evolution of the RL method during the training process. The objective was to reduce the total number of LoSs. In the testing phase, the hybrid RL + MVP method was applied to unknown traffic scenarios. Its performance was directly compared to the baseline MVP method with the same scenarios.

7.6.1. TRAINING OF THE REINFORCEMENT LEARNING AGENT

This section shows the evolution of the RL method during training. An episode was a full run of the simulation environment described in Section 7.4.1. During training, each episode lasted 20 min. Sixteen different episodes with random flight trajectories and a medium traffic density (see Section 7.4.6) were created for the training phase. These 16 episodes were run consecutively during training, so it could be evaluated whether the RL method was improving by reducing the number of LoSs for these 16 training scenarios. In total, 150 episodes were run, or roughly nine cycles of the 16 training episodes. For reference, without intervention from the CR method, when aircraft followed their nominal trajectories, the training scenarios had, on average, roughly 1800 conflicts and 600 LoSs.

SAFETY ANALYSIS

Figure 7.4 shows the evolution of the RL method for both action formulations in terms of pairwise conflicts and LoSs. A conflict was found once it was identified that two aircraft would be closer than the minimum required separation at a future point in time. Regardless of the number of aircraft involved in a conflict situation, conflicts were counted in pairs. Note that an aircraft could be involved in multiple pairwise conflicts simultaneously. A pairwise conflict was counted only once, independently of its duration.

The values obtained when the hybrid RL + MVP method controlled only heading and speed variations are indicated by 'RL + MVP Method (H + S)'. The values with 'RL + MVP Method (H + S + A)' indicate the performance when the method controlled altitude variation, on top of heading and speed variations. Both methods converged towards optimal conflict resolution manoeuvres after approximately 90 episodes (see Figure 7.4(a). They both achieved a comparable number of LoSs. However, 'RL + MVP (H + S + A)' did it with considerably fewer conflicts (see Figure 7.4(b)). This was the result of the traffic

conditions of the simulated scenarios. As all aircraft were initially set to travel at the same altitude, vertical deviations removed aircraft out of this main layer of traffic, reducing the chance of secondary conflicts.

(a) Total number of LoSs during training of the RL + MVP method.

(b) Total number of pairwise conflicts during training of the RL+MVP method.
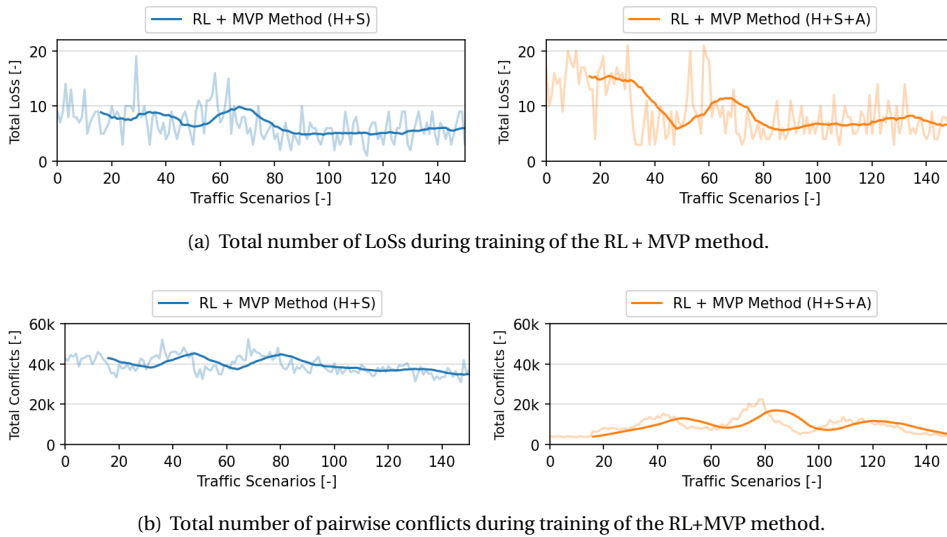
Figure 7.4: Evolution of the hybrid RL + MVP method during training.

Table 7.5 shows the actions performed by the hybrid RL + MVP method at the end of training. Naturally, the exact values used were dependent on the conflict situations that the RL faced. However, the general preference for certain actions was common to all training episodes. When RL + MVP controlled only heading and speed, it strongly favoured performing both heading and speed variations simultaneously (around 99.7% of the total). Moreover, the method favoured speed-only over heading-only actions. Subliminal speed changes can be helpful in resolving conflicts with intruders far away, without the ownship having to occupy a larger amount of airspace. However, given that speed-only actions were employed a small number of times (around 0.3% of the total), it was not clear whether the RL method understood this.

When heading, speed, and altitude were controlled, the method opted for either using all degrees of freedom simultaneously or heading and speed only (around 49.8% and 45.4% of the total, respectively). This shows that the RL method found that these two combinations were the most effective manoeuvres with the MVP conflict resolution method. Furthermore, the RL method found it advantageous to combine these two manoeuvres. As mentioned in Section 7.5, having aircraft in conflict move in different directions can be beneficial for the resolution of the conflict. The conditions in which each combination was employed are developed further in the following sections.

The RL method uses a wide range of look-ahead values. The look-ahead value directly affects *when* and *how* the ownship resolves conflicts. On average, the RL method chooses a larger look-ahead value than the baseline value of 300 s. Nevertheless, implementing these averages instead of using 300 s did not improve the efficacy of the baseline MVP method.

Table 7.5: Summary of the actions employed by the RL method in the training episodes.

Experiment	Manoeuvre (State Variation)				Look-Ahead Time	
(Degrees of Freedom)	Heading	Speed	Altitude	Usage	Average	Standard Deviation
Heading + Speed	✓			≈0%	-	-
		✓		0.3%	-	-
	✓	✓		99.7%	426 s	40 s
Heading + Speed + Altitude	✓			1.2%	-	-
		✓		0.4%	-	-
			✓	≈0%	-	-
	✓	✓		46.7%	512 s	80 s
	✓		✓	0.5%	-	-
		✓	✓	≈0%	-	-
	✓	✓	✓	51.2%	137 s	163 s

The optimal moment to act against a conflict was highly dependent on the conflict geometry, as shown by the standard deviations of the values generated by the RL method. The RL method selects shorter look-ahead values when altitude is employed. The method learnt to use altitude deviations as a tool to quickly resolve short-term conflicts.

The following sections further explore the actions of the hybrid RL+MVP method in relation to the state of the environment.

ACTIONS BY THE REINFORCEMENT LEARNING MODULE (HEADING + SPEED)

Figure 7.5 connects some of the data available in the state formulation to the actions chosen by the RL method, which could vary only the heading and speed. The look-ahead time value employed by the RL method in relation to the average distance at the CPA and time to the CPA are shown on the left. The right image displays look-ahead values in relation to the average current distance and average relative bearing.

Figure 7.5 displays the look-ahead values when both heading and speed variations were used to resolve conflicts. The RL + MVP method had a strong preference for look-ahead values above the baseline value of 300 s, as colour values below 300 s are rare in the graph. Finally, the RL method seemed to prioritise conflicts based on their average time to the CPA and distance at the CPA, as visible from the darker points on the bottom left corner of the left graph.

Figure 7.6 shows the final heading and speed variation of the RL + MVP method for the training episodes. As expected, the heading and speed variations were greater when the surrounding aircraft were closer in distance (see darker points at the bottom of the graphs on the right). The graph to the right of Figure 7.6(b) presents acceleration points (in red) when the surrounding aircraft were closer in distance, and deceleration points (in blue) when aircraft were farther away. When aircraft were closer, the ownship accelerated in order to quickly increase the distance from the surrounding aircraft. When the latter were farther away, the ownship decreased its speed in order to delay the start of the loss of minimum separation. Note, however, that although the CR method output these speeds, it was not guaranteed that the ownship would adopt the final output speed. The adoption of the new deconflicting state was dependent on the performance limits of the ownship.
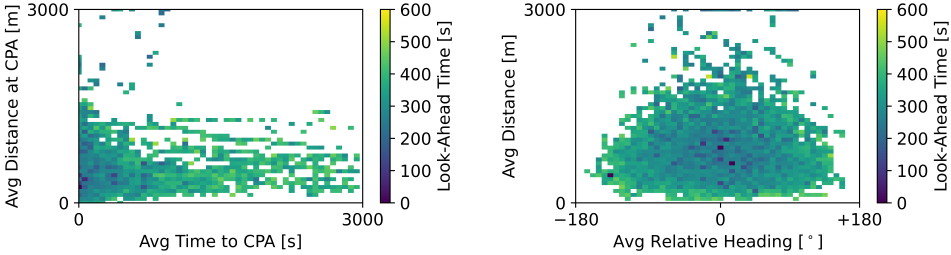
Figure 7.5: Look-ahead time values employed by the RL + MVP method when both heading and speed variations were used to resolve conflicts.
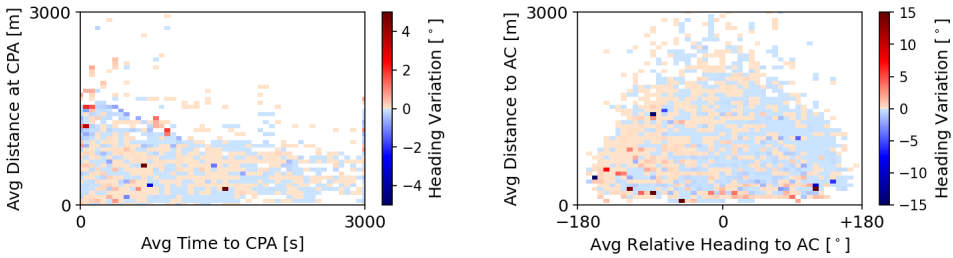
(a) Heading variation performed by the hybrid RL+MVP method.
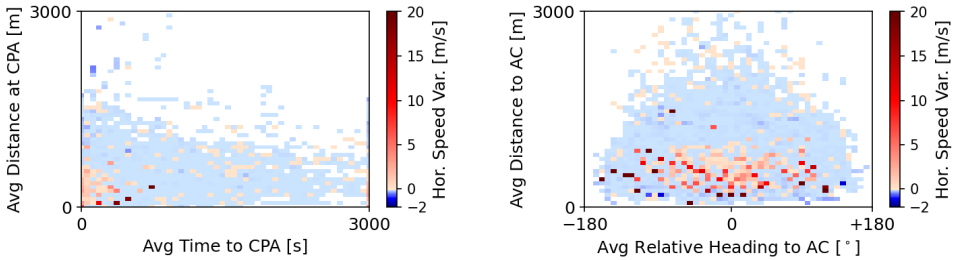
(b) Horizontal speed variation performed by the hybrid RL+MVP method.

Figure 7.6: State variation output by the hybrid RL+MVP method for conflict resolution in the training scenarios.

The results of the baseline MVP method are not shown, as differences from Figure 7.6 are not clearly visible to the naked eye. The use of an RL method to define the parameters that the MVP method used to generate the CR manoeuvre does not greatly change the magnitude of the heading and speed variations performed. However, the RL method impacts the number of intruders considered in the calculation and how far in advance the ownship initiated the CR manoeuvre. For example, when the method selects a longer look-ahead time than the baseline value of 300 s, it initiated CR manoeuvres before the baseline MVP did. In contrast, when the RL method selected lower values, it was both prioritising short-term conflicts and delaying the reaction towards conflicts farther away.

Figure 7.7 displays the situations for which the RL + MVP method instructed the

ownship to defend against a conflict and the baseline MVP method did not. This referred to situations where the RL method output a look-ahead time greater than 300 s. This resulted in the RL + MVP method defending in advance against many conflicts that the baseline MVP would only consider later in time. The graphs show that these intruders were still far away (see graph on the right) and thus would not lead to a LoS situation shortly. Nevertheless, some represent severe LoSs given the small distance at the CPA (see graph on the left).
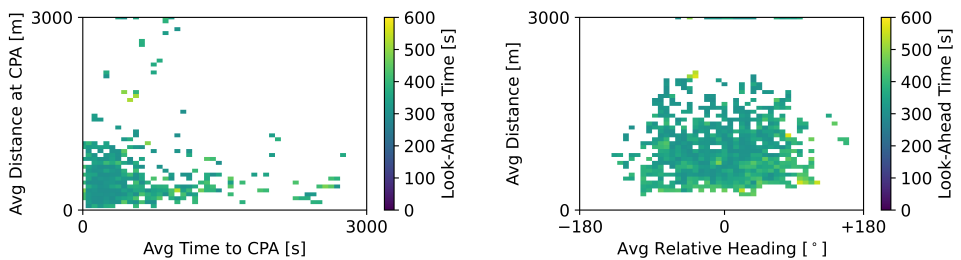
Figure 7.7: Situations in which the RL + MVP defended against surrounding aircraft, but the baseline MVP method did not.

ACTIONS BY THE REINFORCEMENT LEARNING MODULE (HEADING + SPEED + ALTITUDE)
Figure 7.8 shows the different look-ahead times selected by the RL method depending on the varied degrees of freedom. Figure 7.8(a) displays the look-ahead values produced by the RL method when it varied the heading, speed, and altitude of the ownship to resolve conflicts. Here, it had a preference for look-ahead time values under 200 s (points on the graphs are overwhelming on the darker side of the spectrum representing lower values). As a result, conflicts were resolved later than with the RL method that could only vary heading and speed. This was expected given the extra degree of freedom, i.e., the altitude variation. Since all traffic was set to fly at the same altitude level, vertical deviations were a fast way to resolve a conflict, as the ownship moved away from the main traffic layer.

Figure 7.8(b) displays the look-ahead values used for manoeuvres varying only the heading and horizontal speed. Compared to the values shown previously in Figure 7.5, the points in the graph here are lighter in colour, indicating larger look-ahead values. In this case, the RL method resorted to larger look-ahead values. This is in line with the information displayed in Table 7.5.

The direct comparison between Figures 7.8(a) and 7.8(b) show that (1) heading, speed, and altitude deviations were used with short look-ahead time values, and (2) heading and speed variations were used with larger values. Thus, it seemed that the RL method used heading and speed manoeuvres to resolve conflicts with more time in advance and resorted to altitude variation to resolve the remaining short-term conflicts. Prioritisation of short-term conflicts benefits its resolution, as the generated deconflicting manoeuvre was calculated by taking into account only the best solution for these conflicts. Moreover, by having fewer aircraft resolve conflicts in the vertical dimension, when the latter was used, it was more effective, as most of the aircraft were in the main traffic layer.

(a) When heading, speed, and altitude variation is employed.

(b) When heading, and speed variation is employed.
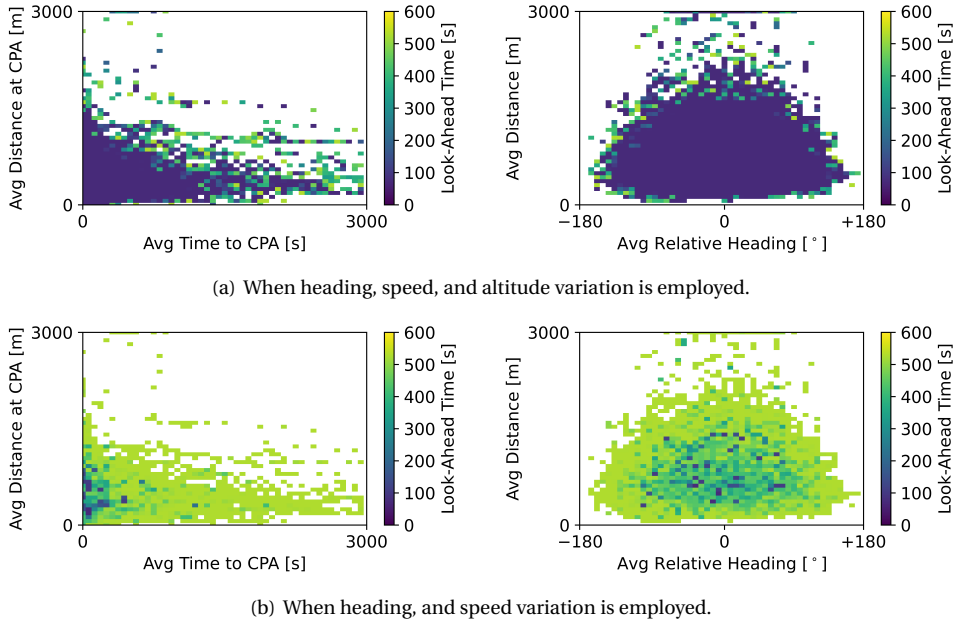
Figure 7.8: Different look-ahead time values employed by the hybrid RL+MVP method.

Figure 7.9 shows the final heading and speed variation of the RL + MVP method varying the heading, speed, and altitude. Compared to the heading and speed variations performed by the RL method that controls only heading and speed variations (Figure 7.6), stronger speed variations are visible (darker points at the bottom of the right graph). Furthermore, based on the average values of Table 7.5, this RL method defended against conflicts later than its counterpart that controlled only heading and speed variation. In conclusion, more imminent conflicts required larger state variations to resolve.

Figures 7.10(a) and 7.10(b) show the final vertical speed variation for the baseline MVP and the hybrid RL + MVP methods, respectively. There were differences in the number of conflict situations in which the methods employed altitude deviation, as Figure 7.10(b) has fewer data points than Figure 7.10(a). The baseline MVP method employed headings, speed, and altitude in all conflict situations. However, as previously shown in Table 7.5, the hybrid RL + MVP method employed altitude variation in approximately ≈50% of the total conflict situations. Thus, there were fewer occasions with altitude deviation.

Finally, analogously to the method examined in Section 7.6.1, certain look-ahead values resulted in no defensive action being adopted by the ownship. Figure 7.11(a) shows the situations where the baseline MVP did not perform a deconflicting manoeuvre but the hybrid method RL + MVP did. This was the result of the RL + MVP selecting a longer look-ahead time than the baseline value of 300 s. In turn, Figure 7.11(b) displays the situations for which the hybrid RL + MVP method did not instruct the ownship to initiate conflict resolution, and the baseline MVP method did. Here, the look-ahead values were below 300 s, during which no intruder was found. The hybrid RL + MVP method defended against conflicts more frequently than the baseline MVP method.
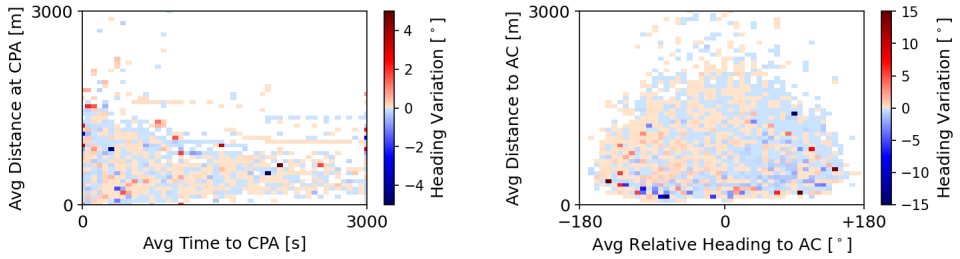
(a) Heading variation performed by the hybrid RL+MVP method.
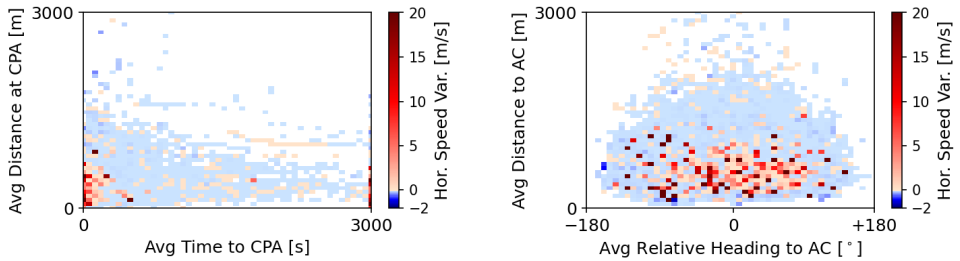
(b) Horizontal speed variation performed by the hybrid RL+MVP method.

