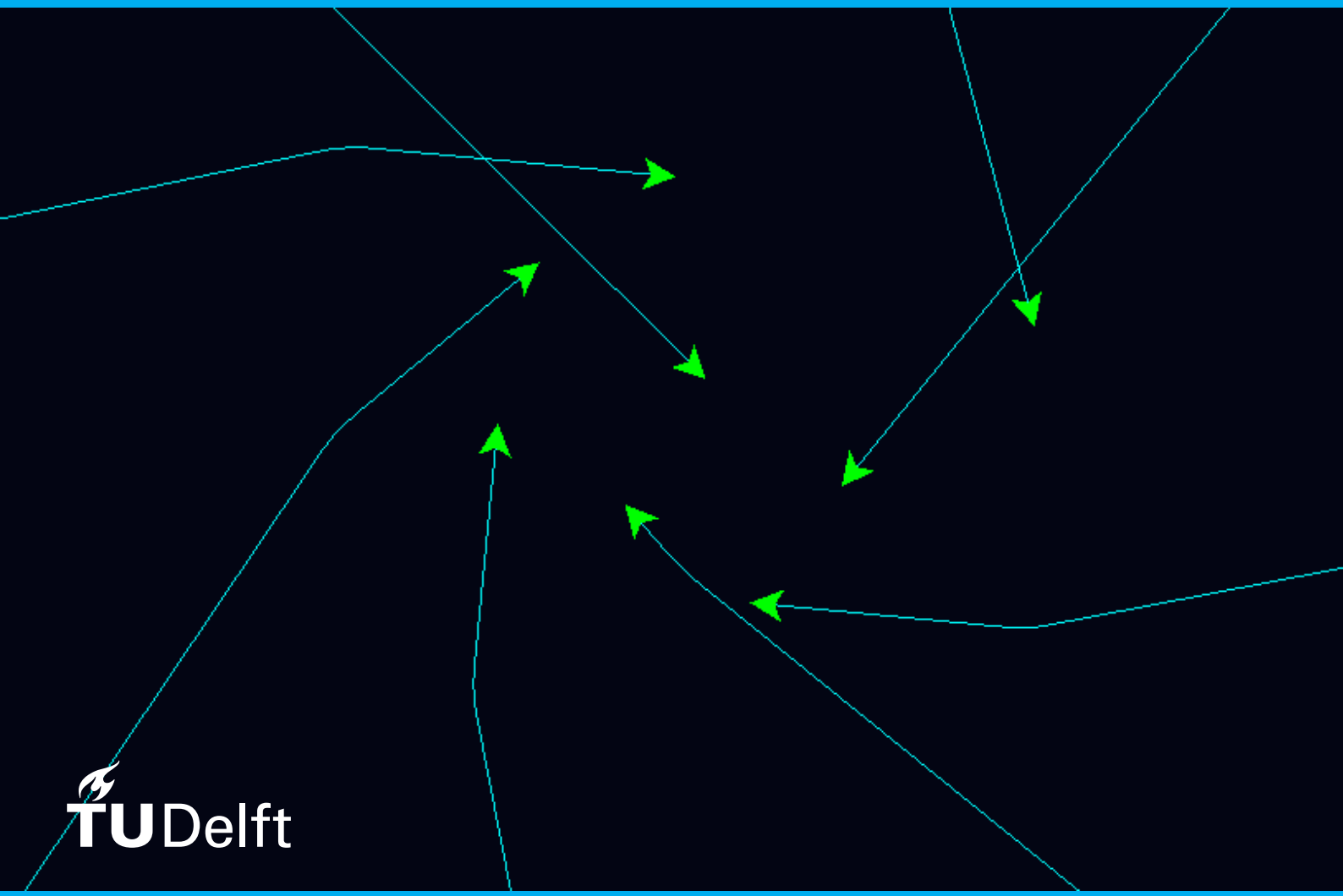
# iFly in BlueSky

Implementation and Comparison of the A3 CD&R Model in Open Source ATM Simulator BlueSky

C.J. van Baardwijk



TU Delft

# iFly in BlueSky

## Implementation and Comparison of the A3 CD&R Model in Open Source ATM Simulator BlueSky

by


## C.J. van Baardwijk


to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Friday the 20th of March 2026.

| | |
|---|---|
| Student number: | 4652347 |
| Project duration: | 09/09/2024 – 20/03/2026 |
| Thesis committee: | Dr. — |
| | Dr.ir. J. Ellerbroek |
| | Dr. M.J. Ribeiro |
| | Dr. A. Amiri-Simkooei |



TU Delft

# Acknowledgements

Dear reader,

As I write this preface, I bring a finishing touch to my master's thesis. It has been a tough challenge for me personally, however I feel proud that I managed to get there in the end. I have researched much during this project, not only air traffic management (as interesting as that may be) but also myself. The results were rather unexpected, but I look forward to taking on a new challenge.

I would like to thank my supervisors Joost, Marta, and Henk. Your continued support during this rather long trajectory kept me going, even when I almost stalled. I am grateful for the space you gave me to share my struggles, and I hope you will have some use of my work.

Furthermore, I would like to thank my parents for their everlasting patience and love. My time at the TU Delft was a bit longer than anticipated beforehand, with some twists and turns along the way. However, you have always encouraged me to do what I want, even now when I choose to do something completely different.

I also want to express my gratitude and love for my girlfriend. You have supported and comforted me during this more difficult period of my life, and you always make me feel stronger and more confident.

Now that I have reached the destination at TU Delft, I can see that it is only a waypoint turning me into a new direction and a new challenge, and I look forward to it!

Chiel van Baardwijk
Delft, March 2026

# Contents

# List of Figures

# List of Tables

# Nomenclature

**Abbreviations**

| | |
|---|---|
| 4DTBO | 4-Dimensional Trajectory-Based Operation |
| A3 | Autonomous Aircraft Advanced |
| ACAS | Airborne Collision Avoidance System |
| AFR | Autonomous Flight Rules |
| AIS | Aeronautical Information Services |
| ANSP | Air Navigation Service Provider |
| ASAS | Airborne Separation Assistance System |
| ATC | Air Traffic Control |
| ATM | Air Traffic Management |
| CD&R | Conflict Detection & Resolution |
| CDTI | Cockpit Display of Traffic Information |
| ConOps | Concept of Operations |
| CPL | Conflict Processing Logic |
| CPP | Conflict Processing Performance |
| CSZ | Comfort Separation Zone |
| CTA | Controlled Time of Arrival |
| DCPN | Dynamically Coloured Petri Net |
| DEP | Domino Effect Parameter |
| FB | Functional Block |
| FOC | Flight Operation Centre |
| GNC | Guidance, Navigation, and Control |
| GPWS | Ground Proximity Warning System |
| HMI | Human Machine Interface |

| | |
|---|---|
| ICAO | International Civil Aviation Organization |
| IPN | Interaction Petri Net |
| IPR | Intrusion Prevention Rate |
| LoS | Loss of Separation |
| LPN | Local Petri Net |
| LTACD | Long Term Area Conflict Detection |
| LTAZ | Long Term Awareness Zone |
| MC | Monte Carlo |
| MSZ | Minimum Separation Zone |
| MTAZ | Medium Term Awareness Zone |
| MTC | Medium Term Conflict |
| MTCD&R | Medium Term Conflict Detection & Resolution |
| MTCR | Medium Term Conflict Resolution |
| MTTH | Medium Term Time Horizon |
| MVP | Modified Voltage Potential |
| OSED | Operational Services and Environment Description |
| PAZ | Protected Aircraft Zone |
| PF | Pilot-Flying |
| PNF | Pilot-Not-Flying |
| RAV | Reachable Avoidance Velocity |
| RBT | Reference Business Trajectory |
| RT | Reference Time |
| RTTL | Remaining Time To Loss of separation |
| SL | Service Level |
| SSA | Self Separating Airspace |
| SSD | Solution Space Diagram |
| STAZ | Short Term Awareness Zone |
| STC | Short Term Conflict |

| | |
|---|---|
| STCD&R | Short Term Conflict Detection & Resolution |
| STCR | Short Term Conflict Resolution |
| STT | Short Term time Threshhold |
| STTH | Short Term Time Horizon |
| SWIM | System Wide Information Management system |
| TAS | True Air Speed |
| TMA | Terminal Maneuvring Area |
| TTL | Time To Loss of separation |
| VO | Velocity Obstacle |

**Greek Symbols**

| | |
|---|---|
| $\mu_{bank}$ | Bank angle of aircraft [$deg$] |
| $\lambda$ | Turn geometry parameter |
| $\mu$ | Mean |
| $\phi$ | Heading angle of aircraft [$deg$] |
| $\sigma$ | Standard deviation |

**Symbols**

| | |
|---|---|
| $Q$ | Intent information matrix containing trajectory waypoints |
| $A$ | Size of an area [$NM^2$] |
| $D$ | Distance [$NM$] |
| $d_{CPA}$ | Distance at closest point of approach [$morNM$] |
| $DEP$ | Domino Effect Parameter |
| $H_{PAZ}$ | Vertical separation minimum [$morft$] |
| $I$ | Intrusion [%] |
| $i_H$ | Horizontal intrusion [$m$] |
| $IPR$ | Intrusion Prevention Rate |
| $L$ | Length of the side of an area [$NM$] |
| $m$ | Vertical mode: 0 = level, 1 = climb/descend |
| $n_{cfl}$ | Number of conflicts |

| | |
|---|---|
| $n_{cfl}^{OFF}$ | Number of conflicts without conflict resolution |
| $n_{cfl}^{ON}$ | Number of conflicts with conflict resolution active |
| $n_{int}$ | Number of intrusions |
| $P$ | Priority |
| $Q_{turn}$ | Trajectory with approximated turn points |
| $R_{PAZ}$ | Horizontal separation minimum [$morNM$] |
| $R_{turn}$ | Turn radius of aircraft [$m$] |
| $s$ | Turn direction: $-1$ = left, 0 = straight, 1 = right |
| $t$ | Time [$s$] |
| $t_i$ | Intended passage time of trajectory change point $i$ [$s$] |
| $W$ | Work done [$J$] |
| $x_i, y_i, z_i$ | Intended position coordinates of TCP $i$ [$m$] |

**Vectors**

| | |
|---|---|
| $\vec{a}$ | Normalized velocity vector of aircraft at start of turn |
| $\vec{b}$ | Normalized velocity vector of aircraft at end of turn |
| $\vec{c}$ | Perpendicular direction vector at start of turn |
| $\vec{x}$ | Distance vector [$m$] |
| $\vec{P}_A(t)$ | Position vector of aircraft A at time $t$ [$m$] |
| $\vec{P}_B(t)$ | Position vector of aircraft B at time $t$ [$m$] |
| $\vec{s}$ | Displacement vector [$m$] |
| $\vec{T}$ | Thrust vector [$N$] |
| $\vec{v}$ | Velocity vector [$m/s$] |
| $\vec{v}_c(t_c)$ | Collision course velocity vector at time $t_c$ [$m/s$] |
| $\vec{x}$ | Position vector [$m$] |

# Introduction

As air traffic continues to grow worldwide, finding safer and more efficient ways to manage the sky becomes increasingly important. Currently, air traffic controllers on the ground are responsible for keeping aircraft safely separated from each other a system that works well but faces limitations as the skies become more crowded. This research explores an alternative vision for the future: what if aircraft could take responsibility for maintaining their own safe distances from each other, similar to how cars navigate on highways?