Figure 7.9: State variation output by the hybrid RL+MVP method for conflict resolution in the training scenarios.
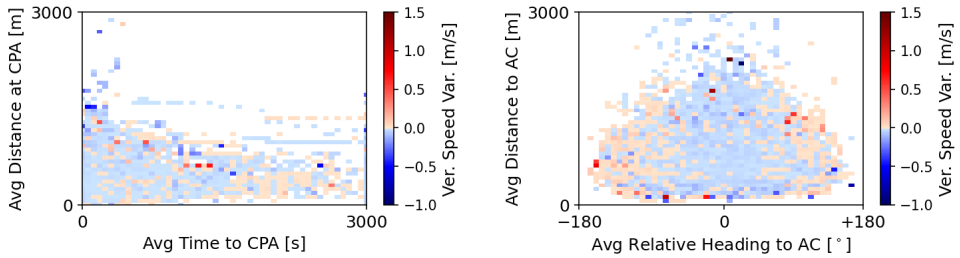
(a) Vertical speed variation performed by the baseline MVP method (look-ahead time = 300 seconds. Heading, speed, and altitude variation is always active).
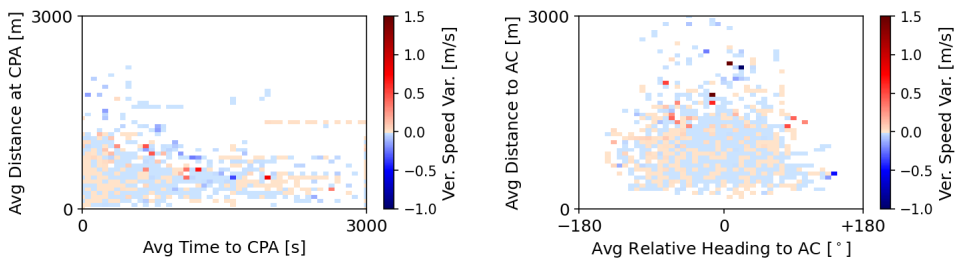
(b) Vertical speed variation performed by the hybrid RL+MVP method.

Figure 7.10: Vertical speed variation performed by the baseline MVP and the hybrid RL+MVP methods.
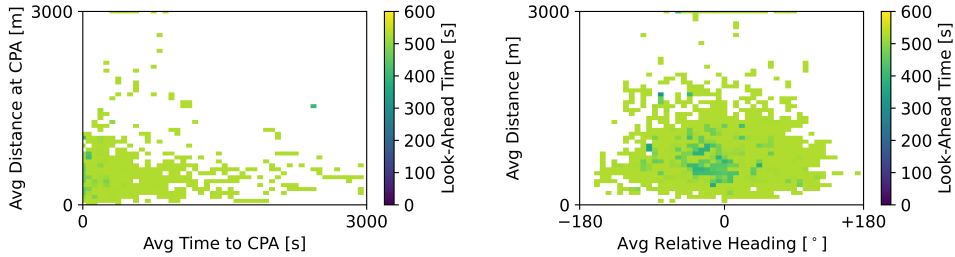
(a) Situations in which the RL+MVP defended against surrounding aircraft, but the baseline MVP method did not.
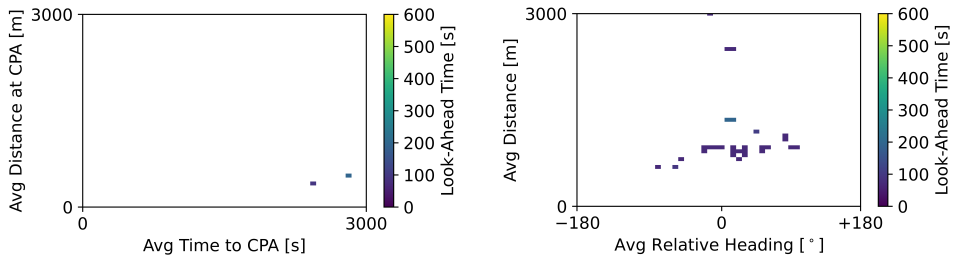
(b) Situations in which the MVP defended against surrounding aircraft, but the tested RL+MVP method did not.

Figure 7.11: Situations in which only of the methods, either the hybrid RL+MVP method or the baseline MVP, defended against surrounding aircraft but the other method did not.

7.6.2. Testing of the Reinforcement Learning Agent

The trained RL + MVP method was then tested with different traffic scenarios at low, medium, and high traffic densities. For each traffic density, three repetitions were run with three different route scenarios, for a total of nine different traffic scenarios. During the testing phase, each scenario was run for 30 min. In both phases, the results of the RL method were compared directly with those of the baseline MVP method.

Safety Analysis

Figure 7.12 displays the mean total number of pairwise conflicts. A pairwise conflict was counted only once, independently of its duration. Note that this was the number of detected conflicts with a baseline value of 300 s, independent of the final value selected by the RL method, to warrant a direct comparison. Employing the RL method with MVP resulted in a considerably higher number of conflicts than using the baseline MVP method. However, this is not necessarily negative, as previous research has shown that conflicts help spread aircraft within the airspace [2].

The increase in the total number of conflicts was a direct consequence of the higher number of deconflicting manoeuvres performed by the hybrid RL + MVP method in comparison to the baseline MVP method (see Figures 7.7). At high traffic densities, conflict avoidance manoeuvres led to secondary conflicts, as aircraft occupied more airspace by deviating from their nominal, straight path. The increase in the total number of conflicts was less significant when MVP could vary altitude, on top of heading and

speed (see graph on the right). This was related to the fact that most aircraft travelled at the same altitude; thus, varying the altitude was less likely to result in secondary conflicts.
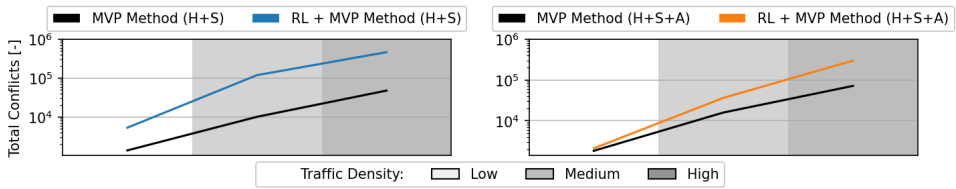
Figure 7.12: Mean total number of pairwise conflicts during testing of the RL agent.

Figure 7.13 displays the time in conflict per aircraft. An aircraft entered 'conflict mode' when it adopted a new state computed by the MVP method. An aircraft exited this mode once it was detected that it was past the previously calculated time to the CPA (and no other conflict was expected between now and the look-ahead time). At this point, the aircraft redirected its course to the next waypoint in its route. The time redirecting towards the next waypoint was not included in the total time spent in conflict.

While aircraft spent more time in conflict with the hybrid RL + MVP method, the increase in time in conflict did not correlate directly with the increase in the total number of conflicts (see Figure 7.12). This meant that most conflicts were short in duration and quickly resolved. Additionally, the hybrid RL + MVP method controlling only heading and speed variation (see graph on the left), at the lowest traffic density, had the highest time in conflict, although it had the fewest conflicts. This indicated that the way conflicts were resolved had a greater impact on the total time spent in the conflict.
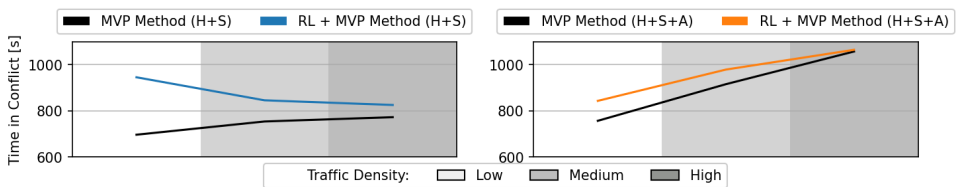
Figure 7.13: Time in conflict per aircraft during testing of the RL agent.

Figure 7.14 displays the total number of LoSs. Reducing the total number of LoSs was the main objective of the RL method. The results show that having the RL method decide the input values for the MVP method led to a reduction of the total number of LoSs on all traffic densities, even at a higher traffic density than the RL method was trained on. This proved that the elements optimised by the RL method, namely, (1) the prioritisation of conflicts depending on the degrees of freedom and (2) the heterogeneity of directions between aircraft in a conflict situation, were common to all traffic densities.

Figure 7.15 displays the LoS severity. With the hybrid RL + MVP method, the LoS severity was slightly higher, but not to a significant extent. It was likely that the RL + MVP method prevented some of the lowest severity LoSs, leaving out the more severe ones.
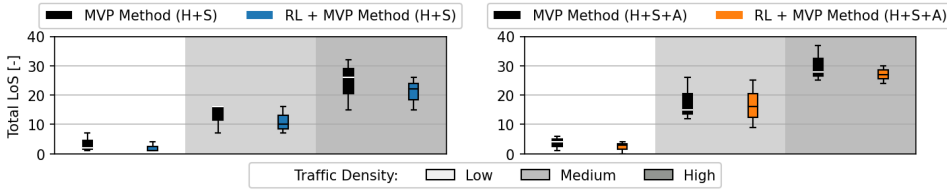
Figure 7.14: Total number of LoS during testing of the RL agent.
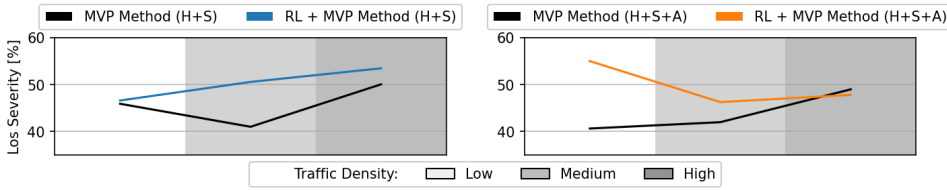
Figure 7.15: LoS severity during testing of the RL agent.

STABILITY ANALYSIS

Figure 7.16 shows the DEP during the testing of the RL method. The increase in DEP was comparable to the increase in the total number of conflicts (see Figure 7.12). The increase in the total number of conflicts was a result of a higher number of deconflicting manoeuvres, leading to a higher number of secondary conflicts.
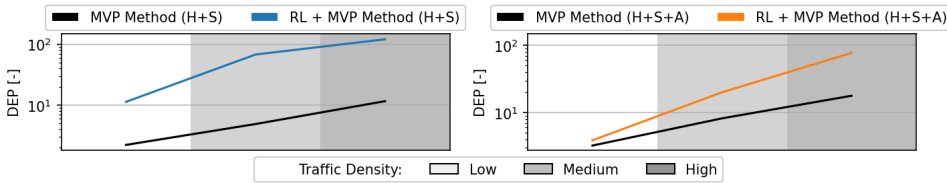
Figure 7.16: Domino effect parameter (DEP) during testing of the RL agent.

EFFICIENCY ANALYSIS

Figures 7.17 and 7.18 show the flight time and 3D flight path, per aircraft, during testing of the RL method, respectively. The total flight time was a direct result of the time in conflict (see Figure 7.13). The resolution of the higher number of conflicts also increased the 3D path travelled as aircraft moved away from their nominal, straight path to resolve conflicts.

7.7. DISCUSSION

Recent studies have focused on using RL approaches to decide the state deviation that aircraft should adopt for successful CR. However, the efficacy of these methods still cannot surpass the performance of state-of-the-art geometric CR algorithms at higher traffic densities. This study posed the question of whether the RL method could, instead, be used to improve the behaviour of these geometric CR algorithms. This was an exploratory work
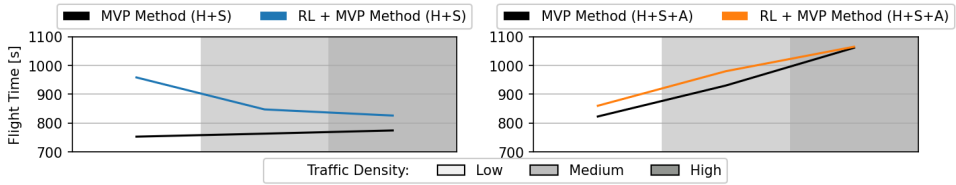
Figure 7.17: Flight time during testing of the RL agent.
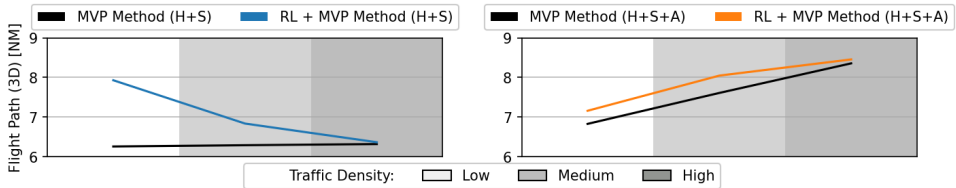
Figure 7.18: 3D flight path during testing of the RL agent.

in which an RL method was used to generate the parameters used by an CR algorithm to generate a CR manoeuvre.

The results showed that a hybrid method, combining the strengths of both RL and geometric CR algorithms, led to fewer losses of minimum separation when compared to using fixed, predefined rules for CR. The benefit from the RL method lay on (1) the ability to determine how far in advance the ownship should initiate the CR manoeuvre and (2) in which directions to resolve the conflict. The former allowed for the prioritising of short-term conflicts or the advanced defence against far away conflicts. The latter induced an heterogeneity of the directions that aircraft used to move out of a conflict, which could be beneficial as it prevented aircraft from moving in the same direction. The following subsections further develop these topics.

Finally, questions remain regarding the application of this RL approach to other geometric CR methods and operational environments. Here, the RL method had a limited action formulation, being responsible for modelling only four parameters. Nevertheless, a generation of a conflict resolution manoeuvre can include a multitude of parameters. Additionally, the efficacy of a hybrid RL + CR method is dependent on the RL understanding how the values affect the performance of the CR algorithm. More research is needed to determine whether this approach can be successfully applied in a real-world scenario.

7.7.1. CONFLICT PRIORITISATION

By controlling look-ahead time, the RL method selected against which layers of aircraft the ownship would defend. With shorter look-ahead values, the ownship defended against the closest layers of aircraft, prioritising short-term conflicts. With larger look-ahead values, a higher number of layers of aircraft were defended against, and conflicts were resolved with more time in advance. However, considering a greater number of intruders in the generation of the resolution could make the final CR manoeuvre less effective against each of these intruders.

The RL method prioritised short-term conflicts when the surrounding aircraft were closer in time to a loss of minimum separation. This ensured that CR focuses only on these conflicts, increasing the likelihood of successfully resolving them. Larger look-ahead values were employed when the surrounding aircraft were farther away. In a way, the look-ahead values varied so that a limited number of aircraft were included in the generation of the CR manoeuvre.

Furthermore, how far the RL method defended in advance depended on the efficacy of the CR manoeuvre. With a less efficient manoeuvre, the RL method understood that a longer reaction time was needed, as the manoeuvre employed needed a longer time to establish a safe distance. On the contrary, when altitude variation was used to resolve conflicts, lower look-ahead values were used. As all aircraft flew at the same altitude, climbing or descending was a powerful tool, moving the ownship out of the main traffic layer. Thus, less reaction time was needed in that case.

7.7.2. HETEROGENEITY OF CONFLICT RESOLUTION DIRECTIONS

The RL method found that, to resolve a conflict, moving the ownship in multiple directions simultaneously was beneficial. However, altitude or heading deviations were less effective when intruding aircraft moved in the same direction to resolve the conflict. The RL method understood that having different combinations of state variations (i.e., heading, speed, and/or altitude variation) led the intruding aircraft to resolve in different directions, increasing the chances of the resolution manoeuvres being effective.

It can be considered that the biggest advantage of allowing a CR algorithm to control multiple degrees of freedom is the ability to use different combinations of state variations to resolve conflicts. However, this heterogeneity should be based on rules to ensure that different combinations are used per aircraft in conflict with each other. From the results obtained, it was not clear how the RL method made these decisions.

7.7.3. FUTURE WORK

The values chosen by the RL method depended on the operational environment, flight routes, and performance limits of the aircraft involved. Future work will explore this RL approach under uncertainties. In this case, it is likely that smaller look-ahead times would be picked, as higher values would entail the propagation of more uncertainties. Additionally, having a single RL method responsible for separating aircraft under position uncertainties may lead the method to adopt a more defensive stance and start considering bigger separation distances. The latter will lead to larger CR manoeuvres, which, in turn, increase flight path and time. A better option may be to have a second RL method responsible for determining the most likely position of the intruders under uncertainties [255]. This new position would then be used by the RL method responsible for guaranteeing the minimum separation between aircraft.

Finally, future work can benefit from using RL methods to directly prioritise specific intruders. This is far from a trivial task. Previous work has shown that a great deal of training is necessary for an RL method to understand the effect of enabling/disabling each intruder in the generation of a CR manoeuvre [256]. However, it is of interest to explore this area of research. Different aircraft at similar look-ahead values can then be included/excluded from the CR manoeuvre.

7.8. CONCLUSION

This chapter proposed a different application of reinforcement learning (RL) in the area of conflict resolution. RL is typically used as the method that is fully responsible for safeguarding the separation between aircraft. Although great progress has been made in this area, these methods cannot yet surpass the performance of the state-of-the-art distributed, geometric CR methods. This work used RL, instead, to help improve the efficacy of the latter geometric methods. This article employed an RL method responsible for optimising the values that a geometric CR algorithm used for the generation of conflict resolution manoeuvres. Namely, the RL was responsible for defining the look-ahead time at which the geometric CR method started defending against conflicts, as well as in which directions to move towards a nonconflicting trajectory.

The advantage of RL approaches is that they can find optimal solutions to a multitude of different conflict geometries, which would be arduous to develop through man-written rules. The hybrid combination of RL + CR successfully obtained fewer losses of minimum separation than a baseline CR method which used hard-coded, predefined values for all conflict geometries. The main benefits resulted from (1) the prioritisation of conflicts depending on the degrees of freedom and (2) the heterogeneity of deconflicting directions between aircraft in a conflict situation. These two rules improved the resolution of conflicts at different traffic densities.

However, more research is needed to validate whether this approach is still effective in real-world scenarios under uncertainties, which can increase the gap between the practical efficacy of RL methods and its expected theoretical performance. Additionally, future work will focus on translating this application to other geometric CR algorithms and operational environments. The work performed herein was focused solely on one geometric CR algorithm that took two values as input. To fully analyse whether RL approaches can define the best values for the generation of CR manoeuvres, this work must be expanded to different algorithms, especially those with a larger number of variables.

7

8

ON THE LIMITATIONS OF USING REINFORCEMENT LEARNING IN AVIATION

Reinforcement learning (RL) approaches have been successfully used in multiple areas of research. Deep reinforcement learning, in particular, has recently emerged as a very successful method to tackle decision-making problems where the objective is to increase the distance between operating agents.

The introduction of RL approaches in aviation is still modest, due to hesitation in moving from human to autonomous control, in part due to the black-box nature of RL approaches. However, the need for methods capable of faster processing, in order to enable the higher traffic densities predicted for future operations, has led researchers to study how reinforcement learning can improve current aviation procedures.

Throughout the previous chapters of this thesis, published works were introduced that looked directly at techniques on how to improve multi-agent conflict resolution, specifically in an urban airspace. This chapter falls outside of this scope, working as a reflection of the work that did not get published, mainly due to unsuccessful techniques, limitations of employing reinforcement learning, and the lessons resulting from it.

8.1. PREFACE

Many studies, including those presented in this thesis, have been developed on the use of reinforcement learning (RL) to resolve conflict resolution problems. These offer a variety of different methods and approaches. However, research often fails to explore the several issues that one may encounter when developing an RL approach. Published works focus only on its achievements under a positive outlook. However, several iterations and decisions taken towards facilitating the convergence of the RL method are often not given enough attention. Moreover, attempts that did not succeed are not published; although much can be learnt from these. This chapter aims to shed light on the other side, namely, the various problems that one may encounter when employing RL for conflict resolution.

8.2. CHAPTER ORGANISATION

A graphical representation of the content of this chapter can be found in Figure 8.1. Three different elements belonging to the development process of an RL method were considered: the necessary steps (in blue), and the parameters to consider before implementing these steps (in yellow) to prevent potential issues (in orange). The chapter is divided into three main sections that follow a timeline progression on the implementation of an RL method:

1. Building the RL method: describes the considerations one must take before actually implementing the RL method. This includes decisions regarding the type of RL algorithm, whether a single or multi-RL agent is more appropriate to the problem at hand, and limitations in terms of action type.

2. Training of the RL method: approaches potential issues during training of an RL method, such as long training time or non-convergence of the method. To avoid these issues, several decisions must be made on how much information to give the method and how to balance exploration/exploitation must be made.

3. Testing of the RL method: analyses potential issues with poor generalisation of the method to situations different from the ones it was trained with. Additionally, the need for benchmark situations and a direct comparison with other conflict resolution methods is briefly addressed.

Each section is independent and can be read separately. In some of the sections, the properties of some existing work are shown to illustrate which practises are more common. The same works are referred to in each section. Thus, for a full picture of these works, the reader should consider Tables 8.1 to 8.4 simultaneously. Note that this chapter focuses only on the RL methods used for conflict resolution. Other environments, with different dynamics, can lead to different conclusions.

8.3. BUILDING THE REINFORCEMENT LEARNING METHOD

The decision of which RL algorithm to use is not trivial and has a great impact on the final results obtained. In this section, we describe the main questions that one must consider:
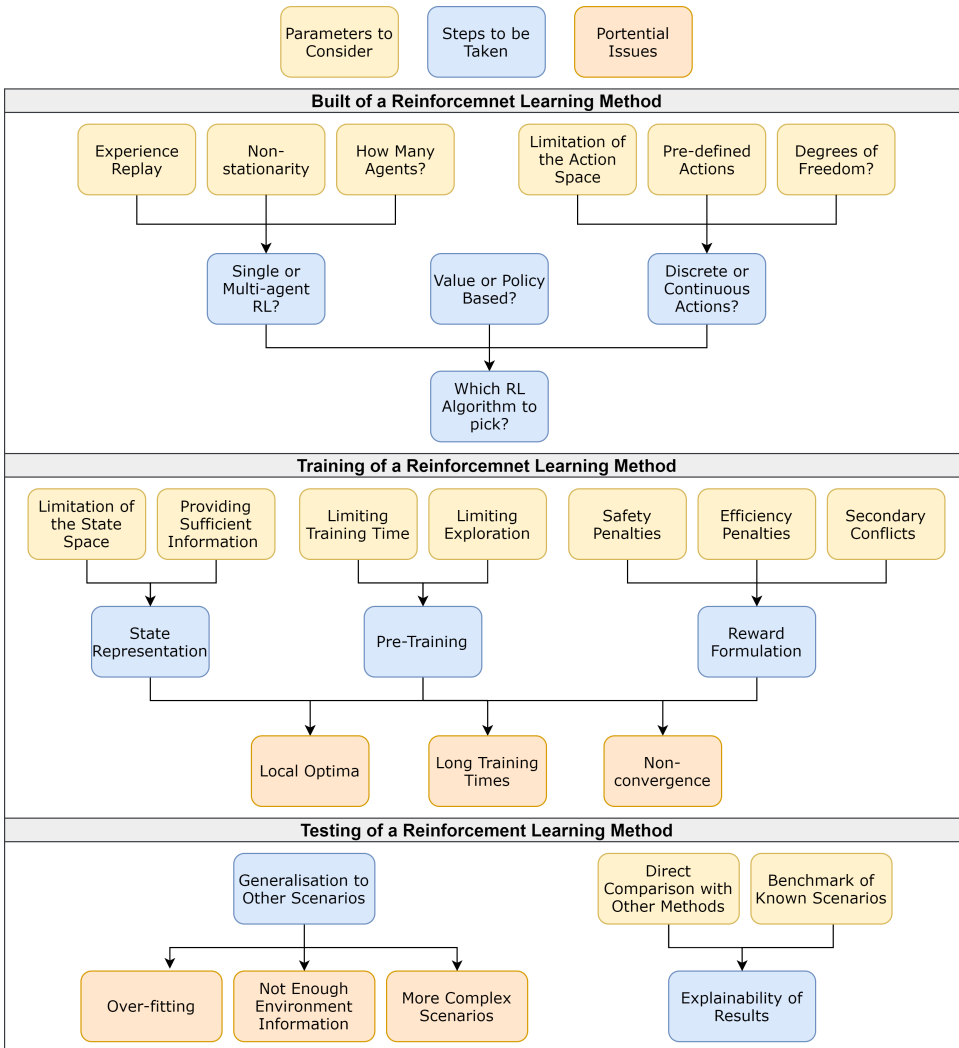
Figure 8.1: High-level diagram of the organisation of this chapter.

- Whether multi or single agent should be applied: multi-agent RL is often used to handle the non-stationarity of the environment. However, it limits the number of agents with which the method can interact and increases the complexity of the implementation.
- Action formulation: the preference for static or continuous actions directly conditions the type of RL algorithms that can be employed, as algorithms may be limited to one type.
- Value or policy based RL algorithms: RL algorithms are value-based, policy-based, or both. Value-based optimises first the value function and then derives the opti-

mal strategy. Policy-based directly optimises the objective function. Actor-critic methods attempt to do both.

The previous questions are developed in further detail in the following sections. Finally, common selected RL algorithms for conflict resolution are shown in Tables 8.2 to 8.1 for reference.

8.3.1. NON-STATIONARITY AND EXPERIENCE REPLAY

Current research [186, 187] shows that emergent behaviour and complexity arise, not as a result of the number of agents, but from the agents interacting and co-evolving. From the point of view of each agent, the environment is non-stationary and, as training progresses, modifies in a way that cannot be explained by the agent's behaviour alone. Non-stationarity is particularly important when the RL method is used for distributed conflict resolution, as the result of an action is also based on the actions taken by the surrounding aircraft.

Non-stationarity is a problem when RL methods make use of experience replay (ER). Off-policy RL methods reuse early experience for training. ER consists of storing, into a buffer, experience about the agent actions in the environment, and then replaying this experience during training. This is, for example, the case with Deep Q-Network (DQN) [257], Deep Deterministic Policy Gradient (DDPG) [163], and the Soft Actor-Critic [229]. These methods sample data uniformly from the buffer when performing parameter updates. A uniform sampling scheme implicitly assumes that the data in the replay buffer are of equal importance. However, this is not the case; the data in the agent's replay memory quickly becomes obsolete as it no longer reflects the current dynamics of the environment, when all agents are learning and evolving throughout their training [258].

Recent studies have tried to overcome these issues with ER by improving the use of replay experience to emphasise recently observed data while not forgetting the past [259]. A simple solution can be to limit the size of the replay buffer. As a result, only information from the more recent past is retained. Larger replay buffers typically contain more off-policy data, since data from older policies remain in the buffer for longer [260]. Other RL methods, such as Proximal Policy Optimisation (PPO) [261], do not rely on ER. When using PPO, the agent learns directly from the environment, and once it uses a batch of experiences, it discards it after performing a gradient update.

8.3.2. SINGLE VS MULTI-AGENT REINFORCEMENT LEARNING

A solution to handle the non-stationarity of having all agents evolving together, is to use multi-agent reinforcement learning (MA-RL). Instead of considering one agent's interaction with the environment, multiple agents that share the same environment are trained. In MA-RL, each agent has its own centralised critic, that approximates and learns the action value function given the observations and actions of all agents. Note, however, that MA-RL uses the actions and observations of all agents as input. Consequently, complexity is proportional to the number of agents.

It is not clear whether it is more advantageous to use single or multi-agent RL. As visible in the last column of Table 8.1, there is no clear preference for one kind in current studies. The selection of a single agent or multi-agent mainly depends on the problem to be tackled. The single agent approach has two advantages. First, it is easier to implement

compared to MA-RL. Second, the single-agent can be used on any aircraft, it does not limit the number of aircraft. Instead, MA-RL represents the observations of all agents in its state formulation. As the state array has a fixed size, it can only be used with a fixed number of aircraft. However, theoretically, MA-RL is expected to achieve a better global safety, by optimising the actions of all agents together. In practise, this is not always observed. For example, through practical experiments, Zhang [248] concluded that multi-agent learning is weaker than single agent under the same amount of training. It is thus necessary to balance the optimisation objectives of multi-agents in an appropriate way. It is reasonable to expect that MA-RL may require more training due to the increasing state and action formulation, as well as the need to correctly identify which actions had more impact on the reward.

Moreover, single agent RL resembles the behaviour of geometric CR methods. These look at the current situation of each aircraft and its closest neighbours to calculate implicitly coordinated conflict resolution manoeuvres. The shortest-way-out principle ensures implicit coordination in one-to-one conflicts. As single conflicts are always geometrically symmetric [43], both aircraft in a conflict will take (opposite) measures to evade the other. Ultimately, the hope would be that a single RL agent could emulate the success of geometric CR methods, and produce implicitly coordinated actions based on the same limited information. Nevertheless, this level of success has not yet been achieved. In previous work [233], we show that the actions of the RL method are (somewhat) coordinated in pairwise actions by having the aircraft always turning in the same direction.

8.3.3. Value-Based vs Policy-Based

RL algorithms can be value-based, policy-based, or both. Value-based algorithms do not store any explicit policy, only a state-action value. Their predictions assign a score (maximum expected future reward) for each possible action, at each time step, given the current state. The policy here is implicit, and the best action is selected. In turn, policy-based methods directly learn the policy function that maps state to action. The method tries to optimise this policy without using a value function. An actor-critic algorithm learns both a policy and a value function. The critic learns a value function, which is then used to update the actor's policy parameters in the direction that increases the expected rewards.

Both types, value and policy based, are theoretically guaranteed to converge to an optimal policy in the end. However, there are different advantages and disadvantages for their implementation:

- ✓ Value-based methods are easier to implement, as they remove the need to balance exploration and exploitation.
- ✗ Value-based methods must create a value for each possible action, which is not feasible for a large number of possible actions.
- ✗ Value-based methods can have a large oscillation during training, as their choice of action may change dramatically due to the small changes in the estimated action values.
- ✗ Value-based methods cannot learn a stochastic policy.
- ✓ Policy-based methods are more effective in high dimensional action spaces, or

when using continuous actions.

✓ Policy-based methods typically have a more linear convergence than value-based methods.

✓ As policy-based methods choose between actions using a distribution, they can learn a stochastic policy.

× Policy-based methods often converge on a local maximum rather than the global optimum.

According to existing research, policy-based methods are typically faster and more stable than value-based methods [262]. However, some works have also found that value-based can outperform policy-based with simple state formulations [263].

8.3.4. WHICH REINFORCEMENT LEARNING ALGORITHM TO PICK?

The previous subsections defined the main considerations that limit the number of possible RL algorithms from which to choose. From the final set of suitable algorithms, it is not always clear which one to choose. Typically, the choice depends on how often the algorithm has been successfully implemented and tested in other work.

As a reference, Table 8.1 shows some examples of studies that use reinforcement learning to resolve conflicts. The column 'Algorithm' defines the RL algorithm employed by the study, and 'Type' whether it is value-based, policy-based, or both. The column 'Discrete/Continuous Actions' indicates whether the RL method produces discrete or continuous actions, and finally the 'Single/Multi Agent RL' indicates whether Single or Multi-Agent RL is employed. Based on the overview in Table 8.1, PPO and DDPG are the most commonly used methods. This may serve as an indication for researchers. However, ultimately the best algorithm can only be decided by empirical testing.

Table 8.1: Example of algorithms employed in RL methods for conflict resolution in recent studies.

Study	Algorithm	Type	Discrete/Continuous Action	Single/Multi Agent RL
Brittain [180]	PPO	Value + Policy	Discrete	Multi
Dalmau [240]	DGN [264]	Value	Discrete	Multi
Henry [237]	Q-Learning	Value	Discrete	Single
Isufah [228]	PPO	Value + Policy	Continuous	Multi
Li [236]	DQN	Value	Discrete	Single
Pham [227]	DDPG	Value + Policy	Continuous	Single
Ribeiro [233]	SAC	Value + Policy	Continuous	Single
Tran [265]	DDPG	Value + Policy	Continuous	Single
Zhao [21]	PPO	Value + Policy	Continuous	Single/Multi

8.3.5. ACTION FORMULATION

Table 8.2 identifies the action formulations for the same studies whose RL algorithms were previously described in Table 8.1. The column labelled 'When?' indicates when the action is taken, and the column 'Degrees of Freedom' describes which degrees of movement of the aircraft the RL method controls. Note that whether the RL method employs discrete or continuous actions was previously defined in Table 8.1 for the same studies.

Table 8.2: Example of action formulations employed in RL methods for conflict resolution in recent studies.

Study	When?	Degrees of Freedom
Brittain [180]	Every 12 seconds	3 discrete speed options (decelerate, hold current speed, acceleretate)
Dalmau [240]	Every 5 seconds	Heading + Speed discrete options
Henry [237]	Every 20 seconds	Discrete options: accelerating, decelerating, increasing/decreasing entry time in TMA, increasing decreasing Point Merging System arc length, changing landing runway, no action
Isufah [228]	Every 15 seconds	Heading + speed variation
Li [236]	Every seconds	Discrete advisories: clear of conflict, weak turn left, weak turn right, strong turn left, strong turn right
Pham [227]	Every 30 seconds	Heading change, time, x and y coordinates of the return point
Ribeiro [233]	Every second	Heading + speed (+ altitude) variation
Tran [265]	Once per episode	Heading variation
Zhao [21]	Every 40 seconds	Heading + speed variation

First, the number of degrees of freedom that the RL method controls should be carefully decided. A larger action formulation increases the number of possible state-action combinations, increasing the necessary training time to find the optimal combination. In certain cases, it may even limit the ability of the RL method to converge to optimal solutions in an acceptable amount of time.

The choice of RL method is directly connected to the type of action space. DQN, for example, can handle only a discrete action space. Other methods, such as SAC and PPO, can be used for both discrete and continuous action spaces. With regard to discrete and continuous actions, there is an advantage for both. Discrete solutions limit the number of possible solutions, likely resulting in faster training. Works such as Zhao [21], show that discrete action spaces perform better than the continuous action space in convergence speed and robustness. However, limiting the number of actions of the method also limits the impact that it may have on the environment. With continuous actions, the method may perform more efficient conflict resolution manoeuvres, with the state varying only the amount necessary to resolve the conflict. However, more precise actions come at the cost of longer training times.

But even with continuous actions, the range of possible values that this action may take must be limited [228]. The action output by the RL methods must be translated into the state variation of the aircraft. For example, when the RL method outputs values between 0 and 1 which are then translated into heading values of 0° to 360°, it means that a variation of 0.1 in the action of the method results in a heading variation of 36°. This is a large heading variation that may have a great impact on the final reward. When small-magnitude differences in the actions of the method result in very different rewards, it is likely that the method will not be able to correctly identify the impact of the action in the environment. Finally, performance limits must be taken into account. At each timestep, there is a maximum state variation that an aircraft can achieve. With great state variations, the reward received by the RL method may not be based on the results with the state output by the method but, instead, on the maximum variation that the aircraft was able to achieve within the available time.

8.4. TRAINING OF THE REINFORCEMENT LEARNING METHOD

This section discusses the decisions to be made during the training of the RL method:

- State representation: it must provide enough information for the RL method to be able to decide upon the action and correctly understand its impact on the environment. However, larger state arrays considerably increase training time and the complexity of finding the optimal action per state.
- Reward formulation: the penalties given to the method must be a direct consequence of its produced actions. Additionally, the variation of rewards must be enough for the method to be able to determine which actions maximise the rewards received.

Tables 8.3 and 8.4 describe the state and reward formulations defined by the RL methods previously included in Tables 8.1 and 8.2. State and reward information should be optimised to prevent issues with the RL method, such as getting stuck on local optima, impractical training times, and non-convergence. Finally, this section refers to previous usages of pre-training with RL methods and its advantages and disadvantages.

8.4.1. STATE REPRESENTATION

Table 8.3 describes the state formulation for the same studies analysed previously in Tables 8.1 and 8.2. The type of information in the state formulation is indicated in column 'Information'. Column 'Number of Neighbours' defines total number of aircraft represented in the state. Finally, column 'Conflicts' identifies the type of conflicts that aircraft may run into, whether pairwise or multi-actor conflicts. We consider that the type of conflict strongly influences the amount of information represented in the state formulation.

Table 8.3: Example of state formulations employed in RL methods for conflict resolution in recent studies.

Study	Information	Number Aircraft	Conflicts
Brittain [180]	Distance to target, ownship's speed, acceleration, route identifier, distance to other aircraft, distance from ownship to intersection, distance from other aircraft to intersection	1 aircraft	Single
Dalmau [240]	Distance and bearing to target, ownship's speed. Distance at CPA, time to CPA. and bearing to other aircraft	All other aircraft (N=15)	Multi-Actor
Henry [237]	Ownship's speed, entry time TMA, Point Merge System arc length, runway assignment	0 aircraft	Single
Isufah [228]	Latitude, longitude, heading, and speed of each aircraft	0 aircraft	Pairwise
Li [236]	Distance from the ownship to other aircraft, relative angle to other aircraft, speed of the ownship, speed of the other aircraft	Closest 4 aircraft	Pairwise
Pham [227]	Environmental uncertainty level, x/y positions of the ownship at CPA, x/y directions of the ownship at CPA, distance at CPA	1 - 13 aircraft	Pairwise/ Multi-Actor
Ribeiro [233]	Ownship's speed, drift to target, distance at CPA, time to CPA	4 aircraft	Multi-actor
Tran [265]	Relative position between aircraft	3 aircraft	Pairwise
Zhao [21]	Raw pixels of velocity plane with velocity obstacles	4 aircraft	Multi-Actor

The main concern when defining a state formulation is whether the information present in the state is sufficient for the RL method to successfully resolve conflicts. This is not trivial. The state will never be a complete representation of the real-world scenario in which the method is applied. Such would result in a large state dimension, which can lead to unpractical training times or even non-convergence towards an optimal solution.

As seen in Table 8.3, most methods employ a state formulation that contains part of the current state of the aircraft that is resolving the present conflict, as well as some relative data to other nearby aircraft. However, having a high number of aircraft represented in the state formulation would also lead to a large state array. Most examples represent a maximum of 4 neighbouring aircraft. However, most studies also focus only on pairwise conflicts or multi-actor conflicts with a small number of involved aircraft. Thus, these 4 aircraft are sufficient to resolve the conflicts and, in most cases, guarantee that the aircraft do not move towards creating secondary conflicts with adjacent aircraft. This is not the case in high traffic densities, where more aircraft may be involved in a conflict.

Studies have developed state formulations that can describe an infinite number of aircraft. In previous work [182], we set the number of aircraft in the areas surrounding the ownship in each position of the state formulation array. However, this system also has its disadvantages. All aircraft in one portion of the airspace are considered identical, and their relative position is rounded to the same position. In turn, Zhao [21] uses raw pixels from the velocity space surrounding the ownship to define the environment. An image may contain unlimited information. However, to be used by an RL method, an image must be discretised into a limited amount of information, resulting in the same generalisation issues.

8.4.2. Reward Formulation

Table 8.4 shows the reward formulation for the same studies that have been previously evaluated in Tables 8.2 to 8.1. Column 'When?' indicates when the reward is received by the RL method. Columns 'Safety Penalty' and 'Efficiency Penalty' describe the safety and efficiency elements that are penalised with a negative reward, respectively. Finally, the column 'Single/Global' defines whether a single reward, taking only into account the current state of the ownship, or whether a global reward that considers the state of all aircraft is considered.

Safety Penalty

The choice of rewards can greatly influence the results of a reinforcement learning method. In particular, in terms of providing enough information for the method to learn. Two main elements can be defined for safety: a loss of minimum separation (LoS) when two aircraft get closer to each other than the predefined minimum separation distance. A conflict is a future prediction of a LoS. Previous work [266], tested two different RL methods: (1) one RL method rewarded only on the total number of conflicts; (2) one RL method rewarded only on the total number of losses of minimum separation. We show that the great variation in the number of conflicts may make it harder for the RL method to correctly relate actions to potential rewards. As a consequence, the training time was considerably longer when the reward was based on the total number of conflicts. Nevertheless, the total number of LoSs, may be too scarce for a successful training.

Table 8.4: Example of reward formulations employed in RL methods for conflict resolution in recent studies.