This thesis project originated from an interest in understanding how such "free flight" operations might work in practice. What are the challenges holding the development and implementation of free flight operations back, and what can be done to overcome those challenges? This thesis may not answer such big questions, but it might be a small step forward.

From a societal perspective, the research contributes to ongoing discussions about the future of aviation. As demand for air travel grows particularly in regions experiencing rapid economic development finding ways to increase airspace capacity, or decrease climate impact, without compromising safety is important. Free flight concepts could help accommodate this decrease in climate impact, allowing aircraft to fly optimal routes at optimal speeds and altitudes. However, understanding the limitations and challenges is equally important, ensuring that new technologies are implemented responsibly and with full awareness of their implications for safety.

This thesis report is organized as follows: Part I presents the scientific paper containing the core research findings. Part II contains the relevant literature study that supports the research. Finally, Part III provides some supporting work and the bibliography.

# I

## Scientific Paper

# Analysing the Effect of Intent Information in Conflict Resolution

C.J. van Baardwijk,*

Delft University of Technology, Delft, The Netherlands

**Abstract**

The anticipated growth in air traffic is expected to increase demand on air traffic control capacity beyond its current limits, motivating the development of autonomous conflict detection and resolution (CD&R) methods for self-separating airspace. While both state-based and intent-based conflict resolution methods have been studied extensively, a direct comparison between the two approaches under identical conditions is lacking. This paper addresses that gap by comparing the intent-based A3 CD&R model to the state-based Modified Voltage Potential (MVP) resolution method. Performance is evaluated through verification scenarios and random traffic simulations at three density levels, using simulations in the BlueSky ATM simulator. Results show that the A3 method achieves substantially fewer conflicts and losses of separation than MVP across all density levels, with a markedly lower domino effect parameter indicating better airspace stability. However, when losses of separation do occur under the A3 model, the intrusions are larger than those observed with MVP. Regarding efficiency, MVP incurs large flight time increase due to speed-based resolutions, whereas the A3 model resolutions result in minimal flight time increases at the cost of larger route deviations. These findings demonstrate that intent information provides meaningful advantages for conflict resolution performance.

## 1 Introduction

Air traffic has been growing steadily over the past 20 years [1]. The International Air Transport Association (IATA) expects that the total number of yearly passengers will reach 10 billion by 2050 [2]. This growth introduces numerous challenges. One significant challenge is the increase in air traffic flow management delays. According to Eurocontrol, en route delays reached the second highest value in 20 years, mainly due to limitations in the capacity of the air traffic control (ATC) [3]. ATC currently uses structured airways to manage traffic, allowing a single controller to monitor an entire sector. However, this approach has several drawbacks: airways may not always be the most direct or optimal routes, aircraft may not fly at their optimal altitude or speed, and a lot of airspace remains unused, artificially increasing local traffic density [4].

To address this challenge the concept of Free Flight has been developed. Free Flight is based on the optimization of routes by airlines, making the aircraft responsible for maintaining separation [4]. This approach allows aircraft to fly direct, optimal routes to their destinations at preferred altitudes and speeds, utilizing all available airspace.

To maintain safe operations within a Self Separating Airspace (SSA), aircraft must be able to detect and resolve potential conflicts with other aircraft. This involves conflict detection,

*MSc Student, Sustainable Air Transport, Faculty of Aerospace Engineering, Delft University of Technology

which is the identification of a future Loss of Separation (LoS) where an aircraft intrudes into the Protected Aircraft Zone (PAZ) of another aircraft. The PAZ is typically defined by a horizontal radius of 5 nautical miles (NM) and a vertical height of 1000 feet (ft) [5, 6].

Automatic Dependent Surveillance - Broadcast (ADS-B) technology allows aircraft to communicate their positions and trajectories to other aircraft. For shorter look-ahead times, the trajectory of surrounding aircraft can be estimated using state information, assuming a linear continuation of the current state. However, this information becomes less accurate for longer look-ahead times, as aircraft may deviate from this linear continuation due to planned turns and route changes. To address this, intent information, which is the planned future 4D trajectory of an aircraft, can be broadcast to surrounding aircraft, enabling more accurate conflict detection over medium-term time horizons [7, 8, 9]. Using intent information, conflicts can be detected and thus resolved earlier with smaller deviations from the planned route [10].

A number of review studies have examined and categorised existing CD&R methods. Kuchar and Yang (2000) provide a foundational taxonomy of conflict detection and resolution modelling methods, covering the diversity in approaches to trajectory propagation, uncertainty handling, and resolution manoeuvre types [5]. More recently, Ribeiro et al. (2020) extended this overview with an up-to-date review of conflict resolution methods for both manned and unmanned aviation, covering a range of reactive, prescribed, and optimisation-based approaches and comparing their performance characteristics in random traffic simulations [6]. These reviews illustrate that, while a large number of state-based conflict resolution methods have been developed and evaluated, comparative studies between state-based and intent-based methods remain limited, with most studies evaluating individual methods in isolation or comparing different state-based approaches against each other.

Thus, it is relatively unclear whether the additional information provided by intent translates into meaningful performance improvements in conflict resolution over state-based approaches. This paper aims to fill this gap by comparing the intent-based Autonomous Aircraft Advanced (A3) CD&R model with the state-based Modified Voltage Potential (MVP) resolution method.

Developed within the European iFly project, the A3 Concept of Operations describes an autonomous en-route ATM environment in which all aircraft are responsible for airborne self-separation without ground support from ATC, using intent information broadcast via ADS-B to perform conflict detection and resolution [11]. This paper focuses on the Airborne Separation Assistance System (ASAS) functionality of the A3 ConOps. The A3 CD&R model is an example of a system employing intent information for conflict detection and resolution [12]. The A3 CD&R model is an agent-based model that implements both medium-term and short-term conflict detection and resolution using intent information [11].

For this comparison, the MVP method is selected as the state-based conflict resolution method. MVP is a reactive, force-field-based conflict resolution method in which conflicting aircraft are modelled as charged particles generating a repelling avoidance vector [4]. A key property of MVP is its implicit coordination: since each aircraft independently computes a resolution assuming the other will not manoeuvre, the conflict geometry naturally produces complementary avoidance directions without requiring explicit communication [6, 4]. MVP has been used as a reference method in previous simulation studies, including the analysis of airspace structures by Sunil et al. and the review of Ribeiro et al. (2020), which makes it a suitable conflict resolution method for the present comparison [6, 13].

To evaluate both methods, several experiments are conducted using the BlueSky Air Traffic Management (ATM) simulator. First, scenarios with two aircraft and eight aircraft are used to compare the basic functionality and behaviour of both conflict resolution methods. Subsequently, random traffic simulations at three traffic density levels are performed with both the A3 and MVP method. The scenarios considered are horizontal, with no climb or descend

considered.

First the methodology is described in section 2, elaborating on the numerical implementation of the A3 CD&R model and the MVP conflict resolution method. Additionally, the experimental design is described in section 3. Next, the results are presented in section 4. The method and results are discussed in section 5 and the conclusions drawn in section 6.

# 2 Methodology

To test the performance of the A3 CD&R model it is implemented in the BlueSky ATM simulator using a plugin [14]. A description of this numerical implementation is given in section 2.1. Additionally, the numerical implementation of the state-based conflict detection and the MVP resolution method are described in section 2.2.

## 2.1 A3 CD&R Model

The implementation of the A3 CD&R model consists of three main parts. First, the intent information is described in section 2.1.1. Secondly, the conflict detection is elaborated upon in section 2.1.2. Finally, the conflict resolution is described in section 2.1.3.

### 2.1.1 Intent Information

The intent information of an aircraft is when and where the aircraft intends to be in the future. This starts at the origin, where the aircraft is at that moment, and ends at the destination, where the aircraft leaves the SSA. In between, if the route is not direct, trajectory change points (TCP) are stored in the route as waypoints. This route is called $Q$ and looks like Equation 1:

$$Q = \begin{bmatrix} t_1 & x_1 & y_1 & z_1 \\ \cdots & \cdots & \cdots & \cdots \\ t_n & x_n & y_n & z_n \end{bmatrix}, \tag{1}$$

with $t_i$ the intended passage time of TCP $i$, and $x_i, y_i, z_i$ the intended position of TCP $i$ [15].

However, aircraft usually do not fly directly over a TCP. They start turning before and end turning after the TCP, flying by the TCP. To approximate the trajectory more accurately, the fly by turn is cut up into two straight segments [15]. This is shown in Figure 1. One TCP is thus split up into a point where the aircraft starts the turn, a point at the middle of the turn, and a point where the aircraft ends the turn [15, 16].