Study	When?	Safety Penalty	Efficiency Penalty	Single/Global
Brittain [180]	Every step	Losses of Separation, Distance Between Aircraft	Speed changes	Global (to all aircraft in conflict)
Dalmau [240]	Every step	Losses of Separation, Conflict	Delay, drift, speed change	Global
Henry [237]	Every step	Losses of Separation	Landed in the correct runway	Single
Isufah [228]	Every step	Time until loss of separation and CPA, LoSs	Difference from track and optimal speed, fuel consumption, airspace complexity	Global
Li [236]	Every step	Distance between aircraft, LoSs, conflicts	Heading deviations	Multi-actor
Pham [227]	Every step	Distance between aircraft	Trajectory deviation	Single
Ribeiro [233]	Every step	Losses of Separation		Global
Tran [265]	Termination step	Losses of Separation	Deviation from route	Single
Zhao [21]	Termination step	Losses of Separation	Deviation from route	Single

In chapter 4, both the total number of LoSs and conflicts were used to give enough information to the RL method. However, the relationship between two safety elements must be taken into account. For example, tactical conflict resolution often results in an increase in the total number of conflicts. However, these additional conflicts can also have a positive impact on safety [2]. Additionally, studies often reward an increase in the relative distance between the ownship and intruders (see Table 8.4). Nevertheless, this should only be rewarded up to the minimum separation distance. Creating larger distance will force considerably more state variations that can have a negative impact on global safety, as the behaviour of the intruders becomes more unpredictable as they constantly modify their movement to increase their distance to neighbouring aircraft.

Additionally, it often happens that the RL methods adopt a defensive behaviour towards preventing negative rewards. In previous work [233], we showed that the RL method can apply, on average, greater distances than the minimum separation distance to ensure that it will not receive a negative reward for slightly crossing the protected zone of an intruder. However, often non-severe LoSs can be better for stability of the environment than forcing aircraft to change their state. A recommendation is for the behaviour of the RL methods to be smoothed out by reward shaping [267] in the future, where the RL method may choose not to alter the state of aircraft significantly in the case of short, non-severe LoSs.

EFFICIENCY PENALTY

One of the biggest advantages of RL methods is their ability to consider several factors simultaneously. A human can successfully react to the distance between multiple aircraft. This task becomes more complex when, together with the distance, the fuel cost, and the increase in travel time, are taken into account simultaneously. When multiple parameters

are considered, especially with different weight values, this can become too complex for a human, but not for an RL method.

A common way to incorporate efficiency into the reward is to penalise great state variations (Table 8.4), hopefully redirecting the RL methods towards learning to perform minimum state deviations to resolve conflicts. This is very helpful in environments with very low traffic density, where the RL method is not penalised by performing large state variations that move the ownship into the trajectory of neighbouring aircraft, thus creating secondary conflicts. In these cases, the only penalty for large trajectory changes is the one present in the reward. However, at high traffic densities, an RL method is expected to learn to perform small variations by choosing directions that do not result in secondary conflicts. Naturally, to prevent secondary conflicts, the RL method must be aware of the position of neighbouring aircraft.

GLOBAL REWARDS

An action that improves the current situation of the ownship, can still have a negative influence on global safety. For example, the conflict resolution manoeuvre performed by an ownship may force neighbouring aircraft to perform large state deviations, which may ultimately result in LoSs far from the ownship. In a single-agent RL, it is very difficult for an RL method to understand how an action taken at a particular time step affects the final outcome.

A possible solution to prevent actions that negatively affect global safety is to employ global rewards [233]. The objective of global rewards is for the RL method to favour actions that improve global safety. However, the RL method must then estimate how much the action of each aircraft contributes to the global reward. This is known as the credit assignment problem [268]. An action that leads to a higher final cumulative reward should have more value (credit), than an action that leads to a lower final reward. This is especially complicated when the final outcome of an action is only clear after several time steps (delayed reward). This is a difficult problem that requires further research. So far, studies have focused, for example, on methods that replace the original reward with a shaped reward that compares the reward received when that agent's action is replaced with a default action [269].

SECONDARY CONFLICTS

A consequence of resolving conflicts, which may not be clear to an RL method, is the creation of secondary conflicts when aircraft move into the path of nearby aircraft while changing their state to resolve a conflict. This is not studied when RL methods are trained only in pairwise conflicts. Nevertheless, it is an important factor if the RL method is to be tested in environment with higher traffic densities, where there is less free airpace.

Secondary conflicts are not always negative, studies have shown that more conflicts do not always necessarily mean more intrusions [162]. Sometimes these secondary conflicts are useful to create room and improve stability on a larger scale [15]. However, a solution that solves the current conflicts while leading to a smaller number of secondary conflicts is often preferable to one that creates a higher number of secondary conflicts. Therefore, it is beneficial to include this information in the design of the reward function. Any action by the RL method should be rewarded not only in terms of the resolution of the existent conflicts, but also on the final position of the ownship regarding nearby aircraft.

8.4.3. FALLING INTO LOCAL OPTIMA

A common issue with RL methods, similarly to many numerical methods, is the tendency for these to get stuck in local optima. This means that the method will not find the best possible solution (i.e., the global optimum) and, as a consequence, may not result in any improvement compared to current geometric CR methods, for example. There is a difficult balance between the rates of 'explore' or 'exploit'. On the one hand, the method should explore the solution space, or otherwise it may never find the global optimum. On the other hand, it should also exploit the known solutions to return a high reward.

Several works, such as [248, 270], mention the practical difficulties in implementing the theoretical high performance of RL methods. Ilyas [270] concludes that there is a significant gap between the theory that defines the current algorithms and the actual mechanisms driving their performance. There is no single good mechanism that can be implemented to ensure that the RL methods find the most optimal actions. There are only a handful of possibilities with which researchers must experiment until they achieve a better result:

- Fine-tuning the hyperparameters such as the learning rate: finding a perfect combination of hyperparameter values can be quite challenging, as these are specific for each particular environment. The most common way of selecting values for hyperparameters is unfortunately through manual tuning. The learning rate, often considered the most important parameter, must be discovered through trial and error. However, as an indication, the learning rate value should be increased if the method converges too slowly, meaning that the updates to the weights in the network are too small, and should be decreased if there are a lot of oscillations throughout the training of the method.

- Learning rate annealing: a varying learning rate throughout the training of the RL method can help create a better balance between exploitation and exploration. The simplest implementations employ higher learning at the beginning of the training so that the method gains information over the entire action space. As training progresses, the learning rate is decreased, so that the method can exploit the learnt action space to achieve high rewards.

- Addition of noise to help with exploration: noise increases the selection of unknown actions to the current policy, which can increase the ability of the method to jump out of local optima. The amount of noise is, however, another value that must be manually tuned. Large amounts of noise may lead to policy instability. Low noise quantities may not benefit explorability enough.

- Increasing buffer size: the replay buffer should be large enough to contain a wide range of experiences, which benefits a stable behaviour. Only using the very-most recent data may lead the method to overfit on that data. However, a large experience replay also requires a lot of memory and might slow the training. Again, manual tuning is required to achieve a proper buffer size.

8.4.4. UTILISATION OF PRIOR KNOWLEDGE (PRE-TRAINING)

Due to exploration, the RL method is often slow in the early stage of training. In theory, pre-training can be carried out in order to decrease training time of the RL method, as

it would thus 'start' from learnt behaviour. However, in practise, this implementation is not trivial. Previous research [245], used ATC controllers' decisions to train a supervised learning method. However, this is a typical framework for supervised learning, which is different from reinforcement learning. However, this study raised interesting points. Both controllers used different strategies to resolve conflicts and prioritised different elements of the conflict geometry. Furthermore, both controllers were shown to have a defensive position, ensuring an average separation between aircraft of more than 8 NM. This is considerably more than the predefined separation of 5 NM, leading to larger state deviations than necessary. This shows that it is not clear which tactics should be used as a reference.

More recently, Zhao [21] used the solution space diagram (SSD) method to form an image that is then input into a convolutional network. The embedded physics knowledge shows significantly improved scalability of the RL method. In previous work [271], we showed that pre-training of a DDPG method with actions performed by a geometric CR method can help the RL method start exploration with better decisions. Nevertheless, this may also have negative results, such as putting the RL method into a local optima from which it does not improve, or even limiting the probability of the method reaching new solutions that are not included in current procedures.

8.5. Testing of the Reinforcement Learning Method

This section describes potential issues when testing an RL method. Overfitting of the training scenarios, or limited training cases, can hinder the ability of the RL method to generalise to unseen situations. In this case, larger or more diverse training scenarios are necessary. Nevertheless, the limitations inputted into the method for faster training, i.e. small state and action formulations, may have limited the capacity of the method to successfully prevent conflicts in more complex scenarios. Finally, there is the need for a benchmark of known scenarios for explainability of the actions produced by the method.

8.5.1. Generalisation to Multiple Traffic Scenarios

At high traffic densities, each aircraft will face multiple conflict situations with the number of intruders and their positions varying greatly. It is often hard to train an RL method in a number of scenarios large enough to generalise well to different conflict geometries. Throughout this thesis, training in a specific traffic density led to models that proved inefficient at higher densities. The RL method should at least be trained at the highest traffic density that is expected in actual operations. It may also be that different traffic densities require different resolution strategies, as hypothesised in the Metropolis project [13]. In this case, the RL method must learn different responses according to the complexity of emergent behaviour resulting from increasing traffic densities.

Another possibility is that the limitations often set in the state and action formulations, to ensure fast training and convergence to optimal values, limits the ability of the RL method to generalise to more complex conflict geometries. The first issue is the reduction of the number of aircraft represented in the state formulation to a minimum value considered necessary in the training environment (see Table 8.3). At higher traffic densities, the method may not have enough information to resolve all conflicts, as all intruders may

not be represented in the state formulation. Additionally, without enough information on neighbouring aircraft, the RL method is not able to evaluate when a conflict resolution manoeuvre will result in secondary conflicts.

A solution may be to introduce lifelong learning and to have the RL method retrain itself when faced with unseen data. Isufah [241], for example, first trained an MA-RL agent in multi-actor conflicts with 3 aircraft, and then re-trained the method in multi-actor conflicts with 4 aircraft. The authors concluded that retraining of the method refines the policies learnt in the previous setting, while the new agent (the fourth agent) learnt a desirable policy.

8.5.2. VERIFICATION OF SOLUTIONS

Reinforcement learning applications are often a 'black-box', which does not inspire confidence. Their behaviour must be interpretable and traceable if they are to be certified. It is not trivial to rationalise the actions of an RL method, especially since these result from the combination of multiple factors in the state formulation. Several works, such as Zhao [21] and Li [236], show the final path of the aircraft when conflict resolution is carried out using the RL method. Nevertheless, these are for a limited set of conflict geometries tested.

There is a need for unified testing ATC scenarios with a multitude of different pairwise and multi-actor conflict geometries, so that the different resolution decisions by each RL method can be directly compared. Analysing the different solutions found by the methods, from simple pairwise to complex multi-actor conflict geometries, can help provide a base from which results can be examined and trusted upon. A known benchmark set of testing scenarios can lead to the creation of automatic tools for the analysis of the RL decisions for this set.

8.6. SUGGESTIONS FOR FUTURE RESEARCH

The following are proposed research directions and suggestions for future work:

- It is recommended to develop a library of reference traffic scenarios, containing a multitude of conflict scenarios for different use cases. The different state, action, and reward formulations could then be directly compared, as it is not yet clear which formulations optimise the behaviour of an RL method for conflict resolution. Reference traffic scenarios would also help to make the results of simulation studies more transparent and accessible.

- Deeper study of the best state formulations: RL methods for conflict resolution still show limitations in terms of generalising to more complex conflict geometries. This is often due to an oversimplification of the information available in the state formulation. Nevertheless, it is also true that geometric CR states do not consider all aircraft. Far away aircraft are dismissed as there is a high level of uncertainty regarding their future trajectory. On the one hand, there must be a balance between providing information on enough nearby aircraft, so that the RL method can resolve conflicts while avoiding secondary conflicts. On the other hand, the state formulation should not consider aircraft far away that may generate a false alarm and should limit the state formulation to a size that does not exponentially increase

      training time. More research is needed on this topic.

- Common evaluators: this chapter showed an example of studies that evaluate RL methods in different ways (see Tables 8.1 to 8.4). Different numbers of intruders are considered. Some studies only consider pairwise conflicts, and different safety and efficiency preferences are implemented. Having common evaluators can help establish an universally accepted evaluation system, so that different RL implementations for conflict resolution can be directly compared. Finally, it can also help create common reward formulations.

8.7. FINAL NOTES

Looking at the existing body of work, there is a significant gap between the theory that inspires the current algorithms and the actual result of the implementation. Simplification of state and actions formulations to limit training and convergence time, as well as the complexity of implementing methods that can handle the non-stationarity of a multi-aircraft environment, still limits the potential of RL methods to generalise their policies to real-world scenarios. More studies must be performed on RL methods capable of dealing with multi-actor conflict geometries, and finding solutions that do not result in a great number of secondary conflicts.

At present, the explainability of deep RL algorithms in conflict resolution remains to be explored. New guidelines and tools must be developed to better understand the decisions taken by RL methods. One way could be to develop a set of unified ATC scenarios, where a multitude of different pairwise and multi-actor conflict geometries can be tested. Finally, it is likely that researchers will find that (currently) RL methods do not generalise correctly for every conflict geometry. However, this is an important stepping stone for research.

8

9

DISCUSSION AND RECOMMENDATIONS

High traffic densities, as expected in future unmanned aviation operations, increase the likelihood of aircraft encountering multi-actor conflict situations where they have to coordinate their own actions with those of neighbouring aircraft to successfully avoid getting too close to each other. However, successive conflict resolution manoeuvres can lead to traffic patterns with a negative effect on the global safety. Knock-on effects of intruders avoiding each other may result in unexpected trajectory changes. It is practically impossible to know which actions are more efficient in coordination with the future unknown actions from multiple other aircraft. This challenge was formulated as follows:

<div style="background:gray">

Primary Research Objective

</div>

Investigate whether reinforcement learning applications can improve aircraft self-separation efficacy at higher traffic densities, with an emphasis on employing airspace designs and approaches applicable to future unmanned operations.

To meet the previous research object, this thesis covered the usage of several reinforcement learning approaches to decrease conflict rate and severity, in a high density aviation environment. This chapter provides a comprehensive discussion of all these approaches. In addition, recommendations for presented for future work are presented.

9.1. DISCUSSION

The discussion of the results of this thesis is divided into the following subsections.

9.1.1. TRANSITIONING FROM MANNED TO UNMANNED AVIATION

To be able to improve the current performance of self-separation methods, these were first evaluated and analysed. Chapter 2 directly compares the performance of commonly used CD&R methods both in manned and unmanned aviation operational environments. The results showed that the approaches that reduce the total number of conflicts and losses of minimum separation are common to both types of aviation. Geometric CR methods that opt for the 'shortest-way-out' tend to resolve conflicts more efficiently. At high traffic densities, resolutions that require aircraft occupying a larger portion of aircraft will likely result in an increased number of conflicts, as it leads to crossing the path of more aircraft.

Within geometric conflict resolutions, a direct comparison was made between methods that resolve pairwise conflicts (i.e., the Modified Voltage Potential (MVP) [15]) and methods that resolve all existing conflict simultaneously (i.e., the Solution Space Diagram (SDD) [42]). The results show that, although resolving all conflicts simultaneously offers faster conflict resolution and less time in conflict, it has the disadvantage of the solution space becoming easily saturated in conflicts with multiple aircraft. In the worst-case scenario, no solution might be available.

9.1.2. REDUCING CONFLICT RATE AND SEVERITY *(Answer to RQ1/RQ2)*

Chapters 3 and 4 focus on the re-evaluation of coordination elements to decrease conflict rate and severity. Speed heterogeneity between neighbouring aircraft is one of the main causes for increased conflict rate and probability. In Chapter 3, a reinforcement learning (RL) method was used to set velocity speed limits in areas with vertical deviations between layers. The results show that speed control during merging actions helps reduce the likelihood of aircraft meeting in conflict.

However, safety during a merging manoeuvre is dictated not only by the direct conflicts and intrusions suffered by the merging aircraft. Often the ownship remains at a safe distance from the follower and leader aircraft, as these alter their speed to avoid getting too close to the ownship. This creates a chain reaction in which aircraft decelerate to avoid getting too close to the leader aircraft. Ultimately, this may result in conflicts and intrusions far from the ownship, which increase instability in the airspace. In Chapter 4, an RL method is responsible for deciding when to perform a lane change manoeuvre. The results show that delaying merging manoeuvres until a safe distance gap is reached, between the aircraft and the leader and follower aircraft in the target layer, significantly reduces the severity of intrusions during a lane change procedure. However, it may also delay the dispersion of the local traffic density.

USAGE OF INTENT INFORMATION TO DECREASE CONFLICT RATE

Intent sharing between aircraft was explored in Chapter 3. Adding explicit intent improves conflict detection, as aircraft are informed, in advance, of future conflicts resulting from future state changes, such as turns, and can better avoid them. However, resolution manoeuvres cannot be calculated on the basis of intention information alone. As aircraft

opt for a 'shortest-way-out' conflict resolution, the current state projection must also be used in conflict detection, so the aircraft can prepare in advance for situations where intruders 'miss' trajectory changes points and instead remain close to their current state for conflict resolution. However, a disadvantage of using both intent and state information simultaneously is that the solution space becomes saturated faster, especially as the traffic density increases.

Moreover, the cost of implementing intent information must be carefully considered. Delays in data transmission and processing can delay the reaction to state changes in neighbouring aircraft. Second, the effect on safety is directly associated with the number of aircraft that can share and analyse intent information. To achieve the desired improvement, the majority of aircraft in the airspace would require this capability.

9.1.3. CONFLICT RESOLUTION WITH RL *(Answer to RQ4)*

Regarding the usage of RL methods to improve conflict resolution, we make the distinction between two different approaches. The first are model-free approaches, where the RL method is fully responsible for defining resolution manoeuvres to be taken by all aircraft to safe keep a minimum distance from each other. Second, we also define 'hybrid approaches' that combine RL approaches and existing CR algorithms. The former are used to improve the efficacy of the latter. Rather than depend on hard-code rules, laboriously designed by domain experts, RL methods can be used to set values by detecting complex patterns on the data. In this work, we define 'hybrid approaches' where RL methods are used dynamically defining the parameters that it uses to calculate resolution manoeuvres.

Chapters 6 and 7 explored the previous two options. We argue that at the current state-of-the-art, reinforcement learning cannot outperform known conflict resolution geometric methods. These employ geometrically calculated shortest-way manoeuvres that guarantee implicit coordination with minimum deviation. The RL methods developed in this thesis did not achieve this level of efficiency and coordination. However, conflict resolution algorithms often have predefined simple rules (e.g., one predefined look-ahead time, one predefined manoeuvre type). Reinforcement learning can instead create a much larger set of rules, adapted to a multitude of different conflict situations.

Moreover, reinforcement learning methods can be used to improve the behaviour in situations for which researchers do not have a clear guideline (e.g., return to the nominal path after conflict resolution, prioritisation of intruders, or deconflicting manoeuvres). Chapter 6 presents an RL method which was able to further reduce LoSs at a low traffic density, when compared to an CR geometrical method, by defending in advance against non-conflicting nearby aircraft, and initiating early resolution manoeuvres for head-on conflicts. Moreover, research resulting from this thesis showed that pre-emptive subliminal modifications to the ownship's state, before it enters into conflict with other aircraft, can decrease the severity of a conflict situation [239]. These actions should not replicate the behaviour of the conflict resolution method; these should be subliminal changes that do not lead to conflict chain reactions. For instance, the ownship may change its speed/heading slightly so that fewer intruders are included in a future conflict situation. It is not yet clear how to better perform these preventive actions, but reinforcement learning can be a useful tool to research an answer to this question.

REWARD FORMULATION FOR CONFLICT RESOLUTION

With conflict resolution, the objective is to have decisions that increase safety. Safety must then be quantified into absolute values. In this thesis, the method was primarily rewarded on the basis of the number of LoSs. Nevertheless, other forms of reward are worthy of study, especially when the scarcity of LoSs leads to a slow evolving method. For example, the number of conflicts is often also used in the reward formulation. However, these should have a considerably smaller weight than LoSs. Additionally, penalising conflicts may also reduce the potential benefits of conflicts. For example, with MVP, it has been shown that secondary conflicts can be beneficial, as they cause a redistribution of the traffic. This creates space for new resolution manoeuvres, which were not apparent before [15]. LoS severity may also be considered to put emphasis into preventing severe LoSs, or even avoiding state changes by suffering non-severe LoSs with a negligible risk.

Furthermore, on the subject of scarcity of LoSs, using global rewards (i.e., LoSs suffered by all aircraft in a given amount of time, not just the ones suffered by the ownship who performed the action), can help provide more information to the method. A global LoS reward can additionally favour coordinated actions that contribute to global safety.