To inform other aircraft of the trajectory, the aircraft onboard intent information is broadcast. The broadcast intent information is the coded intent information stored in $Q_{turn}$. To add information to the TCPs, two columns are added: whether the point is the start of a turn and if so whether to the left or to the right ($s \in \{-1, 0, 1\}$), and whether the point is the start of a climb or descend ($m \in \{0, 1\}$) [15, 16]. Since there is no climb or descend considered, the latter is always 0. Furthermore, at the origin or destination the aircraft will make no turn and does not climb or descend. The intent information will then look as Equation 2 [15]:

$$Q_{turn} = \begin{bmatrix} t_1 & x_1 & y_1 & z_1 & s_1 = 0 & m_1 = 0 \\ t_2 & x_2 & y_2 & z_2 & s_2 & m_2 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ t_n & x_n & y_n & z_n & s_n = 0 & m_n = 0 \end{bmatrix}. \tag{2}$$

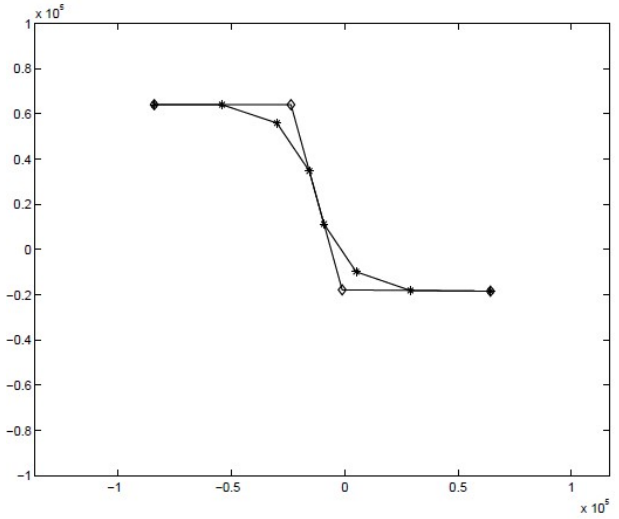



Figure 1: Example trajectory, black diamond shapes are origin and destination, white diamond shapes are TCPs, and stars are the approximation of the fly by turn [15].

### 2.1.2 Conflict Detection

Now using the received $Q_{turn}$ matrices from surrounding aircraft, the own aircraft can detect conflicts using a dynamic velocity obstacle (DVO) approach [12]. The intended trajectory consists of segments, each with a centerline (from position $i$ to position $i+1$) and a 3D volume (the PAZ) around this centerline. For each surrounding aircraft, the own aircraft matches each segment which is in the same time interval of both own intent and other intent and computes the distance at the closest point of approach (CPA) $\mathbf{d}_{CPA}$ and time when the aircraft enters the conflict $t_{in}$ [15, 16]. This is done similarly to state-based conflict detection, described in section 2.2.1, with each segment being a future state of the aircraft. The horizontal conflict geometry between the ownship aircraft and an intruder aircraft on such a segment is shown in Figure 2a. To construct this conflict geometry for each segment, the starting positions of each segment are extrapolated backwards to $t_0$. This is shown in Figure 2b. The summation of these segments form a DVO.

For each segment, using the relative position and velocity, the time at the CPA at that segment is computed using Equation 3:

$$t_{CPA} = -\frac{\mathbf{d}_{rel}{}^T \mathbf{v}_{rel}}{||\mathbf{v}_{rel}||^2}, \tag{3}$$

where $t_{CPA}$ is the time at the CPA, $\mathbf{d}_{rel}$ the relative distance vector between the aircraft, and $\mathbf{v}_{rel}$ the relative velocity vector between the aircraft [6, 15, 16]. From this the distance vector at the CPA $\mathbf{d}_{CPA}$ is computed by Equation 4 [6, 15, 16]:

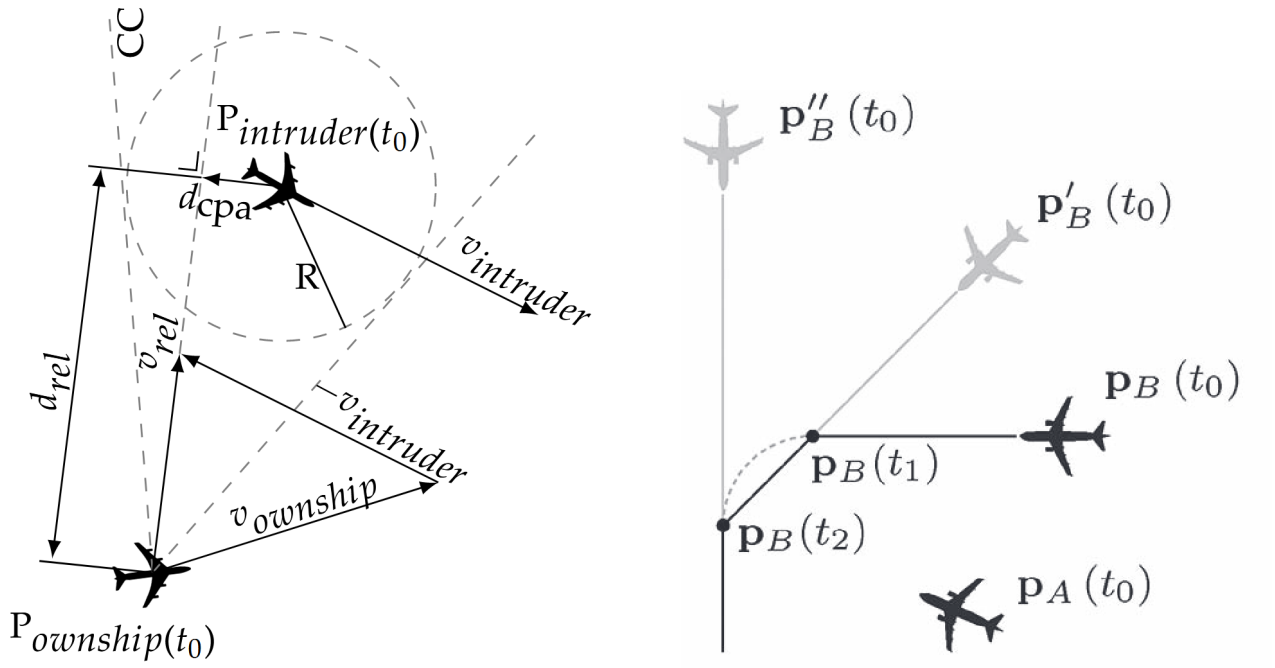
$$\mathbf{d}_{CPA} = \mathbf{d}_{rel} - t_{CPA} \cdot \mathbf{v}_{rel}. \tag{4}$$

The time at which the aircraft enters the conflict $t_{in}$ is then computed using Equation 5:

$$t_{in} = t_{CPA} - \frac{\sqrt{R_{PAZ}^2 - ||\mathbf{d}_{CPA}||^2}}{||\mathbf{v}_{rel}||}, \tag{5}$$

where $R_{PAZ}$ is the radius of the protected aircraft zone [6, 15, 16].

Conflict detection in the A3 model is split up into two phases: short-term conflict detection (STCD) and medium-term conflict detection (MTCD). STCD is performed every $1.5sec$ and




(a) Conflict geometry per segment [6].

(b) Backward extrapolation to $t_0$ [10].

Figure 2: Conflict detection using intent information.

detects conflicts up to three minutes ahead [12, 15, 16]. MTCD is performed every $90sec$ and detects conflicts up to 10 minutes ahead [12, 15, 16]. If $||\mathbf{d}_{CPA}||$ is less than the PAZ horizontal radius of $5NM$ and $t_{in}$ is within the look ahead time of the conflict detection phase performed, a conflict is detected. It is assumed aircraft always receive up to date intent information from other aircraft.

### 2.1.3 Conflict Resolution

The A3 CD&R model uses two conflict resolution phases: the medium-term conflict resolution (MTCR) and the short-term conflict resolution (STCR). The MTCR is activated for conflicts that are more than 3 minutes and less than 10 minutes ahead, and are thus detected by MTCD [12]. The STCR is activated for conflicts less than 3 minutes ahead, thus detected by STCD [12]. The aircraft will always prioritize solving the most imminent conflict. Both MTCR and STCR use the above described DVO approach to find conflict-free trajectories [12, 15].

**Medium-Term Conflict Resolution**

In MTCR, the aircraft with the higher priority has the right-of-way, and the aircraft with the lower priority value must resolve the conflict by changing its intent [12]. Priority is mainly based on the aircraft's remaining distance to its destination [15]. The MTCR priority $PRIO^i_{MTCR}$ of aircraft $i$ is computed using Equation 6:

$$PRIO^i_{MTCR} = PRIO^i_{dist} + HCAP^i \cdot 1000 + \frac{i}{1000}, \tag{6}$$

where $i$ is the aircraft index in the traffic list, $PRIO^i_{dist}$ is the distance-based priority of aircraft $i$, and $HCAP^i$ is the handicap value of aircraft $i$ [15, 16]. The distance-based priority $PRIO^i_{dist}$ is computed using Equation 7:

$$PRIO^i_{dist} = \max\left(1, 100 - \frac{100}{d_{max}} \cdot \left\lfloor \frac{d^i_{dest}}{10} \right\rfloor \cdot 10\right) \tag{7}$$

where $d^i_{dest}$ is the distance to the destination in nautical miles, $d_{max} = 10000$ is the maximum distance parameter, and $\lfloor \cdot \rfloor$ denotes the floor function [15, 16]. The $PRIO^i_{dist}$ is discretized into 10 NM increments and ranges from 1 to 101, with higher values indicating higher priority. The floor operation with $10NM$ increments ensures that the distance-based priority remains constant within each $10NM$ range, preventing frequent priority changes due to small distance variations [15]. The term $i/1000$ ensures unique priorities by breaking ties between aircraft with equal distance-based priorities [15].

When $HCAP^i > 0$, the aircraft is designated as high priority (e.g., emergency or VIP), and its MTCR priority is increased by $HCAP^i \cdot 1000$, ensuring it has priority over normal aircraft [15, 16]. In this paper, it is assumed all aircraft start without a handicap ($HCAP^i = 0$) and there are no VIP aircraft ($HCAP^i = 2$). If an aircraft fails to resolve a conflict using a 5 NM radius of the PAZ, it will try to generate a resolution with the least amount of undershoot of the PAZ [12, 15]. However, as that would still lead to a loss of separation, the aircraft will increase its priority by being 'handicapped' ($HCAP^i = 1$) [12, 15]. Other aircraft will then be forced to help resolve the conflict and make room for the 'handicapped' aircraft. The handicap is reset when the aircraft is out of conflict [15].

The MTCR generates a conflict free route up to 15 minutes ahead [12, 15]. This conflict free route consists of the route up to that point which is still applicable, two waypoints that are added to resolve the conflict, and at the end the goal of the aircraft. The first of the two waypoints is a TCP marking a turn away from the conflict with a maximum turn angle of 60° [15, 16]. The second waypoint is the TCP marking the turn back to goal with a maximum turn angle of 45° [15, 16]. A schematic of such a resolution trajectory is shown in Figure 3.

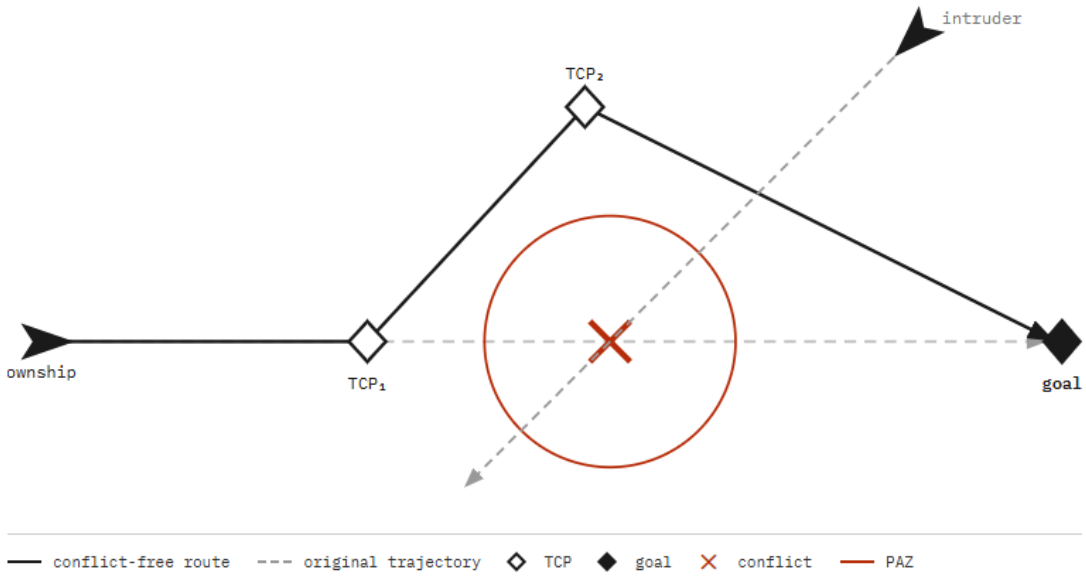



Figure 3: Schematic of the MTCR resolution trajectory

The MTCR resolution algorithm computes the two waypoints through a two-step iterative process that searches for conflict-free trajectories while minimizing deviation from the planned route [16]. In Step 1, the algorithm determines the start-of-turn point (SoT1) based on the current intent and identifies the velocity vector at this point. In Step 2, the algorithm performs a dual-loop search: the outer loop iterates over possible turn angles $\Delta\phi_{res}$ from 0° to the maximum turn angle $\Delta\phi_{max} = 60°$ in 1° increments, while for each turn angle, an inner loop searches over time intervals $\Delta t_{WP2}$ starting from $10s$ in $15s$ increments [16]. For each candidate turn angle, the first waypoint (WP1) is computed using the turn radius $R_{turn}$ determined by

the bank angle ($\mu_{bank} = 25°$) and ground speed. The position of WP1 ($\mathbf{x}_{WP1}$) is calculated using Equation 8:

$$\mathbf{x}_{WP1} = \mathbf{x}_{SoT1} - \lambda \cdot \mathbf{a}, \tag{8}$$

where $\mathbf{a}$ is the normalized velocity at start-of-turn of a waypoint, $\mathbf{x}_{SoT1}$ the point at which the aircraft can start the first turn and $\lambda$ the turn geometry parameter [15, 16]. $\lambda$ is defined by Equation 9:

$$\lambda = \frac{R_{turn}}{\mathbf{b}^T\mathbf{c}}(\mathbf{b} + \mathbf{a})^T\mathbf{a}, \tag{9}$$

where $\mathbf{b}$ is the normalized velocity at the end-of-turn after rotation by $\Delta\phi_{res}$, and $\mathbf{c}$ is the perpendicular direction vector of $\mathbf{a}$ such that $\mathbf{c} \perp \mathbf{a}$ [15, 16].

The second waypoint (WP2) is positioned at a distance $\mathbf{v}_{EoT1} \cdot \Delta t_{WP2}$ along the straight segment following the first turn, where $\mathbf{v}_{EoT1}$ is the velocity at the end of the first turn. For each WP2 candidate, the algorithm validates seven geometric constraints: (1) the turn at WP2 must reach the goal within the maximum back-to-goal angle ($\Delta\phi_{B2G,max} = 45°$), (2) the total detour distance must not exceed $15NM$, (3) the heading change at WP2 must be feasible, (4-7) proper spacing between turn segments must be maintained [16]. the complete trajectory formed by WP1, WP2, and the destination is then checked for conflicts with lower priority aircraft intents using their broadcast $Q_{turn}$ trajectories up to 15 minutes ahead. This is done using the DVO approach described in section 2.1.2 [12, 15]. Additionally, the MTCR is not allowed to turn into a short-term conflict with any aircraft. This is also checked with all other aircraft intents, using the DVO approach described in section 2.1.2 up to 3 minutes and 10 seconds ahead. The first combination of $\Delta\phi_{res}$ and $\Delta t_{WP2}$ that satisfies all geometric constraints and produces a conflict-free trajectory is selected. After the pilot has approved of the computed resolution, the new $Q_{turn}$ of the aircraft is broadcast to all other aircraft, showing it has resolved the conflict.

**Short-Term Conflict Resolution**

If the conflict is less than three minutes away, STCR is used. In this case, both aircraft involved in the conflict are responsible to resolve. Instead of generating a completely new route, STCR computes a conflict free trajectory up to three minutes and ten seconds ahead [12]. This conflict free trajectory is a minimum heading change to avoid the conflict and not generate new short term conflicts [12]. The maximum heading change is 60° [15, 16]. If the STCR is unable to compute any conflict free trajectories, it will try to find a resolution with the least amount of undershoot of the PAZ.

Unlike MTCR, STCR operates on the aircraft's current state vector rather than its planned intent trajectory, allowing for rapid response to imminent conflicts. The algorithm iterates over turn angles from 0° to $\Delta\phi_{max} = 60°$ in 0.5° increments, evaluating both left and right turns. For each candidate angle $\Delta\phi_{res}$, the algorithm simulates the turn trajectory using the same bank angle ($\mu_{bank} = 25°$) as MTCR and checks for conflicts with all other aircraft up to three minutes and 10 seconds ahead. This is done using the DVO approach described in section 2.1.2 [12, 15].

When the STCR is activated it waits 10 seconds after which a trajectory is computed [15].[1] If an aircraft fails to resolve a conflict using a 5 NM radius of the PAZ, it will try to generate a resolution with the least amount of undershoot of the PAZ [12, 15]. After the pilot has approved of the trajectory the temporary $Q_{turn}$, containing only the turn trajectory, is broadcast to all other aircraft.

While STCR is computing a trajectory, MTCR is generating a conflict free back to goal route up to 15 minutes ahead. This route is not allowed to undershoot the PAZ, and has a

[1] In hindsight, according to [15, 16] this $10s$ delay only applies for conflicts detected using state-based STCD.

maximum turn angle of 60° and back to goal turn angle of 90°. After the STCR is approved by the pilot, the back to goal MTCR is presented for approval. When it is approved, it will be broadcast as the new $Q_{turn}$ to all other aircraft.

**Divergence**

When a conflict resolution is computed that results in a heading nearly parallel to the current heading, there is a risk that multiple aircraft involved in the same conflict may execute similar resolution manoeuvers, potentially creating new conflicts or maintaining parallel trajectories. To prevent this, the A3 model implements a divergence mechanism [16].

The divergence logic is triggered when the computed resolution heading is nearly parallel to the aircraft's current heading and the computed resolution will undershoot the PAZ [16]. This is determined by computing the cosine of the angle difference between the two headings. If this value exceeds 0.9994, which corresponds to an angular difference of approximately 2° or less, the divergence mechanism is activated.

When divergence is activated, the model evaluates three potential heading adjustments: maintaining the current heading, turning left by 5°, and turning right by 5° [16]. For each of these options, the minimum separation distance to other aircraft is predicted one second ahead. The heading adjustment that provides the maximum minimum separation is selected as the divergence angle.

This divergence angle is then applied as an initial offset when recomputing the conflict resolution [16]. The resolution algorithm searches for a conflict-free trajectory starting from this diverged heading, ensuring that aircraft will have trajectories that diverge rather than remain parallel or converge.

The divergence logic is applied during both the MTCR and STCR trajectory computation phases, before the resolution is presented to the pilot for approval [16].[2]

## 2.2 Modified Voltage Potential

To put the performance of the A3 model in context, it is compared to the MVP conflict resolution method. The method uses state-based conflict detection, which is elaborated upon in section 2.2.1. The numerical implementation of the MVP resolution method is described in section 2.2.2.