9.1.4. IMPACT OF AIRSPACE STRUCTURE ON SAFETY *(Answer to RQ3)*

The simulated airspace was divided according to the layered airspace concept, as researched by the Metropolis project [13]. Optimal segmentation of aircraft per the available space helps increase the distance between cruising aircraft. Additionally, aircraft are limited to speed and altitude variation for conflict resolution, to avoid crossing the barriers of the surrounding urban infrastructure. In Chapter 4, an RL method is responsible for layer change decision making, selecting which layer an aircraft should move into based on current traffic in each layer, and the aircraft's distance to the next turn. The results show that aircraft should be distributed per layer per distance to the next turn. Aircraft closer to a turn are placed on an outward layer, already closer to their next layer.

In Chapter 5, reinforcement learning is used to find the optimal heading ranges per vertical layer, given the expected traffic scenario. Optimal segmentation of aircraft through the airspace helps reduce the number and severity of conflicts. The likelihood of aircraft meeting in conflict can only be reduced when aircraft are fully dispersed per the airspace. Expecting a uniform heading distribution and setting the airspace structure in accordance will not result in any meaningful segmentation when all aircraft predominantly adopt one direction.

Additionally, given the differences in the topology of different constrained environments, travelling in one specific direction may potentially result in more conflicts than travelling in other directions. Having humans create navigation rules for every different operational environment is impracticable time wise. However, a reinforcement learning method can be trained in each environment where operations are expected to be set, shaping its policy according to the requirements.

AUXILIARY VERTICAL LAYERS

The results show that the use of the auxiliary layers allows for a more complex segmentation of aircraft, which can improve airspace safety. Chapters 3 to 5 resort to 'auxiliar' vertical layers to decrease speed heterogeneity. Having the aircraft slowing down to re-

spect the maximum turn radius, at the same altitude that other aircraft are cruising in, would cause back-end conflicts with aircraft following the ownship suddenly slowing in order collision. Second, vertical space was reserved to allow vertical manoeuvres for conflict resolution.

Naturally, the introduction of intermediate layers is conditioned by the amount of vertical space that aircraft are allowed to operate in. In the event that only very few vertical layers are possible in the available airspace, priority should be given to allow for more traffic layers so as to improve the segmentation and alignment of traffic. However, in a less limited airspace where several layers are possible, having intermediate layers reduces the number of conflicts and intrusions.

9.1.5. ADDITIONAL CONSIDERATIONS

SIMULATED TRAFFIC SCENARIOS

In this thesis, commonly used aircraft models were used with their respective performance limits. The work presented herein was not extended to other models with considerably different speeds/acceleration limits. However, all designs/models are capable of being executed with different performance values. Due to their generic nature, all airspace designs and safety methods discussed in this thesis can be generalised beyond the specific conditions that have been considered here.

Furthermore, it was assumed that all aircraft applied the same conflict detection and resolution algorithm. When aircraft do not act towards defending again others (e.g., because they do not have the technology to sense and avoid other aircraft), different safety levels can be expected when compared to the results herein presented. Additionally, different speed/acceleration limits will naturally affect the efficiency and success of resolution manoeuvres.

DECENTRALISED VS CENTRALISED AIRSPACE

The advantages and disadvantages of both control methods (decentralised and centralised), and which one should be implemented, is still an on-going discussion. This thesis is not meant as a direct argument pro the implementation of one control method versus the other. From existing research, it is clear that both methods have disadvantages and advantages and should be adopted based on the operational environment. For a review of these advantages/disadvantages, the reader should refer to Chapter 2.

However, this thesis focused on how safety values can be improved in decentralised control, due to the reduced research in this area compared to centralised control. Additionally, this type of control becomes even more worthy of research when considering that it may enable the high traffic densities expected for future operations, where the number of agents would lead to a slow centralised environment. However, there are some doubts on whether the same level of coordination/safety of centralised control can be achieved, due to lower coordination and increased uncertainty regarding intruders' movements. The main takeaway from this thesis on this matter should be that these disadvantages can be considerably reduced through airspace design/procedures, and conflict detection and resolution methods that reduce conflict probability/rate.

EXTENSION TO OTHER OPERATIONAL ENVIRONMENTS

Due to the empirical nature of the results, the conclusions drawn in this thesis are, to some degree, sensitive to the parameter settings of the simulated airspace. However, the same methods can be adapted to different environments. First, the detection and resolution algorithms employed are independent of the environment; the only limitation is the number of degrees of freedom the aircraft is allowed to use to avoid conflicts (e.g., in some operational environments, heading changes might not be possible). Second, the reinforcement learning methods used can be trained in most environments and will adapt to its characteristics. Even applying the same methods to a centralised control environment would not require significant changes: the RL method could be set in the centralised system, receiving and transmitting information to all aircraft.

9.2. RECOMMENDATIONS FOR FUTURE WORK

This section describes opportunities for future research. The expected social impact of this work is discussed briefly.

9.2.1. REINFORCEMENT LEARNING APPROACHES IN AVIATION

Whether and how to use RL in aviation is an ongoing debate. Although there may be consensus that it can push the boundaries of human knowledge and applications, is it also considered that current measures and human policies should not be directly replaced by fully computational methods. This is due both to the possible risks of this implementation (i.e., there can be harmful decisions taken by this method in new situations), and the need for more research that can be compared and analysed. The decision on which areas of aviation can benefit from the introduction of this new tool is still at an early stage.

The experimental results developed in this thesis showed that reinforcement learning is optimal when offering solutions to humans in terms of expanding on a current set of rules or providing feedback on the best actions in a multi-agent environment. This thesis aimed to assist the introduction of reinforcement learning into conflict detection and resolution. However, there are still many applications to explore. For example, reinforcement learning can be used to reduce uncertainties resulting from weather factors on conflict detection, to improve upon the prioritisation of intruders when resolving conflicts, or to aid conflict prevention behaviour before a conflict situation.

Finally, no reinforcement learning application can be blindly implemented in the real world. Not only must it be further tested with different traffic densities and trajectories, improving its ability to generalise, but also examination is necessary into the choices made by the method. Additionally, safeguards must be implemented for potential bad decisions when applied to traffic densities and trajectories not previously seen.

CERTIFICATION REINFORCEMENT LEARNING METHODS

In the future RL methods will move from a research focus to a commercial application focus. This thesis proposes placing these methods at the heart of the most important task in aviation: ensuring safety. There are many challenges related to the trust and certification of these safety-critical systems. How certification can be achieved will be the focus of several upcoming studies. However, at this point, we can already establish the main focus points: *robustness, uncertainty, explainable, verification* [272].

First, RL methods must be proven resilient against unexpected or corner cases situations. Such requires extensive testing, where other machine learning methods can potentially be used to enlarge the number of training cases to which the method is exposed. Second, uncertainty analysis must be part of the testing process, as it evaluates the ability of a method to detect unknown situations that are outside its normal range of operation. Finally, safeguards should also be implemented. These can be an application on top of the method that looks for potentially harmful actions.

Reinforcement learning applications are often a 'black-box'; however, this setting does not inspire confidence. Their behaviour must be interpretable and traceable if they are to be certified. Several works have tried to do so by performing, for example, feature importance [273]. However, research is still far from having guidelines on how explanability can be achieved; focus should be put on this task in the future.

9.2.2. Safety Definition

This thesis focused on the number of conflicts and losses of minimum separation as a basis for safety. However, this is not the only definition, as it depends on the objectives of the operational environment. Military applications, for example, may have a more complex or altogether different definition of safety. Intrusion severity may be prioritised over the number of intrusions. In a real-world scenario, factors such as proximity to restricted areas or battery usage may be given preference.

Given the generic nature of the RL methods used, these can be extended to different safety formulations. Elements can be added to the reward given to the method, guiding its actions towards optimisation of other factors as well. Nevertheless, it should be taken into account that joining elements together in one reward formulation can be counter productive. First, a decision must be made about their weight, which defines their priority. Second, improving one element may result in weakening the other. For example, favouring elements such as flight time/path can have a negative impact on safety, as aircraft opt for limiting their deconflicting manoeuvres. In some cases, entering a conflict or loss of minimum separation situation may be more cost-efficient than a sizeable change in trajectory. Therefore, the combination of several elements in a safety formulation requires careful adjustment.

9.2.3. Dynamic Airspace Structuring

When the traffic scenario is not static over time, airspace structures should also not remain static over time. To maximise the efficiency with which the available airspace is used, the constraints imposed by the airspace design should match the changes in the traffic demand pattern that occur throughout the day. Nevertheless, a natural follow-up question is how the safety of operations can be guaranteed during configuration changes, and when and what configurations should be selected. This research is already being conducted at TUDelft.

9.2.4. Influence of Weather on Safety

This thesis only considered ideal weather conditions (e.g., no wind, rain, no performance deterioration resulting from extreme temperature levels). Although weather effects were not included in the results, it must be noted that these can affect operations in the real

word. The effect of weather on traffic flows is a complex topic, and it is the subject of many ongoing studies. Bad weather, and strong winds in particular, can severely reduce aircraft manoeuvrability, and decrease the set of possible manoeuvres for conflict resolution, affecting the safety of the airspace. Additionally, extreme weather conditions may also limit data communication/transfer. An analysis of these consequences, per type and intensity of the weather conditions, is valuable when evaluating the operational constraints affecting decentralised control.

9.2.5. IMPACT ON SOCIAL ACCEPTANCE OF URBAN AIR MOBILITY

Drones raise high economic expectations due to their capabilities and commercial applications. However, their social acceptance is the key to the complete development of their technological potential. Social acceptance can be considered to be based on the balance between the benefits and inconveniences resulting from the usage of such technology. In aviation, this balance is also conditioned by the safety policies and regulations of the airspace and current airspace users [274, 275]. These policies and regulations ensure that safety is within acceptable levels. This thesis focused on finding new tools/designs that help improve safety up to these levels.

Recent publications by SESAR [197] have highlighted the need for strategic and tactical conflict resolution. It is agreed that high traffic densities and/or heterogeneous aircraft types may require aircraft to be equipped with conflict detection and resolution technologies. Moreover, special attention is given to the development of reinforcement learning techniques capable of aiding the monitoring of trajectory deviation, and tactical conflict detection and response. Future work should focus on these areas to increase compliance with safety limits and improve trust in urban air mobility.

9.2.6. IMPACT ON THE ENVIRONMENT

A gap in this thesis is the lack of research on the environmental impact of the selected approaches. This is in itself a field of study, taking into account not only the capabilities of multiple drone models, but also the large differences between deployment scenarios. Regarding the specific scenarios explored in this thesis, future work should explore energy consumption, and resulting environmental impact of climbing, descending, and allocating aircraft to sub-optimal altitudes. Fortunately, drones are (mostly) a fully electric transportation technology, and thus these manoeuvres are not as impacting as when compared to manned aviation.

There are also important questions regarding the noise and impact on wild life that arise from the introduction of drones, which should be compared with the impact of the modes of transportation that drones are replacing. Furthermore, U-Space management is essential for safety control and reduction of the effects of drone traffic on the population and the environment. A sustainability assessment of supplementary infrastructures (e.g., charging, storage, and control stations) must also be compared with potential reductions in traffic congestion and savings in road infrastructure.

9.2.7. OPEN-SOURCE SOFTWARE

Fast-time simulations have been used in all technical chapters of this thesis. The conclusions drawn using this approach are, to some extent, dependent on the traffic scenarios

considered. However, despite the popularity of fast-time simulations, methods for generating standardised traffic scenarios have not yet been researched. This makes it very difficult to compare the results of similar studies that have used fast-time simulations.

To overcome this issue, it is recommended to develop a library of reference traffic scenarios, which contains a multitude of scenarios for different use cases. By making this library open-source, these scenarios can be continuously updated as the needs of the research community change, for example, when new aircraft types are introduced or new airspace structures developed. Reference traffic scenarios would also help to make the results of simulation studies more transparent and accessible.

9

10

CONCLUSIONS

Based on the results presented in the preceding chapters, the following final conclusions are drawn:

ON CONFLICT DETECTION & RESOLUTION METHODS:

- Geometric conflict resolution methods that opt for the 'shortest-way-out' result in the fewest LoSs. At high traffic densities, larger resolution manoeuvres that require aircraft to occupy a larger portion of the airspace are more likely to result in conflict chain reactions.

- Attempting to resolve all conflicts simultaneously, although resulting in less time in conflict, has the disadvantage of the solution space becoming easily saturated in conflicts with multiple agents.

REDUCING CONFLICT RATE AND SEVERITY ON A LAYERED AIRSPACE:

- Speed control reduces the likelihood of aircraft meeting in conflict. Setting velocity speed limits in areas where aircraft vertical deviate between traffic layers, helps increase distance between the merging and cruising aircraft in the target layer.

- Delaying a merging manoeuvre until a safe distance gap exits between the merging, and the leader and follower aircraft, can reduce LoS severity. However, it may also delay the dispersion of the local traffic density. The latter facilitates future merging manoeuvres.

ON MERGING CONFLICTS:

- Aircraft should be distributed per layer according to their distance to the next turn. Aircraft closer to a turn should be placed on an outward layer, already closer to their next target layer.

- A merging action cannot be evaluated solely on the basis of the conflicts suffered by the merging aircraft. It may create a succession of aircraft reducing their speed to avoid getting too close to the leader aircraft, leading to a conflict chain reaction.

ON AIRSPACE STRUCTURE:

- For aircraft to be truly fully segmented per the available airspace, the airspace structure must set in respect to the expected traffic scenario, the operational environment, and applied conflict and detection method.

- In constrained environments, auxiliary vertical layers should be used to reduce speed heterogeneity between cruising aircraft and aircraft decelerating before a turn. Aircraft can then slow down and turn outside of the main cruising layer.

ON REINFORCEMENT LEARNING TECHNIQUES:

- The best usage for reinforcement learning (RL) is 'hybrid approaches', where RL is used to enhance the behaviour of conflict detection and resolution algorithms. The latter often have a predefined simple set of rules. RL can create a much larger set of rules adapted to a multitude of conflict situations and geometries.

- Reinforcement learning can also be used to improve the behaviour in situations for which researchers do not have a clear guideline (e.g., return to the nominal path after conflict resolution, prioritisation of intruders, or deconflicting manoeuvres).

10

REFERENCES

[1] *Celebrating 75 Years of Federal Air Traffic Control By: Theresa L. Kraus, Ph.D. Agency Historian, Federal Aviation Administration,* **6**, 4 (1941).

[2] R. v. G. J.M. Hoekstra, R.C.J. Ruigrok, *Free Flight in a crowded Airspace?* in *FAA/Eurocontrol 3rd USA/Europe Air Traffic Management R&D Seminar* (2000).

[3] SESAR, *European ATM Master Plan : Roadmap for the safe integration of drones into all classes of airspace,* (2018).

[4] V. N. Duong, *FREER: Free-Route Experimental Encounter Resolution - Initial Results,* Tech. Rep. (EURO-CONTROL Experimental Centre BP 15, F.91222 Bretigny-sur-Orge, France, 1997).

[5] I. Wilson, *PHARE Advanced Tools Project Final Report,* Tech. Rep. (DOC 98-70-18, Eurocontrol, 96 rue de la Fusée, Brussels, Belgium, 1999).

[6] A. Barff, *Summary of the results of the mediterranean free flight (mff) programme,* Air Traffic Control Quarterly **15**, 119 (2007).

[7] *Nasa langley and NLR research of distributed Air/Ground Traffic Management,* in *AIAA's Aircraft Technology, Integration, and Operations (ATIO) 2002 Technical Forum* (2002).

[8] Joint Planning and Development Office, Next Generation Air Transportation System (NextGen), *Concept of operations for the next generation air transportation system,* (2011).

[9] M. Eby, *A self-organizational approach for resolving air traffic conflicts,* The Lincoln Laboratory Journal (1994).

[10] Sesar Joint Undertaking, *European drones outlook study - Unlocking the value for Europe,* Tech. Rep. (Sesar Joint Undertaking, 2016).

[11] Congress of the United States of America, *FAA Modernization and Reform Act of 2012,* City , 63 (2012).

[12] *Rules of the Air Annex 2 to the Convention on International Civil Aviation International Civil Aviation Organization International Standards,* 10th ed. (2005).

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

[14] Y. I. Jenie, E.-J. van Kampen, C. C. de Visser, J. Ellerbroek, and J. M. Hoekstra, *Selective velocity obstacle method for deconflicting maneuvers applied to unmanned aerial vehicles,* Journal of Guidance, Control, and Dynamics **38**, 1140 (2015).

[15] J. Hoekstra, R. van Gent, and R. Ruigrok, *Designing for safety: the 'free flight' air traffic management concept,* Reliability Engineering & System Safety **75**, 215 (2002).

[16] B. Mirchevska, C. Pek, M. Werling, M. Althoff, and J. Boedecker, *High-level Decision Making for Safe and Reasonable Autonomous Lane Changing using Reinforcement Learning,* in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)* (IEEE, 2018) pp. 2156–2162.

[17] J. Wang, Q. Zhang, D. Zhao, and Y. Chen, *Lane Change Decision-making through Deep Reinforcement Learning with Rule-based Constraints,* Tech. Rep. (2019) arXiv:1904.00231v2 .

[18] D. M. Saxena, S. Bae, A. Nakhaei, K. Fujimura, and M. Likhachev, *Proceedings - IEEE International Conference on Robotics and Automation,* Tech. Rep. (2020) arXiv:1909.06710 .

[19] R. Isufaj, D. A. 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).

[20] M. Brittain and P. Wei, *Autonomous aircraft sequencing and separation with hierarchical deep reinforcement learning,* in *International conference for Research in Air Transportation* (2018).

[21] P. Zhao and Y. Liu, *Physics informed deep reinforcement learning for aircraft conflict resolution,* IEEE Transactions on Intelligent Transportation Systems , 1 (2021).

[22] M. Doole, J. Ellerbroek, and J. Hoekstra, *Urban airspace traffic density estimation,* Eighth SESAR Innovation Days (2018).

[23] J. Kuchar and L. Yang, *A review of conflict detection and resolution modeling methods,* IEEE Transactions on Intelligent Transportation Systems **1**, 179 (2000).

[24] Y. I. Jenie, E.-J. van Kampen, J. Ellerbroek, and J. M. Hoekstra, *Taxonomy of Conflict Detection and Resolution Approaches for Unmanned Aerial Vehicle in an Integrated Airspace,* IEEE Transactions on Intelligent Transportation Systems **18**, 558 (2017).

[25] J. M. Hoekstra and J. Ellerbroek, *BlueSky ATC simulator project: an open-data and open-source approach,* Proceedings of the 7th International Conference on Research in Air Transportation , 1 (2016).

[26] I. C. A. Organization, *ICAO Circular 328 - Unmanned Aircraft Systems (UAS),* Tech. Rep. (ICAO, 2011).

[27] L. C. Yang and J. K. Kuchar, *Using intent information in probabilistic conflict analysis,* in *1998 Guidance, Navigation, and Control Conference and Exhibit* (American Institute of Aeronautics and Astronautics Inc, AIAA, 1998) pp. 797–806.

[28] Inseok Hwang and Chze Eng Seah, *Intent-Based Probabilistic Conflict Detection for the Next Generation Air Transportation System,* Proceedings of the IEEE **96**, 2040 (2008).

[29] M. Porretta, W. Schuster, A. Majumdar, and W. Ochieng, *Strategic conflict detection and resolution using aircraft intent information,* Journal of Navigation **63**, 61 (2010).

[30] W. Liu and I. Hwang, *Probabilistic trajectory prediction and conflict detection for air traffic control,* Journal of Guidance, Control, and Dynamics **34**, 1779 (2011).

[31] R. Ruigrok and M. V. Clari, *The impact of aircraft intent information and traffic separation assurance responsibility on en-route airspace capacity,* in *Conference: 5th FAA/EUROCONTROL ATM R&D Seminar, Budapest* (2003).

[32] R. Ruigrok and J. Hoekstra, *Human factors evaluations of free flight issues solved and issues remaining,* Applied Ergonomics **38**, 437 (2007).

[33] K. Bilimoria, H. Lee, Z.-H. Mao, and E. Feron, *Comparison of centralized and decentralized conflict resolution strategies for multiple-aircraft problems,* in *18th Applied Aerodynamics Conference* (American Institute of Aeronautics and Astronautics, Reston, Virigina, 2000).

[34] N. Durand and N. Barnier, *Does ATM Need Centralized Coordination? Autonomous Conflict Resolution Analysis in a Constrained Speed Environment,* in *ATM seminar 2015, 11th USA/EUROPE Air Traffic Management R&D Seminar* (FAA & Eurocontrol, Lisboa, Portugal, 2015).