### 2.2.1 Conflict Detection

state-based detection is performed using a linear propagation of the current state of all involved aircraft [6]. This is similar to conflict detection in the A3 model, except there are no intents. Knowing the current state and velocity of other aircraft, the aircraft knows the relative distance vector and relative velocity vector [6]. Using this, the time to the CPA can be computed as in Equation 3 [6]. The distance at the CPA is then calculated with Equation 4 [6]. The time at which the aircraft enters the conflict can be calculated when the computed separation distance is smaller than the radius of the PAZ, as is shown in Equation 5 [6].

If $t_{in}$ is smaller than the look ahead time and $||\mathbf{d}_{CPA}||$ is smaller than the radius of the PAZ a conflict is detected. the look ahead time of the state-based detection is $t_{lookahead} = 5min$ [6].

### 2.2.2 Conflict Resolution

The MVP resolution method represents a force field approach to conflict resolution, where conflicting aircraft positions are modelled as charged particles that simultaneously repel each

[2]In hindsight, according to [16] this divergence does not apply to MTCR.

other. In this method, the predicted positions of aircraft at their CPA generate a repelling force that is converted into a displacement vector, ensuring the minimum distance equals the required separation [6]. More specifically, when a conflict is detected an avoidance vector is calculated from the aircraft's predicted future position at the CPA to the edge of the PAZ, in the direction of the minimum distance vector [4]. The length of this avoidance vector is the horizontal intrusion $i_H$ of the PAZ, and is computed by Equation 10:

$$i_H = R_{PAZ} - ||\mathbf{d}_{CPA}||. \tag{10}$$

By dividing this length by the time remaining until conflict, an advised change in speed is obtained. When vectorized in the direction of the minimum distance vector, this gives an advised change in velocity as computed by Equation 11:

$$d\mathbf{v} = \frac{i_H}{|t_{CPA}|} \cdot \frac{\mathbf{d}_{CPA}}{||\mathbf{d}_{CPA}||}, \tag{11}$$

where $d\mathbf{v}$ is the change in velocity vector [17].

When added to the current velocity, it yields a recommended track and ground speed [4]. A key advantage of MVP is its implicit coordination: since each aircraft calculates resolutions assuming the other will not manoeuver, the conflict geometry naturally produces complementary evasive actions in opposite directions, achieving effective cooperation without negotiation or additional communication [6, 4]. For multi-aircraft conflicts, avoidance vectors are simply summed for each conflict pair [6]. This geometric approach offers computational simplicity, though resolutions may sometimes oppose the desired flight plan direction [6].

# 3 Experimental Design

To compare the performance of the intent-based and state-based conflict resolution methods, several experiments are designed. First, to show and compare the basic behaviour of the resolution methods, two-aircraft scenarios are described in section 3.1. Then, to increase the complexity, the amount of aircraft in conflict is increased to eight. This scenario is elaborated upon in section 3.2. Next, the performance of the conflict resolution methods is tested by simulating random traffic scenarios. These experiments are explained in section 3.3. Finally, the number of simulation runs and their variability is elaborated upon in section 3.4.

## 3.1 Two-Aircraft Scenarios

To verify and compare the conflict resolution trajectories, two-aircraft scenarios are designed to test specific conflict geometries. Two distinct scenarios are evaluated: a head-on encounter in section 3.1.1 and a nearly parallel crossing trajectory in section 3.1.2.

### 3.1.1 Head-on Encounter

A head-on encounter scenario is created by placing two aircraft on opposite sides of a center point, $200NM$ apart, with their destinations directly across from each other. The aircraft fly directly toward each other, creating a symmetric head-on conflict. Both aircraft fly at $36,000ft$ altitude and $250kts$ calibrated airspeed (CAS). This configuration results in a direct head-on conflict, testing the model's ability to handle basic conflicts. The initial position of this scenario is presented in Figure 4, showing AC1 and AC2 and their corresponding destination waypoints.


Figure 4: Head-on encounter between AC1 and AC2, with corresponding waypoints as destinations.

### 3.1.2 Nearly Parallel Crossing Trajectories

To specifically test the divergence logic, a scenario with nearly parallel but crossing trajectories is designed. Two aircraft are positioned approximately $10NM$ apart laterally at the same longitudinal position, both with an initial heading of 90°. Their destinations are placed $200NM$ to the east. The destinations are also $10NM$ apart laterally but on opposite sides creating crossing paths with a small angular difference of approximately 2°. Both aircraft fly at $36,000ft$ altitude and $250kts$ CAS. This configuration creates a situation where the resolution trajectories are likely to be nearly parallel to the original headings, triggering the divergence mechanism described in section 2.1.3. The initial position of this scenario is presented in Figure 5, showing AC1 and AC2 and their corresponding destination waypoints.


Figure 5: Nearly parallel crossing encounter between AC1 and AC2, with corresponding waypoints as destinations.

## 3.2 Eight-Aircraft Scenario

To evaluate and compare the conflict resolution behaviour in a more complex multi-aircraft conflict situation an eight-aircraft scenario is used. Eight aircraft are positioned in a circle from a central point at a radius of $100NM$, arranged in a circular pattern with 45° angular spacing between adjacent aircraft. Each aircraft has its destination $200NM$ directly opposite its starting position across the center point, resulting in all aircraft converging toward the center of the cirle. All aircraft fly at $36,000ft$ altitude and $250kts$ CAS. The initial position of this scenario is presented in Figure 6, showing a circle of eight aircraft and their corresponding destination waypoints.

## 3.3 Random Traffic Scenarios

Fast-time random traffic simulation experiments are conducted to evaluate the performance of the conflict resolution models. First, the simulation design is introduced in section 3.3.1, after which the random traffic generation is elaborated upon in section 3.3.2. Finally, the performance measures for the evaluation of the model are introduced in section 3.3.3.

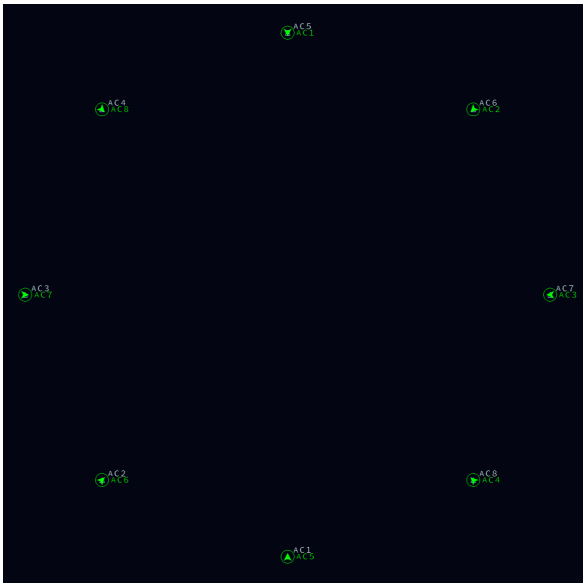


Figure 6: Eight aircraft (AC1-AC8) scenario, with corresponding waypoints as destinations.

### 3.3.1 Simulation Design

The simulation design is similar to the review performed by Ribeiro et. al (2020) [6]. The aircraft type used during the simulations is a Boeing 747-400. All aircraft fly at constant velocity in the range of $v \in [450, 500] kts$ true airspeed (TAS), constant altitude $H = 36,000 ft$, and have a constant bank angle $\mu = 25°$ when making a turn. The duration of the simulations is 3 hours, with an experiment duration of 1.5 hours to measure the performance of the model. This means that all conflicts, losses of separation, and flight parameters of aircraft inside a defined measurement area are logged during this experiment duration. However, to make sure that when an aircraft leaves this measurement area accidentally it is not deleted, an experiment area is defined around the measurement area.

All aircraft are spawned at the edge of this experiment area to prevent conflicts in the measurement area due to spawning. Furthermore, each aircraft spawned has a destination such that the minimum flight time through the measurement area is $30 min$. However, to make sure there is a range of different headings, a longer flight time is preferred. Thus, the average flight time is set to $40 min$.

Assuming an average velocity of $475 kts$, the length of the side of the measurement area is computed by finding the distance an average aircraft will fly. This is given in Equation 12. From squaring the length, the size of the measurement area is then found in Equation 13.

$$L_{meas} = \frac{475[kts]}{60[min]} 40[min] = 316.7[NM] \tag{12}$$

$$A_{meas} = L_{meas}^2 \approx 100,000[NM^2] \tag{13}$$

As the A3 CD&R model has a lookahead time of $10 min$, the time it takes to fly from the spawning point at the edge of the experiment area to the edge of the measurement area should be more than $10 min$. The maximum velocity is $500 kts$ TAS, which means the distance from the edge of the experiment area to the edge of the measurement area can be computed as in Equation 14.

$$D_{exp2meas} = \frac{500[kts]}{60[min]} 11[min] = 91.7[NM] \tag{14}$$

This is then doubled and added to the length of the side of the measurement area, finding the length of the side of the experiment area using Equation 15. The size of the experiment area is then found by squaring the length, as can be seen in Equation 16

$$L_{exp} = L_{meas} + 2 \cdot D_{exp2meas} = 500[NM] \tag{15}$$

$$A_{exp} = L_{exp}^2 = 250,000[NM^2] \tag{16}$$

To avoid aircraft converging on a destination, leading to difficult conflicts, destinations are placed far outside the experiment area. For this, the size of the experiment area is doubled. This makes sure that aircraft, when almost leaving the experiment area and being deleted, do not enter a difficult to solve conflict stalling the simulation. A schematic of the simulation design is shown in Figure 7.

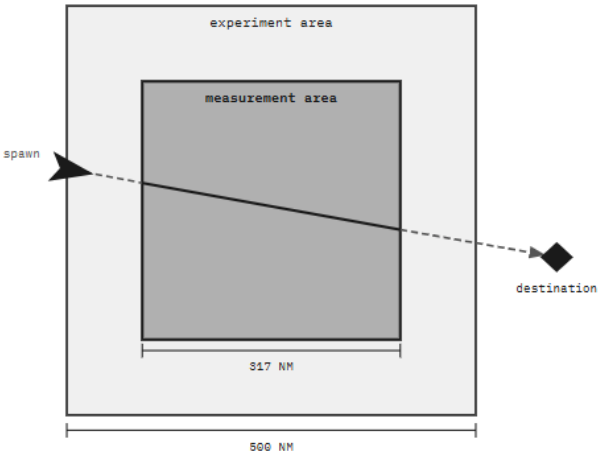



Figure 7: Schematic of simulation design.