[35] F. Borrelli, D. Subramanian, A. Raghunathan, and L. Biegler, *MILP and NLP techniques for centralized trajectory planning of multiple unmanned air vehicles,* in *2006 American Control Conference* (IEEE, 2006).

[36] R. Mart and G. Reinelt, *The Linear Ordering Problem: Exact and Heuristic Methods in Combinatorial Optimization,* 1st ed. (Springer Publishing Company, Incorporated, 2011).

[37] A. Alonso-Ayuso, L. F. Escudero, F. J. Martin-Campo, and N. Mladenovic, *A VNS metaheuristic for solving the aircraft conflict detection and resolution problem by performing turn changes,* Journal of Global Optimization **63**, 583 (2014).

[38] H. Liu, F. Liu, X. Zhang, X. Guan, J. Chen, and P. Savinaud, *Aircraft conflict resolution method based on hybrid ant colony optimization and artificial potential field,* Science China Information Sciences **61** (2018), 10.1007/s11432-017-9310-5.

[39] A. Sathyan, N. Ernest, L. Lavigne, F. Cazaurang, M. Kumar, and K. Cohen, *A genetic fuzzy logic based approach to solving the aircraft conflict resolution problem,* in *AIAA Information Systems-AIAA Infotech @ Aerospace* (American Institute of Aeronautics and Astronautics, 2017).

[40] D. Rathbun, S. Kragelund, A. Pongpunwattana, and B. Capozzi, *An evolution based path planning algorithm for autonomous motion of a UAV through uncertain environments,* in *Proceedings. The 21st Digital Avionics Systems Conference* (IEEE, 2002).

[41] D. O. T. FAA, *Right-of-way rules: Except water operations, 14 cfr, pt. 91.113,* (2004).

[42] S. Balasooriyan, *Multi-aircraft Conflict Resolution using Velocity Obstacles,* Master's thesis, Delft University of Technology (2017).

[43] S. V. Dam, M. Mulder, and R. Paassen, *The use of intent information in an airborne self-separation assistance display design,* in *AIAA Guidance, Navigation, and Control Conference* (American Institute of Aeronautics and Astronautics, 2009).

[44] S. Balachandran, C. Munoz, and M. C. Consiglio, *Implicitly coordinated detect and avoid capability for safe autonomous operation of small UAS,* in *17th AIAA Aviation Technology, Integration, and Operations Conference* (American Institute of Aeronautics and Astronautics, 2017).

[45] J. Yang, D. Yin, Y. Niu, and L. Shen, *Distributed cooperative onboard planning for the conflict resolution of unmanned aerial vehicles,* Journal of Guidance, Control, and Dynamics **42**, 272 (2019).

[46] R. A. Klaus and T. W. McLain, *A radar-based, tree-branching sense and avoid system for small unmanned aircraft,* in *AIAA Guidance, Navigation, and Control (GNC) Conference* (American Institute of Aeronautics and Astronautics, 2013).

[47] R. Teo, J. S. Jang, and C. Tomlin, *Automated multiple UAV flight - the stanford DragonFly UAV program,* in *2004 43rd IEEE Conference on Decision and Control (CDC) (IEEE Cat. No.04CH37601)* (IEEE, 2004).

[48] Z.-H. Mao, D. Dugail, and E. Feron, *Space partition for conflict resolution of intersecting flows of mobile agents,* IEEE Transactions on Intelligent Transportation Systems **8**, 512 (2007).

[49] K. Treleaven and Z.-H. Mao, *Conflict resolution and traffic complexity of multiple intersecting flows of aircraft,* IEEE Transactions on Intelligent Transportation Systems **9**, 633 (2008).

[50] M. Christodoulou and S. Kodaxakis, *Automatic commercial aircraft-collision avoidance in free flight: The three-dimensional problem,* IEEE Transactions on Intelligent Transportation Systems **7**, 242 (2006).

[51] M. F. Lupu, E. Feron, and Z.-H. Mao, *Influence of aircraft maneuver preference variability on airspace usage,* IEEE Transactions on Intelligent Transportation Systems **12**, 1446 (2011).

[52] J. Hu, M. Prandini, and S. Sastry, *Optimal coordinated maneuvers for three-dimensional aircraft conflict resolution,* Journal of Guidance, Control, and Dynamics **25**, 888 (2002).

[53] L. Pallottino, E. Feron, and A. Bicchi, *Conflict resolution problems for air traffic management systems solved with mixed integer programming,* IEEE Transactions on Intelligent Transportation Systems **3**, 3 (2002).

[54] N. Archambault and N. Durand, *Scheduling heuristics for on-board sequential air conflict solving,* in *The 23rd Digital Avionics Systems Conference (IEEE Cat. No.04CH37576)*, Vol. 1 (2004) pp. 481–3.1.

[55] C. Lin, V. Nagarajan, R. Gupta, and B. Rajaram, *Efficient sequential consistency via conflict ordering,* in *Proceedings of the seventeenth international conference on Architectural Support for Programming Languages and Operating Systems*, ASPLOS '12 (ACM, 2012) pp. 273–286.

[56] I. Karatzas and S. E. Shreve, *Brownian motion,* in *Brownian Motion and Stochastic Calculus* (Springer New York, NY, 1998) pp. 47–127.

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

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

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

[60] E. Sunil, J. Ellerbroek, J. M. Hoekstra, and J. Maas, *Three-dimensional conflict count models for unstructured and layered airspace designs,* Transportation Research Part C: Emerging Technologies **95**, 295 (2018).

[61] G. Gawinowski, J.-L. Garcia, R. Guerreau, R. Weber, and M. Brochard, *ERASMUS: a new path for 4D trajectory-based enablers to reduce the traffic complexity,* in *2007 IEEE/AIAA 26th Digital Avionics Systems Conference* (IEEE, 2007).

[62] G. Chaloulos, E. Crück, and J. Lygeros, *A simulation based study of subliminal control for air traffic management,* Transportation Research Part C: Emerging Technologies **18**, 963 (2010), special issue on Transportation Simulation Advances in Air Transportation Research.

[63] D. Rey, C. Rapine, R. Fondacci, and N.-E. E. Faouzi, *Subliminal speed control in air traffic management: Optimization and simulation,* Transportation Science **50**, 240 (2016).

[64] L. H. Mutuel, P. Neri, and E. Paricaud, *Initial 4D Trajectory Management Concept Evaluation,* in *Tenth USA/Europe Air Traffic Management Research and Development Seminar (ATM2013)* (FAA & Eurocontrol, Chicago, Illinois, 2013).

[65] A. A. Lambregts, J. Tadema, R. M. Rademaker, and E. Theunissen, *Defining maximum safe maneuvering authority in 3d space required for autonomous integrated conflict resolution,* in *2009 IEEE/AIAA 28th Digital Avionics Systems Conference* (2009) pp. 5.C.1–1–5.C.1–17.

[66] E. E. Centre, *Base of Aircraft (BADA) Aircraft Performance Modelling Report,* Tech. Rep. EEC Technical/Scientific Report No. 2009-009 (EUROCONTROL, 2009).

[67] T. Dietrich, S. Krug, and A. Zimmermann, *An empirical study on generic multicopter energy consumption profiles,* 2017 Annual IEEE International Systems Conference (SysCon) , 1 (2017).

[68] J. K. Stolaroff, C. Samaras, E. R. O'Neill, A. Lubers, A. S. Mitchell, and D. Ceperley, *Energy use and life cycle*

greenhouse gas emissions of drones for commercial package delivery, Nature Communications **9** (2018), 10.1038/s41467-017-02411-5.

[69] D. Burgess, S. Altman, and M. L. Wood, *TCAS: Maneuvering aircraft in the horizontal plane,* Lincoln Lab. J **7** (1994).

[70] M. Kochenderfer, J. Holland, and J. Chryssanthacopoulos, *Next-generation airborne collision avoidance system,* Lincoln Laboratory Journal **19**, 242 (Jan 2012).

[71] A. Vink, S. Kauppinen, J. Beers, and K. de Jong, *Medium term conflict detection in EATCHIP phase III,* in *16th DASC. AIAA/IEEE Digital Avionics Systems Conference. Reflections to the Future. Proceedings* (IEEE, 1997).

[72] S. Cafieri and R. Omheni, *Mixed-integer nonlinear programming for aircraft conflict avoidance by sequentially applying velocity and heading angle changes,* European Journal of Operational Research **260**, 283 (2017).

[73] A. E. Vela, S. Solak, J.-P. B. Clarke, W. E. Singhose, E. R. Barnes, and E. L. Johnson, *Near real-time fuel-optimal en route conflict resolution,* IEEE Transactions on Intelligent Transportation Systems **11**, 826 (2010).

[74] W. Chen, J. Chen, Z. Shao, and L. T. Biegler, *Three-dimensional aircraft conflict resolution based on smoothing methods,* Journal of Guidance, Control, and Dynamics **39**, 1481 (2016).

[75] J. L. Ny and G. J. Pappas, *Geometric programming and mechanism design for air traffic conflict resolution,* in *Proceedings of the 2010 American Control Conference* (IEEE, 2010).

[76] W. Niedringhaus, *Stream option manager (SOM): automated integration of aircraft separation, merging, stream management, and other air traffic control functions,* IEEE Transactions on Systems, Man, and Cybernetics **25**, 1269 (1995).

[77] A. Alonso-Ayuso, L. F. Escudero, F. J. Martín-Campo, and N. Mladenović, *On the aircraft conflict resolution problem: A VNS approach in a multiobjective framework,* Electronic Notes in Discrete Mathematics **58**, 151 (2017).

[78] N. Durand, J.-M. Alliot, and O. Chansou, *Optimal resolution of en route conflicts,* Air Traffic Control Quarterly **3**, 139 (1995).

[79] Y. Yang, J. Zhang, K.-Q. Cai, and M. Prandini, *Multi-aircraft conflict detection and resolution based on probabilistic reach sets,* IEEE Transactions on Control Systems Technology **25**, 309 (2017).

[80] Y. Yang, J. Zhang, K. quan Cai, and M. Prandini, *A stochastic reachability analysis approach to aircraft conflict detection and resolution,* in *2014 IEEE Conference on Control Applications (CCA)* (IEEE, 2014).

[81] C. Allignol, N. Barnier, N. Durand, and J.-M. Alliot, *A new framework for solving en route conflicts,* Air Traffic Control Quarterly **21**, 233 (2013).

[82] C. Tomlin, I. Mitchell, and R. Ghosh, *Safety verification of conflict resolution manoeuvres,* IEEE Transactions on Intelligent Transportation Systems **2**, 110 (2001).

[83] A. L. Visintini, W. Glover, J. Lygeros, and J. Maciejowski, *Monte carlo optimization for conflict resolution in air traffic control,* IEEE Transactions on Intelligent Transportation Systems **7**, 470 (2006).

[84] M. Prandini, J. Lygeros, A. Nilim, and S. Sastry, *A probabilistic framework for aircraft conflict detection,* in *Guidance, Navigation, and Control Conference and Exhibit* (American Institute of Aeronautics and Astronautics, 1999).

[85] S. Hao, S. Cheng, and Y. Zhang, *A multi-aircraft conflict detection and resolution method for 4-dimensional trajectory-based operation,* Chinese Journal of Aeronautics **31**, 1579 (2018).

[86] R. Chipalkatty, P. Twu, A. R. Rahmani, and M. Egerstedt, *Merging and spacing of heterogeneous aircraft in support of NextGen,* Journal of Guidance, Control, and Dynamics **35**, 1637 (2012).

[87] A. R. Pritchett and A. Genton, *Negotiated decentralized aircraft conflict resolution,* IEEE Transactions on Intelligent Transportation Systems **19**, 81 (2018).

[88] D. Sislak, P. Volf, and M. Pechoucek, *Agent-based cooperative decentralized airplane-collision avoidance,* IEEE Transactions on Intelligent Transportation Systems **12**, 36 (2011).

[89] K. Harper, S. Mulgund, S. Guarino, A. Mehta, and G. Zacharias, *Air traffic controller agent model for free flight,* in *Guidance, Navigation, and Control Conference and Exhibit* (American Institute of Aeronautics and Astronautics, 1999).

[90] E. Hoffman, F. Bonnans, K. Blin, and K. Zeghal, *Conflict resolution in presence of uncertainty - a case study of decision making with dynamic programming,* in *AIAA Guidance, Navigation, and Control Conference and Exhibit* (American Institute of Aeronautics and Astronautics, 2001).

[91] A. Bicchi and L. Pallottino, *On optimal cooperative conflict resolution for air traffic management systems,*

IEEE Transactions on Intelligent Transportation Systems **1**, 221 (2000).

[92] G. Granger, N. Durand, and J.-M. Alliot, *Token allocation strategy for free-flight conflict solving,* Proceedings of the Thirteenth Innovative Applications of Artificial Intelligence (2001).

[93] R. A. Paielli, *Modeling maneuver dynamics in air traffic conflict resolution,* Journal of Guidance, Control, and Dynamics **26**, 407 (2003).

[94] G. A. M. Velasco, C. Borst, J. Ellerbroek, M. M. van Paassen, and M. Mulder, *The use of intent information in conflict detection and resolution models based on dynamic velocity obstacles,* IEEE Transactions on Intelligent Transportation Systems **16**, 2297 (2015).

[95] S. Huang, E. Feron, G. Reed, and Z.-H. Mao, *Compact configuration of aircraft flows at intersections,* IEEE Transactions on Intelligent Transportation Systems **15**, 771 (2014).

[96] H. von Viebahn and J. Schiefele, *Method for detecting and avoiding flight hazards,* in *Enhanced and Synthetic Vision 1997*, edited by J. G. Verly (SPIE, 1997).

[97] S. Devasia, D. Iamratanakul, G. Chatterji, and G. Meyer, *Decoupled conflict-resolution procedures for decentralized air traffic control,* in *2009 IEEE International Conference on Control Applications* (IEEE, 2009).

[98] Y. Zhao, R. Schultz, Y. Zhao, and R. Schultz, *Deterministic resolution of two aircraft conflict in free flight,* in *Guidance, Navigation, and Control Conference* (American Institute of Aeronautics and Astronautics, 1997).

[99] Z.-H. Mao, E. Feron, and K. Bilimoria, *Stability and performance of intersecting aircraft flows under decentralized conflict avoidance rules,* IEEE Transactions on Intelligent Transportation Systems **2**, 101 (2001).

[100] K. Bilimoria, B. Sridhar, and G. Chatterji, *Effects of conflict resolution maneuvers and traffic density on free flight,* in *Guidance, Navigation, and Control Conference* (American Institute of Aeronautics and Astronautics, 1996).

[101] J. Krozel and M. Peters, *Conflict detection and resolution for free flight,* Air Traffic Control Quarterly **5**, 181 (1997).

[102] J. Zhang, J. Wu, and T. Song, *Study of multi-aircraft conflict resolution and algorithm optimization based on genetic algorithm,* in *Proceedings of the 2014 International Conference on Computer, Communications and Information Technology* (Atlantis Press, 2014).

[103] L. Peng and Y. Lin, *Study on the model for horizontal escape maneuvers in TCAS,* IEEE Transactions on Intelligent Transportation Systems **11**, 392 (2010).

[104] P. K. Menon, G. D. Sweriduk, and B. Sridhar, *Optimal strategies for free-flight air traffic conflict resolution,* Journal of Guidance, Control, and Dynamics **22**, 202 (1999).

[105] I. Burdun and O. Parfentyev, *AI knowledge model for self-organizing conflict prevention/resolution in close free-flight air space,* in *1999 IEEE Aerospace Conference. Proceedings (Cat. No.99TH8403)* (IEEE, 1999).

[106] R. B. Patel and P. J. Goulart, *Trajectory generation for aircraft avoidance maneuvers using online optimization,* Journal of Guidance, Control, and Dynamics **34**, 218 (2011).

[107] J. Alonso-Mora, T. Naegeli, R. Siegwart, and P. Beardsley, *Collision avoidance for aerial vehicles in multi-agent scenarios,* Autonomous Robots **39**, 101 (2015).

[108] A. Kelly, A. Stentz, O. Amidi, M. Bode, D. Bradley, A. Diaz-Calderon, M. Happold, H. Herman, R. Mandelbaum, T. Pilarski, P. Rander, S. Thayer, N. Vallidis, and R. Warner, *Toward reliable off road autonomous vehicles operating in challenging environments,* The International Journal of Robotics Research **25**, 449 (2006).

[109] H. Y. Ong and M. J. Kochenderfer, *Markov decision process-based distributed conflict resolution for drone air traffic management,* Journal of Guidance, Control, and Dynamics **40**, 69 (2017).

[110] R. Beard, T. McLain, M. Goodrich, and E. Anderson, *Coordinated target assignment and intercept for unmanned air vehicles,* IEEE Transactions on Robotics and Automation **18**, 911 (2002).

[111] I. Nikolos, K. Valavanis, N. Tsourveloudis, and A. Kostaras, *Evolutionary algorithm based offline/online path planner for uav navigation,* IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics) **33**, 898 (2003).

[112] F. Ho, R. Geraldes, A. Goncalves, M. Cavazza, and H. Prendinger, *Improved conflict detection and resolution for service UAVs in shared airspace,* IEEE Transactions on Vehicular Technology **68**, 1231 (2019).

[113] T. Liao, *UAV Collision Avoidance using A* Algorithm,* Master's thesis, Auburn University (2012).

[114] A. Richards and J. How, *Model predictive control of vehicle maneuvers with guaranteed completion time and robust feasibility,* in *Proceedings of the 2003 American Control Conference, 2003.* (IEEE, 2003).

[115] G. Fasano, D. Accardo, A. Moccia, C. Carbone, U. Ciniglio, F. Corraro, and S. Luongo, *Multi-sensor-based fully autonomous non-cooperative collision avoidance system for unmanned air vehicles,* Journal of Aerospace Computing, Information, and Communication **5**, 338 (2008).

[116] J. W. Langelaan, *State estimation for autonomous flight in cluttered environments,* Journal of Guidance, Control, and Dynamics **30**, 1414 (2007).

[117] K. J. Obermeyer, P. Oberlin, and S. Darbha, *Sampling-based path planning for a visual reconnaissance unmanned air vehicle,* Journal of Guidance, Control, and Dynamics **35**, 619 (2012).

[118] J.-W. Park, H.-D. Oh, and M.-J. Tahk, *UAV collision avoidance based on geometric approach,* in *2008 SICE Annual Conference* (IEEE, 2008).

[119] H. bin Duan, X. yin Zhang, J. Wu, and G. jun Ma, *Max-min adaptive ant colony optimization approach to multi-UAVs coordinated trajectory replanning in dynamic and uncertain environments,* Journal of Bionic Engineering **6**, 161 (2009).

[120] J. G. Manathara and D. Ghose, *Rendezvous of multiple UAVs with collision avoidance using consensus,* Journal of Aerospace Engineering **25**, 480 (2012).

[121] C. G. Prévost, A. Desbiens, E. Gagnon, and D. Hodouin, *Unmanned aerial vehicle optimal cooperative obstacle avoidance in a stochastic dynamic environment,* Journal of Guidance, Control, and Dynamics **34**, 29 (2011).

[122] A. Zeitlin and M. McLaughlin, *Safety of cooperative collision avoidance for unmanned aircraft,* IEEE Aerospace and Electronic Systems Magazine **22**, 9 (2007).

[123] J. Yang, D. Yin, Y. Niu, and L. Zhu, *Cooperative conflict detection and resolution of civil unmanned aerial vehicles in metropolis,* Advances in Mechanical Engineering **8** (2016), 10.1177/1687814016651195.

[124] A. Mujumdar and R. Padhi, *Reactive collision avoidance of using nonlinear geometric and differential geometric guidance,* Journal of Guidance, Control, and Dynamics **34**, 303 (2011).

[125] J. Leonard, A. Savvaris, and A. Tsourdos, *Distributed reactive collision avoidance for a swarm of quadrotors,* Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering **231**, 1035 (2016).

[126] H. Yang and Y. Zhao, *Trajectory planning for autonomous aerospace vehicles amid known obstacles and conflicts,* Journal of Guidance, Control, and Dynamics **27**, 997 (2004).