### 3.3.2 Random Traffic

For the random traffic scenarios, a certain traffic density at a single flight level is to be reached to test the performance. In 2017, traffic density above the Netherlands had a maximum of 32 aircraft per $10.000NM^2$ [13]. However this number will increase in the future [3]. In the safety evaluation by Blom and Bakker (2015), traffic densities per flight level of 20.8 and 41.6 aircraft per $10,000NM^2$ were simulated, although there a different random traffic generation was used [12]. Additionally, in the review by Ribeiro et al. (2020), traffic densities of 32, 37, and 45 aircraft per $10,000NM^2$ were simulated. Therefore, in this paper similar traffic densities are used. In three different scenarios, the density will increase from a low level of 35 aircraft per $10,000NM^2$, to a medium level of 40 aircraft per $10,000NM^2$, up to a high level of 45 aircraft per $10,000NM^2$.

To reach the required traffic density, the first 45 minutes of the scenario are used to build up the amount of aircraft. The scenario starts with no aircraft. Aircraft are added to the scenario either when the required traffic density is not reached or when the traffic density is expected to drop below the required level.

When the required traffic density is not reached, aircraft spawn at a time interval determined by the average flight time and the required amount of aircraft inside the measurement area. This time interval is multiplied by a factor of 0.75 to speed up spawning when below the required density level. This factor is determined empirically.

When the required density level is reached, the time interval is determined by the expected time at which the traffic density will drop below the required level. This way the traffic density will not overshoot the required level, as the spawning of aircraft is paused until it is expected to fall below the required level. This takes into account the average time it takes aircraft to reach the measurement area after spawning.

After the experiment duration, so after $45min$ build up phase and $90min$ experiment, no new aircraft are spawned and the scenario runs an additional $45min$ to build down. The scenario is then terminated and the data logged.

This traffic generation predetermines when and where each aircraft will spawn. This means that the spawning of aircraft is not determined by the actual traffic density during the simulation runs, which might change due to conflict resolution trajectories. A more in detail description of how these scenarios are created is given in Appendix B.

### 3.3.3 Performance Measures

The performance of a conflict resolution method is mainly described by its ability to resolve detected conflicts and prevent losses of separation. The frequency of conflicts $f_{cfl}$ is defined to be the sum of the detected conflicts per aircraft per flight hour. This means that if a pair of aircraft $(AC1, AC2)$ detect a conflict, it counts as a conflict for $AC1$ and as a conflict for $AC2$. The frequency of loss of minimum separation $f_{LoS}$ is defined to be the sum of the losses of separation per aircraft pair per flight hour. Furthermore, to show how well the model prevents larger intrusions of the PAZ, the frequency of loss of separation below $2/3R_{PAZ}$ $f_{2/3LoS}$ is measured and compared [12]. Additionally, the amount of horizontal intrusion $i_H$ as a percentage of the radius of the PAZ $R_{PAZ}$ gives insight in the performance of the conflict resolution method. This intrusion percentage $I$ is computed using Equation 17 [6]:

$$I = \frac{i_H}{R_{PAZ}}. \tag{17}$$

Additionally, the performance of a conflict resolution method can be measured by its tendency to create new conflicts. The domino effect parameter ($P_{DE}$) is the frequency of conflict with conflict resolution compared to the frequency of conflict without. A higher $P_{DE}$ signals a higher instability of the airspace, thus conflict resolution leading to more conflicts. The $P_{DE}$ is shown in Equation 18:

$$P_{DE} = \frac{f_{cfl}^{ON} - f_{cfl}^{OFF}}{f_{cfl}^{OFF}}, \tag{18}$$

with $f_{cfl}^{ON}$ and $f_{cfl}^{OFF}$ the frequency of conflict detected while conflict resolution is on and off, respectively [6].

Lastly, the performance of a conflict resolution method can be determined by its flight efficiency. The efficiency is measured by the work done during the simulation, that is flight effort. This does not include work (labour) activities performed by pilots. If all aircraft fly their trajectories straight, this is minimal. However due to conflict resolution, additional flight effort is required to fly to the destination. The work done $W$ is computed as in Equation 19:

$$W = \int_{path} \mathbf{T} \cdot d\mathbf{s}, \tag{19}$$

with $\mathbf{T}$ the thrust vector and $\mathbf{s}$ the displacement vector [6].

## 3.4 Simulation Variability

Due to computational effort, each of the above mentioned scenarios is run once for both the A3 model and the MVP resolution method. Each simulation is run with a timestep of $\Delta t = 0.1s$. The MVP resolution method resolves detected conflicts within that timestep.

However, there is some variability in the A3 model. In the A3 model, the conflict resolution execution is delayed by the pilot decision time. This means that both MTCR and STCR is delayed by a random generated time before actually resolving the conflict. The MTCR pilot decision time is approximated by a Rayleigh distribution with $\mu = 30s$. The STCR pilot decision time is approximated by a Rayleigh distribution with $\mu = 5.7s$. For the back to goal MTCR performed in STCR, the pilot decision time is approximated like the MTCR pilot decision time. Additionally, if there is no possible back to goal MTCR, the aircraft will retry to generate a conflict free route after a delay. This delay is determined by a Rayleigh distribution with $\mu = 20s$.

As mentioned before, no altitude changes are considered. The MVP resolution method does use speed change to resolve conflicts, whereas the A3 model only considers trajectory changes. Additionally, no wind or performance uncertainties are considered. This is different to the simulations performed with the A3 model in the safety evaluation by Blom and Bakker (2015) [12]. There monte carlo simulations were performed with both wind and performance uncertainties.

# 4 Results

Following the methodology, the results are presented in similar order. First the results of the two aircraft scenarios are shown in section 4.1. Then the eight aircraft scenario results are presented in section 4.2. Finally, the results of the random traffic simulations are presented in section 4.3.

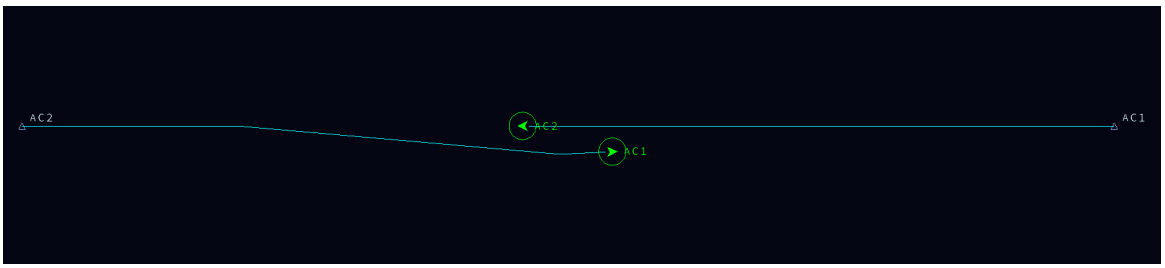
## 4.1 Two-Aircraft Scenarios

The results of the head-on encounter scenario are presented in section 4.1.1. The results of the nearly parallel crossing trajectories scenario are presented in section 4.1.2.

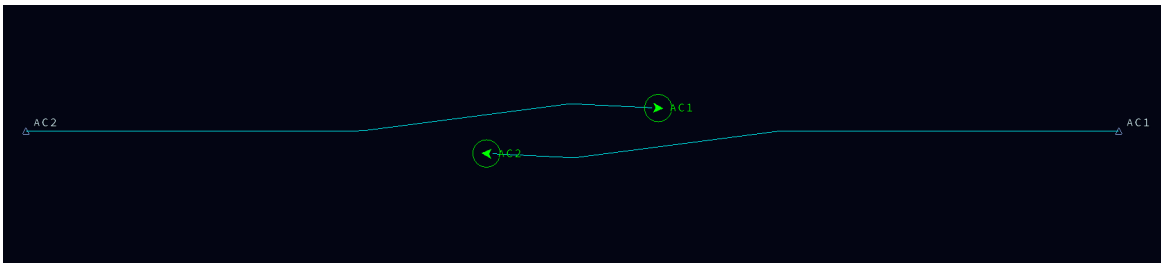
### 4.1.1 Head-on Encounter

Figure 8 illustrates the conflict resolutions for a head-on encounter between AC1 and AC2. Figure 8a shows the conflict resolution performed by the A3 model. As both aircraft are equally far from their destination, AC2 has higher priority based on its index. The CD&R model successfully generates an asymmetric avoidance manoeuver, with AC1 executing a right-turn to avoid AC2. The trajectory then clearly shows a back-to-goal turn after passing AC2.

Figure 8b illustrates the conflict resolution performed by the MVP resolution method. Clearly, both aircraft have moved to resolve the conflict, and after the conflict have turned back to their respective destinations. Due to the shorter look-ahead time of the state-based conflict detection, the resolution using the MVP method starts significantly later than using the A3 model leading to a larger resolution heading change. Additionally, since both aircraft resolve the full conflict using the MVP method, more distance is travelled in total. In contrast, using the A3 model only AC1 resolves the conflict, leaving AC2 to fly its most efficient route.