[127] C. Zhu, X. Liang, L. He, G. Xu, and Y. Li, *Conflict resolution of aircraft swarms based on interactive multi-model,* in *Proceedings of the 2017 International Conference on Artificial Intelligence, Automation and Control Technologies - AIACT '17* (ACM Press, 2017).

[128] I. Hwang, J. Kim, and C. Tomlin, *Protocol-based conflict resolution for air traffic control,* Air Traffic Control Quarterly **15**, 1 (2007).

[129] V. Jilkov, *An efficient algorithm for aircraft conflict detection and resolution using list viterbi algorithm,* 18th International Conference on Information Fusion , 1709 (July 6-9, 2015).

[130] R. Hurley, R. Lind, and J. Kehoe, *A torus based three dimensional motion planning model for very maneuverable micro air vehicles,* in *AIAA Guidance, Navigation, and Control Conference* (American Institute of Aeronautics and Astronautics, 2012).

[131] Y. Kitamura, T. Tanaka, F. Kishino, and M. Yachida, *3-d path planning in a dynamic environment using an octree and an artificial potential field,* in *Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots* (IEEE Comput. Soc. Press, 1995).

[132] S. Hrabar, *Reactive obstacle avoidance for rotorcraft UAVs,* in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems* (IEEE, 2011).

[133] D. Jung and P. Tsiotras, *On-line path generation for unmanned aerial vehicles using b-spline path templates,* Journal of Guidance, Control, and Dynamics **36**, 1642 (2013).

[134] L. Schmitt and W. Fichter, *Collision-avoidance framework for small fixed-wing unmanned aerial vehicles,* Journal of Guidance, Control, and Dynamics **37**, 1323 (2014).

[135] G. Chowdhary, D. M. Sobers, C. Pravitra, C. Christmann, A. Wu, H. Hashimoto, C. Ong, R. Kalghatgi, and E. N. Johnson, *Self-contained autonomous indoor flight with ranging sensor navigation,* Journal of Guidance, Control, and Dynamics **35**, 1843 (2012).

[136] A. Beyeler, J.-C. Zufferey, and D. Floreano, *Vision-based control of near-obstacle flight,* Autonomous Robots **27**, 201 (2009).

[137] G. de Croon, C. D. Wagter, B. Remes, and R. Ruijsink, *Sky segmentation approach to obstacle avoidance,* in *2011 Aerospace Conference* (IEEE, 2011).

[138] G. de Croon, E. de Weerdt, C. de Wagter, and B. Remes, *The appearance variation cue for obstacle*

avoidance, in *2010 IEEE International Conference on Robotics and Biomimetics* (IEEE, 2010).

[139] J. Muller, A. V. Ruiz, and I. Wieser, *Safe & sound: A robust collision avoidance layer for aerial robots based on acoustic sensors,* in *2014 IEEE/ION Position, Location and Navigation Symposium - PLANS 2014* (IEEE, 2014).

[140] J. Ellerbroek, ProfHoekstra, and MJRibeiroTUDelft, *Bluesky implementation: underlying the publication "Review of Conflict Resolution Methods for Manned and Unmanned Aviation,* (2020), 10.5281/zenodo.3863009.

[141] M. Ribeiro, J. Ellerbroek, and J. Hoekstra, *Bluesky data: underlying the publication "review of conflict resolution methods for manned and unmanned aviation". 4tu.centre for research data. dataset,* (2020), 10.4121/uuid:1a2bcb7f-192d-4c0c-9732-5d1ce005d0ee.

[142] E. Sunil, J. Ellerbroek, J. Hoekstra, A. Vidosavljevic, M. Arntzen, F. Bussink, and D. Nieuwenhuisen, *Analysis of airspace structure and capacity for decentralized separation using fast-time simulations,* Journal of Guidance, Control, and Dynamics **40**, 38 (2017).

[143] J. Sun, J. M. Hoekstra, and J. Ellerbroek, *Open Aircraft Performance Modeling: Based on an Analysis of Aircraft Surveillance Data,* Ph.D. thesis, Delft University of Technology (2019).

[144] M. Yousef, F. Iqbal, and M. Hussain, *Drone forensics: A detailed analysis of emerging dji models,* in *2020 11th International Conference on Information and Communication Systems (ICICS)* (2020) pp. 066–071.

[145] S. Dorafshan, M. Maguire, N. V. Hoffer, and C. Coopmans, *Challenges in bridge inspection using small unmanned aerial systems: Results and lessons learned,* in *2017 International Conference on Unmanned Aircraft Systems (ICUAS)* (2017) pp. 1722–1730.

[146] S. Świerczynski and A. Felski, *Determination of the position using receivers installed in uav,* in *2019 European Navigation Conference (ENC)* (2019) pp. 1–4.

[147] EUROCONTROL, *Performance Review Report An Assessment of Air Traffic Management in Europe during the Calendar Year 2018* (2018).

[148] P. Fiorini and Z. Shiller, *Motion planning in dynamic environments using velocity obstacles,* The International Journal of Robotics Research **17**, 760 (1998).

[149] A. Chakravarthy and D. Ghose, *Obstacle avoidance in a dynamic environment: a collision cone approach,* IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans **28**, 562 (1998).

[150] E. Haines, *Point in polygon strategies,* in *Graphics Gems IV* (Academic Press Professional, Inc., USA, 1994) p. 24–46.

[151] K. Bilimoria, K. Sheth, H. Lee, and S. Grabbe, *Performance evaluation of airborne separation assurance for free flight,* in *18th Applied Aerodynamics Conference* (American Institute of Aeronautics and Astronautics, Reston, Virigina, 2000).

[152] L. Piedade, *Aircraft Conflict Prioritization and Resolution using the Solution Space Diagram,* Master's thesis, Instituto Superior Tecnico (2018).

[153] E. Sunil, J. Ellerbroek, J. Hoekstra, and J. Maas, *Modeling airspace stability and capacity for decentralized separation,* in *12th USA/Europe Air Traffic Management R&D Seminar* (2017).

[154] E. Sunil, Olafur, J. Ellerbroek, and J. Hoekstra, *Analyzing the Effect of Traffic Scenario Properties on Conflict Count Models,* in *Conference: International Conference for Research on Air Transportation* (Barcelona, Spain, 2018).

[155] T. Rakha and A. Gorodetsky, *Review of Unmanned Aerial System (UAS) applications in the built environment: Towards automated building inspection procedures using drones,* Automation in Construction **93**, 252 (2018).

[156] J. A. Besada, I. Campana, L. Bergesio, A. M. Bernardos, and G. de Miguel, *Drone Flight Planning for Safe Urban Operations: UTM Requirements and Tools,* in *2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)* (IEEE, 2019) pp. 924–930.

[157] E. Walraven, M. T. Spaan, and B. Bakker, *Traffic flow optimization: A reinforcement learning approach,* Engineering Applications of Artificial Intelligence **52**, 203 (2016).

[158] Z. Li, P. Liu, C. Xu, H. Duan, and W. Wang, *Reinforcement Learning-Based Variable Speed Limit Control Strategy to Reduce Traffic Congestion at Freeway Recurrent Bottlenecks,* IEEE Transactions on Intelligent Transportation Systems **18**, 3204 (2017).

[159] A. K. Agogino and K. Tumer, *A multiagent approach to managing air traffic flow,* Autonomous Agents and Multi-Agent Systems **24**, 1 (2012).

[160] Y. Liu and X. R. Li, *Intent based trajectory prediction by multiple model prediction and smoothing,* in *AIAA Guidance, Navigation, and Control Conference* (2015) https://arc.aiaa.org/doi/pdf/10.2514/6.2015-1324 .

[161] J. d'Engelbronner, C. Borst, J. Ellerbroek, M. Van Paassen, and M. Mulder, *Solution-space–based analysis of dynamic air traffic controller workload,* Journal of Aircraft **52**, 1146 (2015).

[162] M. Ribeiro, J. Ellerbroek, and J. Hoekstra, *Review of conflict resolution methods for manned and unmanned aviation,* Aerospace **7** (2020), 10.3390/aerospace7060079.

[163] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, *Continuous control with deep reinforcement learning,* in *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings* (International Conference on Learning Representations, ICLR, 2016) 1509.02971 .

[164] P. Henderson, R. Islam, P. Bachman, J. Pineau, D. Precup, and D. Meger, *Deep Reinforcement Learning that Matters,* CoRR (2017), arXiv:1709.06560 .

[165] J. Ellerbroek, ProfHoekstra, and MJRibeiroTUDelft, *Bluesky implementation: underlying the publication "Velocity Obstacle Based Conflict Avoidance in Urban Environment with Variable Speed Limit",* (2021), 10.5281/zenodo.4495643.

[166] M. Ribeiro, J. Ellerbroek, and J. Hoekstra, *Bluesky data: underlying the publication "velocity obstacle based conflict avoidance in urban environment with variable speed limit". 4tu.centre for research data. dataset,* (2021), 10.4121/13691353.

[167] G. Boeing, *OSMnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks,* Computers, Environment and Urban Systems **65**, 126 (2017).

[168] R. Irvine, *Guidance, Navigation, and Control Conference and Exhibit,* Tech. Rep. (EUROCONTROL, 1997).

[169] J. Park and N. Cho, *Collision avoidance of hexacopter uav based on lidar data in dynamic environment,* Remote Sensing **12** (2020).

[170] L. Zheng, P. Zhang, J. Tan, and F. Li, *The Obstacle Detection Method of UAV Based on 2D Lidar,* IEEE Access **7**, 163437 (2019).

[171] L. Yang, K. Han, C. Borst, and M. Mulder, *Impact of aircraft speed heterogeneity on contingent flow control in 4d en-route operation,* Transportation Research Part C: Emerging Technologies **119**, 102746 (2020).

[172] M. Doole, J. Ellerbroek, V. L. Knoop, and J. M. Hoekstra, *Constrained urban airspace design for large-scale drone-based delivery traffic,* Aerospace **8** (2021), 10.3390/aerospace8020038.

[173] N. Samir Labib, G. Danoy, J. Musial, M. R. Brust, and P. Bouvry, *Internet of unmanned aerial vehicles—a multilayer low-altitude airspace model for distributed uav traffic management,* Sensors **19** (2019), 10.3390/s19214779.

[174] J. Cho and Y. Yoon, *Extraction and interpretation of geometrical and topological properties of urban airspace for uas operations,* (2019).

[175] M. Tra, E. Sunil, J. Ellerbroek, and J. Hoekstra, *Modeling the intrinsic safety of unstructured and layered airspace designs,* in *Twelfth USA/Europe Air Traffic Management Research and Development Seminar* (2017).

[176] A. Vela, S. Solak, W. Singhose, and J.-P. Clarke, *A mixed integer program for flight-level assignment and speed control for conflict resolution,* (2010) pp. 5219 – 5226.

[177] R. Huang, H. Liang, P. Zhao, B. Yu, and X. Geng, *Intent-estimation and motion-model-based collision avoidance method for autonomous vehicles in urban environments,* Applied Sciences **7** (2017), 10.3390/app7050457.

[178] J. D. Lawrence, *A Catalog of Special Plane Curves* (Guilford Publications, 2013).

[179] Y. Wu, H. Tan, L. Qin, and B. Ran, *Differential variable speed limits control for freeway recurrent bottlenecks via deep actor-critic algorithm,* Transportation Research Part C: Emerging Technologies **117**, 102649 (2020).

[180] M. Brittain, X. Yang, and P. Wei, *A Deep Multi-Agent Reinforcement Learning Approach to Autonomous Separation Assurance,* (2020), arXiv:2003.08353 .

[181] S. Li, M. Egorov, and M. Kochenderfer, *Optimizing Collision Avoidance in Dense Airspace using Deep Reinforcement Learning,* 13th USA/Europe Air Traffic Management Research and Development Seminar 2019 (2019), arXiv:1912.10146 .

[182] M. Ribeiro, J. Ellerbroek, and J. Hoekstra, *Determining Optimal Conflict Avoidance Manoeuvres At High Densities With Reinforcement Learning,* 10th SESAR Innovation Days (2020).

[183] B. Vonk, *Exploring reinforcement learning methods for autonomous sequencing and spacing of aircraft,* Master's thesis, Delft University of Technology (2019).

[184] D. van der Hoff, *A multi-agent learning approach to air traffic control,* Master's thesis, Delft University of Technology (2020).

[185] L. L. Cruciol, A. C. de Arruda, L. Weigang, L. Li, and A. M. Crespo, *Reward functions for learning to control in air traffic flow management,* Transportation Research Part C: Emerging Technologies **35**, 141 (2013).

[186] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. A. Uc, B. Openai, and I. M. Openai, *Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments*, Tech. Rep., arXiv:1706.02275v4 .

[187] L. Matignon, G. J. Laurent, and N. Le Fort-Piat, *Independent reinforcement learners in cooperative Markov games: a survey regarding coordination problems,* The Knowledge Engineering Review **27**, 1 (2012).

[188] Y. Duan, X. Chen, C. X. B. Edu, J. Schulman, P. Abbeel, and P. B. Edu, *Benchmarking Deep Reinforcement Learning for Continuous Control,* CoRR **abs/1604.06778** (2016).

[189] R. Islam, P. Henderson, M. Gomrokchi, and D. Precup, *Reproducibility of Benchmarked Deep Reinforcement Learning Tasks for Continuous Control,* CoRR **abs/1708.04133** (2017), arXiv:1708.04133 .

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

[191] G. E. Uhlenbeck and L. S. Ornstein, *On the theory of the Brownian motion,* Physical Review **36**, 823 (1930).

[192] M. Papageorgiou, E. Kosmatopoulos, and I. Papamichail, *Effects of variable speed limits on motorway traffic flow,* Transportation Research Record **2047**, 37 (2008).

[193] R. Golding, *Metrics to characterize dense airspace traffic*, Tech. Rep. 004 (Altiscope, 2018).

[194] M. Ribeiro, J. Ellerbroek, and J. Hoekstra, *The Effect of Intent on Conflict Detection and Resolution at High Traffic Densities,* 9th International Conference for Research in Air Transportation (ICRAT) (2020).

[195] S. Weikl, K. Bogenberger, and R. L. Bertini, *Traffic management effects of variable speed limit system on a german autobahn: Empirical assessment before and after system implementation,* Transportation Research Record **2380**, 48 (2013).

[196] M. M., *Atm monitoring and evaluation, 4-lane variable mandatory speed limits 12 month report (primary and secondary indicators)*, Tech. Rep. (European Commission. Directorate General Energy and Transport, 2008).

[197] SESAR, *Consolidated Report on SESAR U-Space Research and Innovation Results,* (2020).

[198] P. Wang, C.-Y. Chan, and A. de La Fortelle, *A reinforcement learning based approach for automated lane change maneuvers,* (2018), arXiv:1804.07871 [cs.RO] .

[199] C.-J. Hoel, K. Wolff, and L. Laine, *Automated speed and lane change decision making using deep reinforcement learning,* 2018 21st International Conference on Intelligent Transportation Systems (ITSC) (2018), 10.1109/itsc.2018.8569568.

[200] A. Mushtaq, I. U. Haq, M. U. Imtiaz, A. Khan, and O. Shafiq, *Traffic flow management of autonomous vehicles using deep reinforcement learning and smart rerouting,* IEEE Access **9**, 51005 (2021).

[201] D. Garg, M. Chli, and G. Vogiatzis, *Deep reinforcement learning for autonomous traffic light control,* in *2018 3rd IEEE International Conference on Intelligent Transportation Engineering (ICITE)* (2018) pp. 214–218.

[202] A. Alizadeh, M. Moghadam, Y. Bicer, N. K. Ure, U. Yavas, and C. Kurtulus, *Automated lane change decision making using deep reinforcement learning in dynamic and uncertain highway environment,* in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)* (2019) pp. 1399–1404.

[203] T. Shi, P. Wang, X. Cheng, C.-Y. Chan, and D. Huang, *Driving decision and control for automated lane change behavior based on deep reinforcement learning,* in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)* (2019) pp. 2895–2900.

[204] M. Doole, J. Ellerbroek, and J. Hoekstra, *Investigation of merge assist policies to improve safety of drone traffic in a constrained urban airspace,* Aerospace **9**, 120 (2022).

[205] R. A. Paielli, *Tactical conflict resolution using vertical maneuvers in en route airspace,* Journal of Aircraft **45**, 2111 (2008).

[206] M. Ribeiro, J. Ellerbroek, and J. Hoekstra, *Velocity obstacle based conflict avoidance in urban environment with variable speed limit,* Aerospace **8** (2021), 10.3390/aerospace8040093.

[207] S. M. Galster, J. A. Duley, A. J. Masalonis, and R. Parasuraman, *Air traffic controller performance and workload under mature free flight: Conflict detection and resolution of aircraft self-separation,* The International Journal of Aviation Psychology **11**, 71 (2001).

[208] U. Gunarathna, H. Xie, E. Tanin, S. Karunasekara, and R. Borovica-Gajic, *Real-time lane configuration with coordinated reinforcement learning,* in *Machine Learning and Knowledge Discovery in Databases: Applied Data Science Track*, edited by Y. Dong, D. Mladenić, and C. Saunders (Springer International Publishing, 2021) pp. 291–307.

[209] K. F. Chu, A. Y. Lam, and V. O. Li, *Dynamic lane reversal routing and scheduling for connected autonomous*

vehicles, in *2017 International Smart Cities Conference (ISC2)* (2017) pp. 1–6.

[210] M. Cai, Q. Xu, C. Chen, J. Wang, K. Li, J. Wang, and X. Wu, *Multi-lane unsignalized intersection cooperation with flexible lane direction based on multi-vehicle formation control,* IEEE Transactions on Vehicular Technology **71**, 5787 (2022).

[211] T. Standfuß, I. Gerdes, A. Temme, and M. Schultz, *Dynamic airspace optimisation,* CEAS Aeronautical Journal **9**, 517 (2018).

[212] M. Schultz and S. Reitmann, *Machine learning approach to predict aircraft boarding,* Transportation Research Part C: Emerging Technologies **98**, 391 (2019).

[213] H. Lee, W. Malik, and Y. C. Jung, *Taxi-out time prediction for departures at charlotte airport using machine learning techniques,* in *16th AIAA Aviation Technology, Integration, and Operations Conference,* https://arc.aiaa.org/doi/pdf/10.2514/6.2016-3910 .

[214] D. D. Nguyen, J. Rohacs, and D. Rohacs, *Autonomous flight trajectory control system for drones in smart city traffic management,* ISPRS International Journal of Geo-Information **10** (2021), 10.3390/ijgi10050338.

[215] M. Hassanalian and A. Abdelkefi, *Classifications, applications, and design challenges of drones: A review,* Progress in Aerospace Sciences **91**, 99 (2017).

[216] A. Degas, M. R. Islam, C. Hurter, S. Barua, H. Rahman, M. Poudel, D. Ruscio, M. U. Ahmed, S. Begum, M. A. Rahman, S. Bonelli, G. Cartocci, G. Di Flumeri, G. Borghini, F. Babiloni, and P. Aricó, *A survey on artificial intelligence (ai) and explainable ai in air traffic management: Current trends and development with future research trajectory,* Applied Sciences **12** (2022), 10.3390/app12031295.

[217] I. R. Brito, M. C. R. Murca, M. d. Oliveira, and A. V. Oliveira, *A machine learning-based predictive model of airspace sector occupancy,* in *AIAA AVIATION 2021 FORUM,* https://arc.aiaa.org/doi/pdf/10.2514/6.2021-2324 .

[218] B. Li, W. Du, Y. Zhang, J. Chen, K. Tang, and X. Cao, *A deep unsupervised learning approach for airspace complexity evaluation,* IEEE Transactions on Intelligent Transportation Systems , 1 (2021).

[219] F. Wieland, J. Rebollo, M. Gibbs, and A. Churchill, *Predicting sector complexity using machine learning,* in *AIAA AVIATION 2022 Forum,* https://arc.aiaa.org/doi/pdf/10.2514/6.2022-3754 .

[220] M. Xue, *Airspace sector redesign based on voronoi diagrams,* Journal of Aerospace Computing, Information, and Communication **6**, 624 (2009), https://doi.org/10.2514/1.41159 .

[221] S. Kulkarni, R. Ganesan, and L. Sherry, *Static sectorization approach to dynamic airspace configuration using approximate dynamic programming,* in *2011 Integrated Communications, Navigation, and Surveillance Conference Proceedings* (2011) pp. J2–1–J2–9.

[222] J. Tang, S. Alam, C. Lokan, and H. A. Abbass, *A multi-objective approach for dynamic airspace sectorization using agent based and geometric models,* Transportation Research Part C: Emerging Technologies **21**, 89 (2012).

[223] C. Tang and Y.-C. Lai, *Deep reinforcement learning automatic landing control of fixed-wing aircraft using deep deterministic policy gradient,* in *2020 International Conference on Unmanned Aircraft Systems (ICUAS)* (2020) pp. 1–9.

[224] A. Tsourdos, I. A. Dharma Permana, D. H. Budiarti, H.-S. Shin, and C.-H. Lee, *Developing flight control policy using deep deterministic policy gradient,* in *2019 IEEE International Conference on Aerospace Electronics and Remote Sensing Technology (ICARES)* (2019) pp. 1–7.

[225] H. Wen, H. Li, Z. Wang, X. Hou, and K. He, *Application of ddpg-based collision avoidance algorithm in air traffic control,* in *2019 12th International Symposium on Computational Intelligence and Design (ISCID),* Vol. 1 (2019) pp. 130–133.

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

[227] D.-T. Pham, N. P. Tran, S. Alam, V. Duong, and D. Delahaye, *A Machine Learning Approach for Conflict Resolution in Dense Traffic Scenarios with Uncertainties,* in *ATM 2019, 13th USA/Europe Air Traffic Management Research and Development Seminar* (Vienne, Austria, 2019).

[228] 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).

[229] 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), 10.48550/ARXIV.1812.05905.

[230] C.-C. Wong, S.-Y. Chien, H.-M. Feng, and H. Aoyama, *Motion planning for dual-arm robot based on soft actor-critic,* IEEE Access **9**, 26871 (2021).

[231] J. M. Hoekstra, J. Ellerbroek, E. Sunil, and J. Maas, *Geovectoring: Reducing Traffic Complexity to Increase*

the Capacity of UAV airspace, Icrat 2018 (2018).

[232] 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 **9** (2022), 10.3390/aerospace9060294.

[233] M. Ribeiro, J. Ellerbroek, and J. Hoekstra, *Distributed conflict resolution at high traffic densities with reinforcement learning,* Aerospace **9** (2022), 10.3390/aerospace9090472.

[234] M. Ribeiro, *Bluesky software: underlying the publication "Improving Algorithm Conflict Resolution Manoeuvres with Reinforcement Learning",* https://data.4tu.nl/articles/software/Bluesky_sof tware_underlying_the_publication_Improving_Algorithm_Conflict_Resolution_Manoeuv res_with_Reinforcement_Learning_/21655760 (2022).

[235] M. Soltani, S. Ahmadi, A. Akgunduz, and N. Bhuiyan, *An eco-friendly aircraft taxiing approach with collision and conflict avoidance,* Transportation Research Part C: Emerging Technologies **121**, 102872 (2020).

[236] S. Li, M. Egorov, and M. Kochenderfer, *Optimizing collision avoidance in dense airspace using deep reinforcement learning,* (2019).

[237] A. Henry, D. Delahaye, and A. Valenzuela, *Conflict Resolution with Time Constraints in the Terminal Maneuvering Area Using a Distributed Q-Learning Algorithm,* 9th International Conference for Research in Air Transportation (ICRAT) (2022).

[238] M. Brittain and P. Wei, *Autonomous air traffic controller: A deep multi-agent reinforcement learning approach,* arXiv (2019), arXiv:1905.01303 .

[239] 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).

[240] R. Dalmau-Codina and E. Allard, *Air traffic control using message passing neural networks and multi-agent reinforcement learning,* (2020).

[241] R. Isufaj, M. Omeri, and M. A. Piera, *Multi-uav conflict resolution with graph convolutional reinforcement learning,* Applied Sciences **12** (2022), 10.3390/app12020610.

[242] M. Brittain and P. Wei, *Scalable autonomous separation assurance with heterogeneous multi-agent reinforcement learning,* IEEE Transactions on Automation Science and Engineering **19**, 2837 (2022).

[243] C. Panoutsakopoulos, B. Yuksek, G. Inalhan, and A. Tsourdos, *Towards safe deep reinforcement learning for autonomous airborne collision avoidance systems,* in *AIAA SCITECH 2022 Forum*.

[244] 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 **135**, 103463 (2022).

[245] Y. Guleria, P. N. Tran, D.-T. Pham, N. Durand, and S. Alam, *A Machine Learning Framework for Predicting ATC Conflict Resolution Strategies for Conformal,* 11th SESAR Innovation Days (2021).

[246] L. Caranti, M. Ribeiro, J. Ellerbroek, and J. Hoekstra, *Safety Optimization of a Layered Airspace Structure with Supervised Learning,* 11th SESAR Innovation Days (2021).

[247] Z. Xue and T. Gonsalves, *Vision based drone obstacle avoidance by deep reinforcement learning,* AI **2**, 366 (2021).

[248] J. Zhang, Z. Zhang, S. Han, and S. Lü, *Proximal policy optimization via enhanced exploration efficiency,* (2020).

[249] K. Malialis, S. Devlin, and D. Kudenko, *Resource abstraction for reinforcement learning in multiagent congestion problems,* (2019), 10.48550/ARXIV.1903.05431.

[250] M. Seo, L. F. Vecchietti, S. Lee, and D. Har, *Rewards prediction-based credit assignment for reinforcement learning with sparse binary rewards,* IEEE Access **7**, 118776 (2019).

[251] M. Zhou, Z. Liu, P. Sui, Y. Li, and Y. Y. Chung, *Learning implicit credit assignment for cooperative multi-agent reinforcement learning,* (2020).

[252] B. J. Lansdell, P. R. Prakash, and K. P. Kording, *Learning to solve the credit assignment problem,* (2019).

[253] L. Feng, Y. Xie, B. Liu, and S. Wang, *Multi-level credit assignment for cooperative multi-agent reinforcement learning,* Applied Sciences **12** (2022), 10.3390/app12146938.

[254] Y. Pu, S. Wang, R. Yang, X. Yao, and B. Li, *Decomposed soft actor-critic method for cooperative multi-agent reinforcement learning,* (2021).

[255] 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.

[256] D. Cuppen, *Conflict Prioritization with Multi-Agent Deep Reinforcement Learning*, Master's thesis, Delft University of Technology (2022).

[257] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, *Playing atari with deep reinforcement learning,* (2013).

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

[259] C. Wang and K. Ross, *Boosting soft actor-critic: Emphasizing recent experience without forgetting the past,* (2019).

[260] W. Fedus, P. Ramachandran, R. Agarwal, Y. Bengio, H. Larochelle, M. Rowland, and W. Dabney, *Revisiting fundamentals of experience replay,* (2020).

[261] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, *Proximal policy optimization algorithms,* (2017).

[262] A. Jeerige, D. Bein, and A. Verma, *Comparison of deep reinforcement learning approaches for intelligent game playing,* in *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)* (2019) pp. 0366–0371.

[263] R. Lincoln, S. Galloway, B. Stephen, and G. Burt, *Comparing policy gradient and value function based reinforcement learning methods in simulated electrical power trade,* IEEE Transactions on Power Systems **27**, 373 (2012).

[264] J. Jiang, C. Dun, T. Huang, and Z. Lu, *Graph convolutional reinforcement learning,* (2018).

[265] N. P. Tran, D.-T. Pham, S. K. Goh, S. Alam, and V. Duong, *An intelligent interactive conflict solver incorporating air traffic controllers' preferences using reinforcement learning,* in *2019 Integrated Communications, Navigation and Surveillance Conference (ICNS)* (2019) pp. 1–8.

[266] M. Ribeiro, J. Ellerbroek, and J. Hoekstra, *Using reinforcement learning to improve airspace structuring in an urban environment,* Aerospace **9** (2022), 10.3390/aerospace9080420.

[267] M. Grzes and D. Kudenko, *Theoretical and empirical analysis of reward shaping in reinforcement learning,* in *2009 International Conference on Machine Learning and Applications* (2009) pp. 337–344.

[268] A. Agogino and K. Tumer, *Unifying temporal and structural credit assignment problems,* in *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems, 2004. AAMAS 2004.* (2004) pp. 980–987.

[269] K. Tumer and A. Agogino, *Distributed agent-based air traffic flow management,* in *Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems*, AAMAS '07 (Association for Computing Machinery, New York, NY, USA, 2007).

[270] A. Ilyas, L. Engstrom, S. Santurkar, D. Tsipras, F. Janoos, L. Rudolph, and A. Madry, *A closer look at deep policy gradients,* (2018).

[271] M. Ribeiro, J. Ellerbroek, and J. Hoekstra, *Improvement of Conflict Detection and Resolution at High Densities Through Reinforcement Learning*, 9th International Conference for Research in Air Transportation (ICRAT) (2020).

[272] F. Tambon, G. Laberge, L. An, A. Nikanjam, P. S. N. Mindom, Y. Pequignot, F. Khomh, G. Antoniol, E. Merlo, and F. Laviolette, *How to certify machine learning based safety-critical systems? A systematic literature review,* CoRR **abs/2107.12045** (2021), 2107.12045 .

[273] R. Dalmau, F. Ballerini, H. Naessens, S. Belkoura, and S. Wangnick, *An explainable machine learning approach to improve take-off time predictions,* Journal of Air Transport Management **95**, 102090 (2021).

[274] European Union Aviation Safety Agency, *Study on the societal acceptance of Urban Air Mobility in Europe*, Tech. Rep. (European Union Aviation Safety Agency, 2021).

[275] European Union Aviation Safety Agency, *Easy Access Rules for Unmanned Aircraft Systems*, Tech. Rep. (European Union Aviation Safety Agency, 2021).

ACKNOWLEDGEMENTS

Any piece of research is a group effort, and this thesis is no exception. The latter would not have been possible without the supervision, expertise, and all the valuable feedback from my supervisors Jacco Hoekstra and Joost Ellerbroek. Not only do they have invaluable knowledge in the field, but they are capable of motivating others to dive deeper into the problems at hand. I want to thank them for our weekly meetings and for sharing their love for research with me. The latter heavily inspired me to continue my path at TU Delft.

The mid-years of this thesis were singular in terms of experiencing a one-in-a-century worldwide pandemic. As a consequence, I missed most of the conversations around the coffee machine at the Aerospace Faculty. The online meetings in our department were essential to keep the inspiration flowing throughout the sometimes stressful years of a PhD thesis. I would like to thank my colleagues Andrei Badea, Andres Morfin Veytia, Esther Roosenbrand, Isabel Metz, Jan Groot, Jerom Maas, Julia Rudnyk, Junzi Sun, Malik Doole, Mike Zoutendijk, and Simon van Oosterom for all the companionship and valuable discussions. Doing a PhD would have been much harder without this amazing group. And a heartfelt thank you to their partners (and pets) as well, who I consider equally part of this PhD group.

Apart from the previous direct colleagues, there are other numerous colleagues within the department who made the last four years much more enjoyable. I will not attempt to name all of them, knowing that I would definitely miss someone. However, I would like to thank everyone for the Wednesday lunches, PhD drinks, Christmas dinners, carpentry sessions to build the final PhD plaques, and the climbing and canoeing events. I hope to continue seeing you all at the faculty.

A special thank you to Isabelle El-Hajj, Gijs de Rooij, Rowenna Wijlens, and Tiago Monteiro Nunes with whom I enjoyed many great conversations on the days I went to the faculty. During and after covid, the hallways at the faculty were unfortunately often empty. In the few days I went to the faculty, I always found them there – thank you for always being present. Finally, also thank you to Dirk Van Baelen and Sven Pfeiffer for always making a big effort to keep the companionship spirit within the section, and coming up with great ideas for activities.

Andries Muis, Bertine Markus, Ferdinand Postema, and Harold Thung were also essential to me completing the PhD on time. Thank you for making sure that the lab servers were always up and running, and that we could always run our simulations as smoothly as possible. Additionally, thank you to the TU Delft Library and the data stewards for making the lives of PhD students so much easier.

During the last four years, I found that I enjoyed supervising students more than I thought I would. This is probably because I was lucky to have great students capable of inspiring others. I thank Cristina Dumitrescu, Daan Cuppen, Jan Groot, Leonardo Caranti, and Timon Rowntree for their great work. Their research greatly improved the knowledge within the section and often motivated me to continue with my own

research. Furthermore, thank you to Andrei Anisimov and Mohamad Fathi for their great companionship during the supervision of the DSE student group.

Moreover, I would like to thank everyone who contributed to make the BlueSky ATM simulator what it is today. I used it throughout every step of this thesis. Its current state is the result of numerous hours of work from professors, students, and programming enthusiasts. It is nothing short of impressive how far this simulator has come. At conferences, I often met people who wish they had had something similar during their own PhD thesis. I have no doubt that the number of BlueSky users/developers will continue to grow.

I also acknowledge the European Union for its support of the AW-Drones project, in which I participated. The project gave me valuable insight into the social impact of drones operating in an urban environment, and the certification hurdles that one must overcome to make this possible. I am thankful for the support of EuroUSC, DLR, and NLR in navigating the complicated work of regulations and standards for unmanned aviation.

Finally, I would like to thank all the people who were part of my life outside of the faculty. Most of them can hardly remember the topic of this thesis, which has a humbling effect. Thank you to my parents who, to this day, are still very confused about what a PhD is. Thank you to Vanessa, the only person to read all the draft versions of my papers, saving my supervisors from learning how messy my typing really is. Your aversion to all the heatmaps in this thesis is dully noted. Thank you Mariana for playing an important role in key moments of my life since we were 18, and for pushing me to move to the Netherlands. On the one hand, this move led me to not seeing my Portuguese friends as much as I would like to. Nevertheless, Filipe, Lili, Jonas, Marta, and Sónia, thank you for always wishing me the best no matter where I live. On the other hand, I made new great friends here. First, I want to thank the people who made Delft a lot more enjoyable when I first moved to the Netherlands, almost 8 years ago: Echo, Mariana, Nadine, Natalie, and Sterre. Thank you Marijn and Nadine for being much better than me at bouldering, and for motivating me to be better. The best advice I can give to someone about to start their PhD, is to find friends and hobbies unrelated to your PhD work.

Marta Ribeiro
January, 2023

CURRICULUM VITÆ

Marta Joana RIBEIRO

20/04/1990   Born in Lisbon, Portugal

EDUCATION

2019 – 2023   **PhD. Aerospace Engineering**
Delft University of Technology, Delft, The Netherlands

2011 – 2013   **MSc. Aerospace Engineering**
Instituto Superior Técnico, Lisbon, Portugal

2008 – 2011   **BSc. Aerospace Engineering**
Instituto Superior Técnico, Lisbon, Portugal

EXPERIENCE

2016 – 2018   **Software Architect**
CGI, Rotterdam, The Netherlands

2014 – 2016   **Software Engineer**
Simteq, Vijfhuizen, The Netherlands

2013 – 2014   **Operations Engineer**
Portugália Airlines, Lisbon, Portugal

2012 – 2013   **Research Fellow**
Instituto Superior Técnico, Lisbon, Portugal

LIST OF PUBLICATIONS

JOURNAL ARTICLES

6. **M. Ribeiro**, J. Ellerbroek, and J. Hoekstra, Improving Algorithm Conflict Resolution Manoeuvres With Reinforcement Learning, Aerospace 9 (2022)

5. **M. Ribeiro**, J. Ellerbroek, and J. Hoekstra, Distributed Conflict Resolution at High Traffic Densities with Reinforcement Learning, Aerospace 9 (2022)

4. **M. Ribeiro**, J. Ellerbroek, and J. Hoekstra, Using Reinforcement Learning to Improve Airspace Structuring in an Urban Environment, Aerospace 9 (2022)

3. **M. Ribeiro**, J. Ellerbroek, and J. Hoekstra, Using Reinforcement Learning in Layered Airspace to Improve Layer Change Decision, Aerospace 9 (2022)

2. **M. Ribeiro**, J. Ellerbroek, and J. Hoekstra, Velocity Obstacle Based Conflict Avoidance in Urban Environment with Variable Speed Limit, Aerospace 8 (2021)

1. **M. Ribeiro**, J. Ellerbroek, and J. Hoekstra, Review of Conflict Resolution Methods for Manned and Unmanned Aviation, Aerospace 7 (2020)

CONFERENCE PROCEEDINGS

10. C. A. Cadea & A. M. Veytia & D.J. Groot & **M. Ribeiro**, Lateral and Vertical Air Traffic Control Under Uncertainty Using Reinforcement Learning, 12th SESAR Innovation Days (2022)

9. D.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)

8. I. Panchal, I. C. Metz, **M. Ribeiro** & S. Armanini, Urban Air Traffic Management for Collision Avoidance with Uncooperative Airspace Users, 33rd International Council of the Aeronautical Sciences (ICAS) (2022)

7. C. A. Badea & A. M. Veytia, **M. Ribeiro**, M. Doole, J. Ellerbroek, and J. Hoekstra, Limitations of Conflict Prevention and Resolution in Constrained Very Low-Level Urban Airspace, 11th SESAR Innovation Days (2021)

6. L. Caranti, **M. Ribeiro**, J. Ellerbroek, and J. Hoekstra, Safety Optimization of a Layered Airspace Structure with Supervised Learning, 11th SESAR Innovation Days (2021)

5. S. Cain, C. Torens, A. Volkert, P. Juchmann, F. Tomasello, M. Natale, J. Vreeken, T. v. Birgelen, **M. Ribeiro**, J. Ellerbroek, D. Taurino, and M. Ducci, Standards for UAS-Acceptable Means of Compliance for Low Risk SORA Operations, in AIAA Scitech 2021 Forum

4. S. Cain, C. Torens, A. Volkert, P. Juchmann, F. Tomasello, M. Natale, J. Vreeken, T. v. Birgelen, **M. Ribeiro**, J. Ellerbroek, D. Taurino, and M. Ducci, Standards für UAV -Nachweismöglichkeiten für die Umsetzung des SORA- Prozesses im Bereich niedrigerRisikoklassen, Deutscher Luft- und Raumfahrtkongress, (2020)

3. **M. Ribeiro**, J. Ellerbroek, and J. Hoekstra,Determining Optimal Conflict Avoidance Manoeuvres At High Densities With Reinforcement Learning, 10th SESAR Innovation Days (2020)

2. **M. Ribeiro**, J. Ellerbroek, and J. Hoekstra, The Effect of Intent on Conflict Detection and Resolution at High Traffic Densities, 9th International Conference for Research in Air Transportation (ICRAT) (2020)

1. **M. Ribeiro**, J. Ellerbroek, and J. Hoekstra, Analysis of conflict resolution methods for manned and unmanned aviation using fast-time simulations, 9th SESAR Innovation Days (2019)

OPEN-SOURCE SOFTWARE

5. **M. Ribeiro**, Bluesky data: underlying the publication Bluesky software: underlying the publication 'Improving Algorithm Conflict Resolution Manoeuvres with Reinforcement Learning', TU Delft - 4TU.Research Data, 1 Dec 2022

4. **M. Ribeiro**, Bluesky data: underlying the publication Bluesky software: underlying the publication 'Distributed Conflict Resolution at High Traffic Densities with Reinforcement Learning', TU Delft - 4TU.Research Data, 14 Nov 2022

3. A. Badea, A. M. Veytia, **M. Ribeiro**, and D. Groot, Air Traffic Control Reinforcement Learning Environment, TU Delft - 4TU.Research Data, 9 Aug 2022

2. **M. Ribeiro**, Bluesky data: underlying the publication 'Velocity Obstacle Based Conflict Avoidance in Urban Environment with Variable Speed Limit', TU Delft - 4TU.Research Data, 2 Feb 2021

1. **M. Ribeiro**, Bluesky data: underlying the publication 'Review of Conflict Resolution Methods for Manned and Unmanned Aviation', TU Delft - 4TU.Research Data, 22 Apr 2020

AWARDS

2. **Best Paper Award** at the 10th International Conference for Research in Air Transportation (ICRAT) 2022, in the Safety Track, to "Improving Safety of Vertical Manoeuvres in a Layered Airspace with Deep Reinforcement Learning", D.J. Groot, **M. Ribeiro**, J. Ellerbroek, and J. Hoekstra

1. **1st place** at the EUROCONTROL Innovation MasterClass 2022, on challenge "Conflict Resolution with Reinforcement Learning" (together with other teams members: Andrei Badea, Jan Groot, Andres Morfin Veytia)