(a) A3




(b) MVP

Figure 8: Head-on conflict resolution trajectory

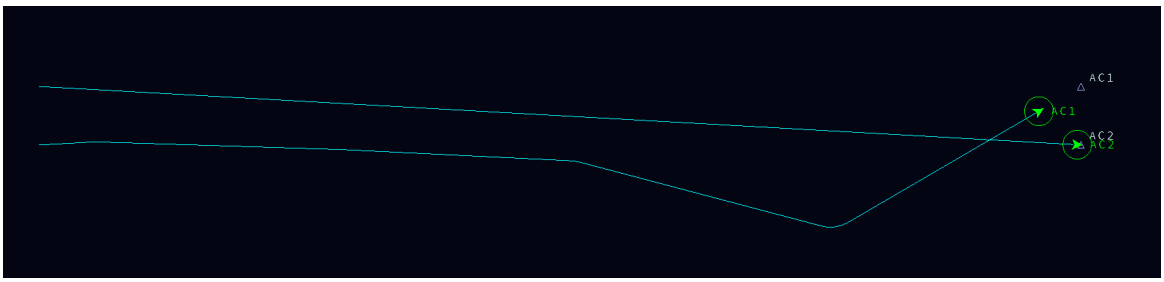
### 4.1.2 Nearly Parallel Crossing Trajectories

The nearly parallel crossing scenario in Figure 9 demonstrates conflict resolutions for converging trajectories at a shallow crossing angle. Figure 9a shows the resulting resolution trajectory using the A3 conflict resolution model. Similar to the head-on encounter, the aircraft are equally far from their destination. Thus AC2 again has higher priority based on its index. AC2 indeed does not make a manoeuver, whereas AC1 is clearly seen abandoning its original heading, flying more parallel to AC2. AC1 now has computed a back-to-goal waypoint 15 minutes ahead, but after 5 minutes detects that this intent is still in conflict with AC2. Since it cannot find a conflict free resolution without losing separation to still reach its goal, it tries to diverge. This divergence can clearly be seen as a turn to the right away from AC2. This creates manoeuvering space to then turn back to goal behind AC2.

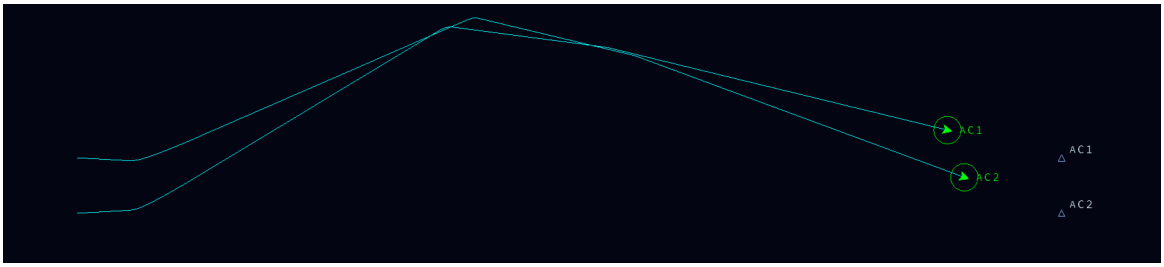
Figure 9b shows the resulting resolution trajectory using the MVP resolution method. As can be seen, quite a large resolution trajectory is required by both aircraft to resolve the conflict. It is not visible on the image, but worth noting that both aircraft slowed down significantly. Both aircraft thus needed more time and flew a larger distance to reach their destinations.

## 4.2 Eight-Aircraft Scenario

To assess the algorithm's performance under more challenging conditions, a symmetric eight-aircraft conflict scenario was simulated. This is done with both the A3 model and the MVP resolution method. Figure 10 demonstrates the trajectories of both the A3 model and the MVP method. Figure 10a illustrates the resolution trajectories using the A3 resolution model. Similar to the two aircraft scenarios, each aircraft has the same distance to its destination. Priority is thus based on the index, with AC8 having highest priority going counter clockwise to AC1 having the lowest priority. This is also visible in the figure, where AC1 performs a larger avoidance manoeuver whereas AC8 can fly straight to its destination and does not manoeuver at all. However, it is noticable that AC4 performs a smaller avoidance manoeuver than AC7, whereas priority dictates the other way around. This is both due to conflict geometry and pilot decision timing. Here, AC7 in its effort to avoid AC8 has to fly a longer route to cross back to goal behind AC8. AC4, deciding on its resolution later, profits by being able to avoid AC7




(a) A3

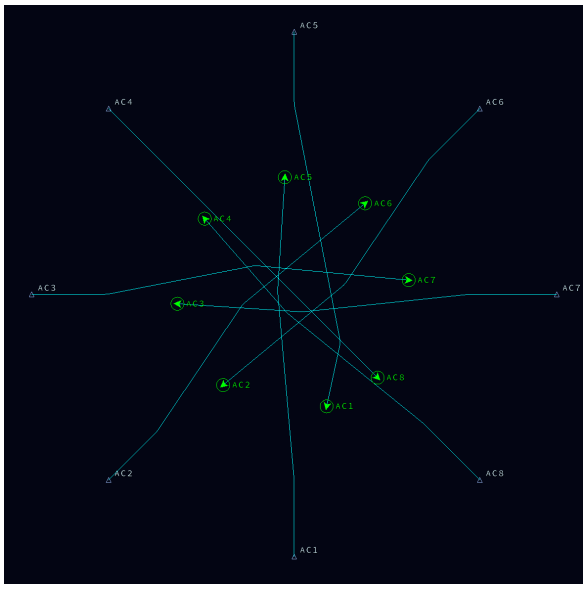



(b) MVP

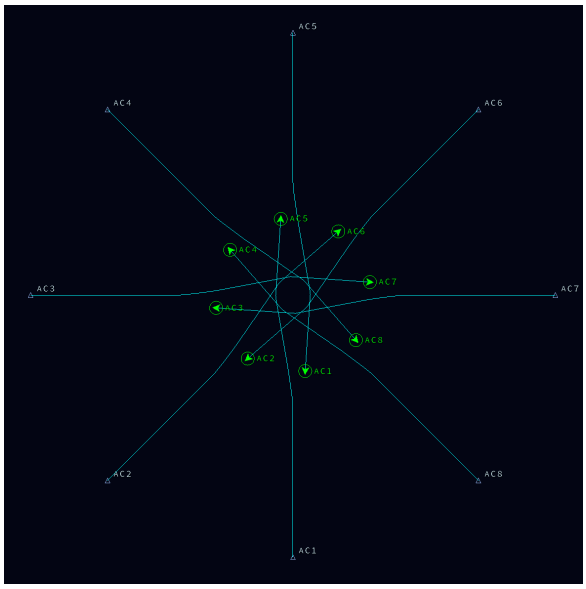
Figure 9: Nearly parallel crossing conflict resolution trajectory

more easily and having a head-on encounter with AC8, thus being able to fly a shorter route. As all aircraft are able to resolve with MTCR, no STCR is performed in this scenario.

Figure 10b shows the resolution trajectories using the MVP method. Similar to the head-on scenario, aircraft are seen resolving conflicts later as the state-based conflict detection has a smaller look-ahead time. However, where this lead to slightly larger resolution heading changes in the head-on scenario, that is not the case in this scenario. As no aircraft has priority, all aircraft work together to resolve the conflicts. Where using the A3 model, AC1 thus makes a large detour avoiding all other aircraft, using the MVP method all aircraft fly equal distances. In this case, no aircraft slowed down to resolve the conflicts.




(a) A3




(b) MVP

Figure 10: Eight-aircraft scenario conflict resolution trajectories

## 4.3 Random Traffic Scenarios

The results of the random traffic scenarios are presented according to the performance measures introduced in section 3.3.3. But first, the actual density during simulations is presented in section 4.3.1. Secondly, loss of separation related results are shown in section 4.3.2. Finally, the efficiency results are presented in section 4.3.3.

### 4.3.1 Density

The density analysis across different traffic scenarios reveals a significant difference between the two resolution methods tested. In Figure 11 the density during the simulations is plotted over time, with the average density and its standard deviation for each simulation shown in the legend.

Figure 11 shows that with the MVP method the density increases significantly. All three scenarios show a significant rise in aircraft inside the measurement area during the simulations using MVP method. This can be due to the resolution method using speed change, in this case slowing down, to resolve conflicts. The additional effect is that aircraft remain longer inside the area, artificially increasing the density.

Figure 11 also shows that, in contrast to the MVP method, the simulations using the A3 conflict resolution model have a significantly lower density than expected. A possible explanation for this are resolutions pushing aircraft out of the measurement area. If aircraft continue to their destination without re-entering the measurement area, this is measured as a decrease in density. Another possible explanation is the avoidance of high density areas by lower priority aircraft. As can be seen in the eight-aircraft scenario, the lower priority aircraft make larger detours to avoid all higher priority aircraft, flying around the busy area. If this effect keeps aircraft outside of the measurement area, this is also measured as a decrease in traffic density.

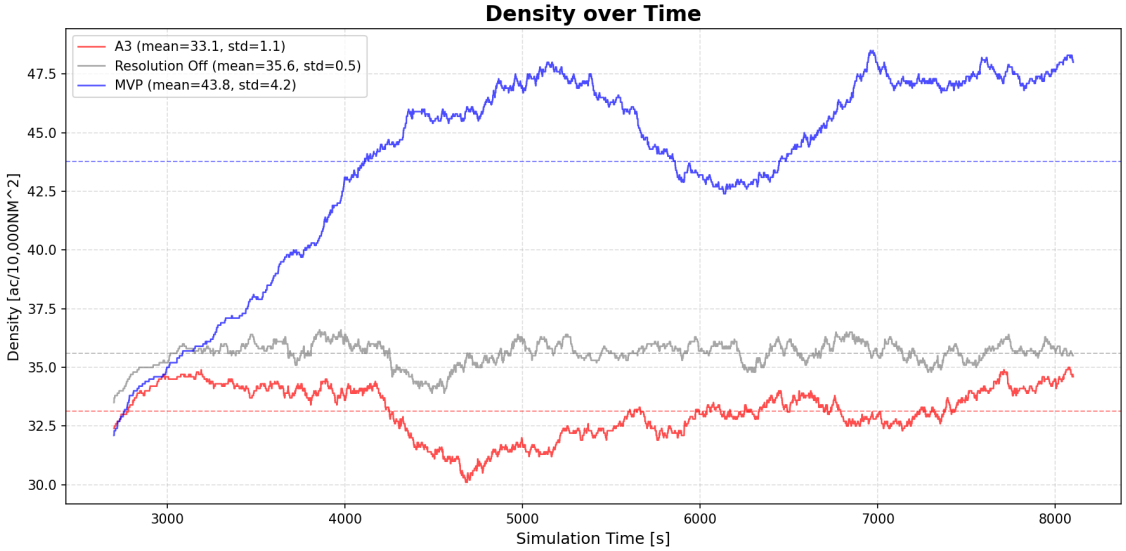
### 4.3.2 Loss of Separation

The frequency of loss of separation is shown in Figure 12. As can be seen in Figure 12a, the loss of minimum separation frequency is significantly lower for the A3 conflict resolution model. Additionally, this seems to increase with increasing density suggesting a correlation. However, interestingly the loss of 2/3 separation frequency shown in Figure 12b is higher for the A3 model, where the MVP method has no losses of 2/3 separation.

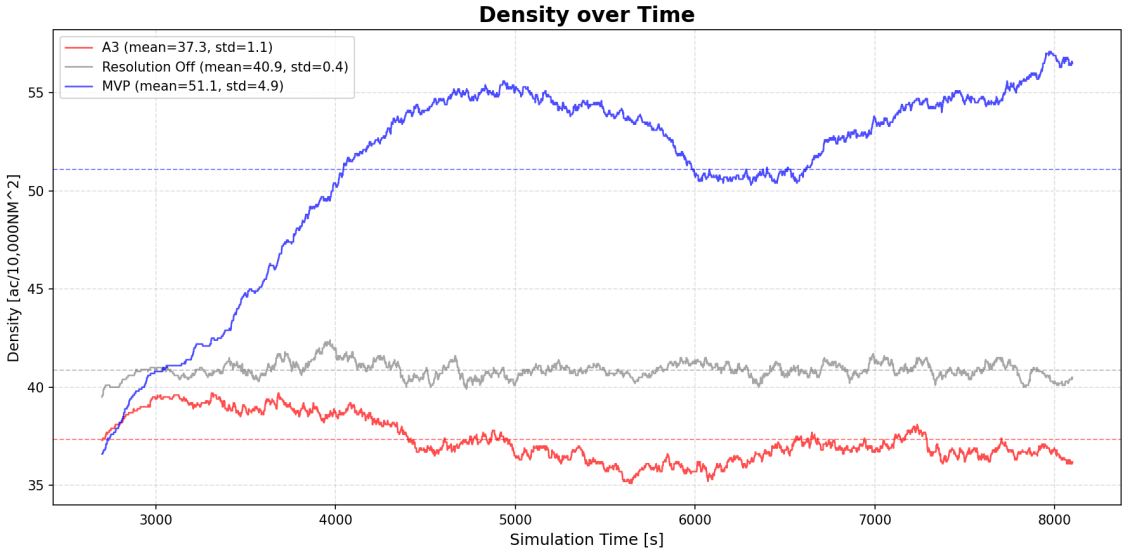
The intrusion distribution in Figure 13 also shows that the intrusions recorded are larger when using the A3 model. Across all densities it shows outliers far inside the 2/3 separation line, as can be seen in Figure 12b. But also for the losses of minimum separation, the intrusions using the A3 model are significantly higher than those of the MVP method. In contrast to the frequency of and loss of minimum separation, here no correlation is seen between the traffic density and the amount of intrusion.

The conflict frequency can be seen in Figure 14. When examining Figure 14, it reveals that the MVP method has a significantly higher conflict frequency than the A3 model. This difference seems to increase in the higher traffic density scenarios. This may be caused by the increase in density using the MVP method leading to a greater amount of aircraft and thus a greater amount of potential conflicts in the measurement area. It may also indicate that the MVP resolution method is less effective at preventing secondary conflicts. This domino effect has also been seen by Sunil et al. and by Ribeiro et al. [6, 13].

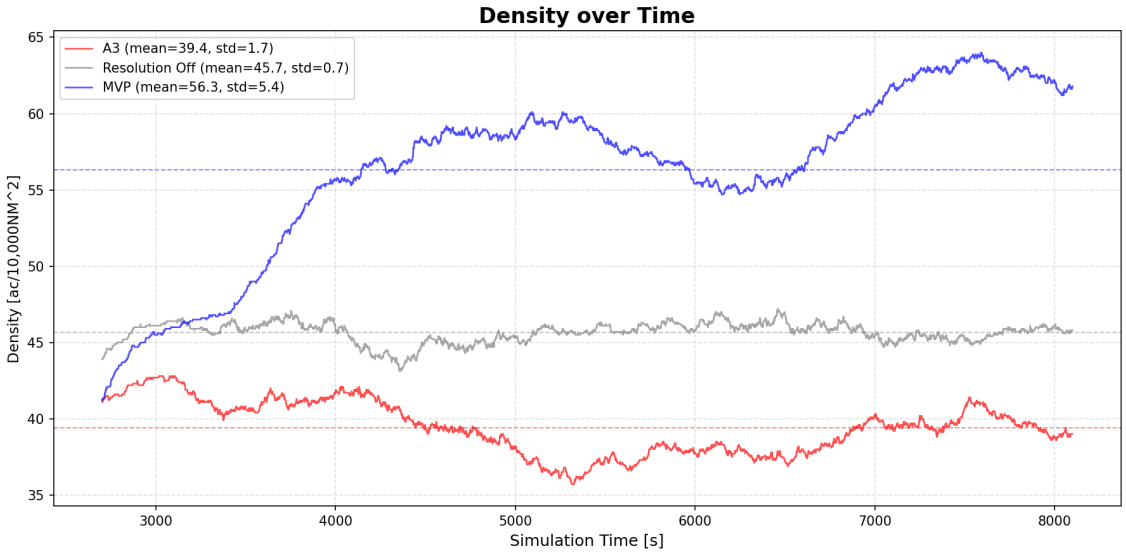
The domino effect parameter is a quantification of this domino effect, presented in Figure 15. Similar to the conflict frequency results shown in Figure 14, the MVP resolution method performs worse in preventing cascading conflicts. The A3 model performs significantly better,




(a) Density $35[ac/10,000^2]$

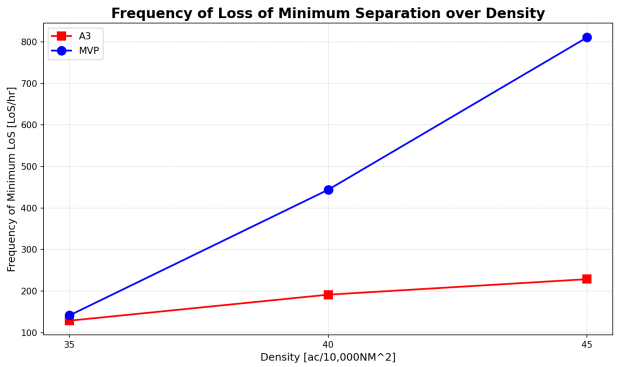



(b) Density $40[ac/10,000^2]$

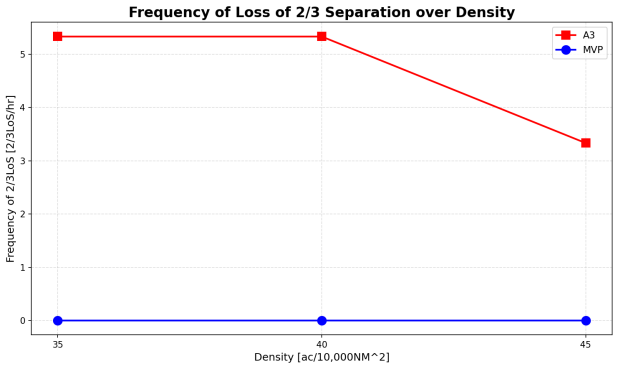



(c) Density $45[ac/10,000^2]$

Figure 11: Density over time per random traffic scenario




(a) Frequency of loss of minimum separation




(b) Frequency of loss of 2/3 separation

Figure 12: Frequency of loss of separation

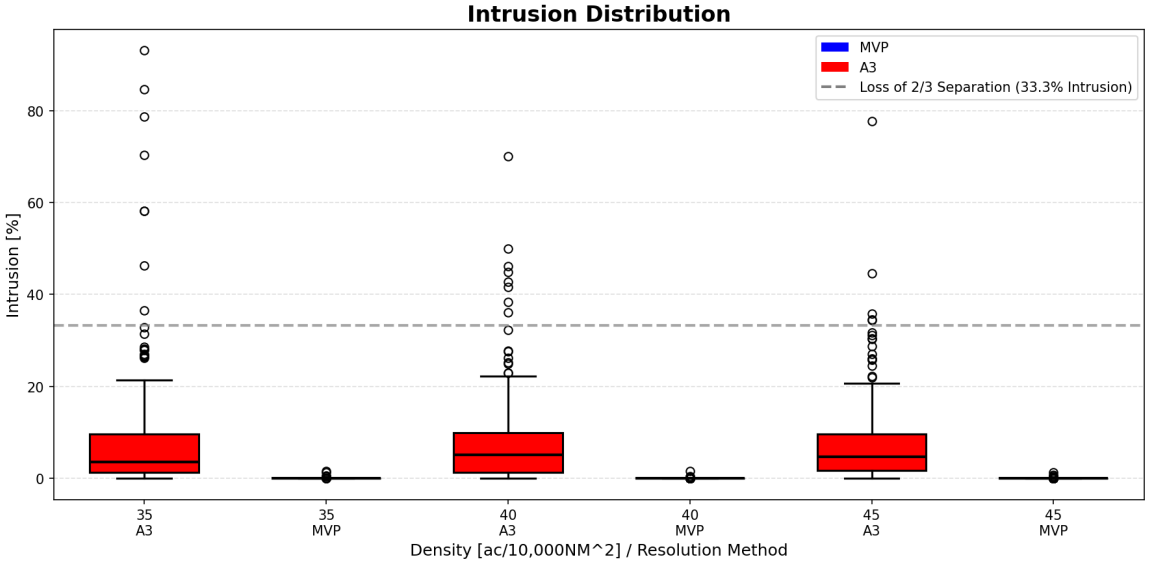



Figure 13: Intrusion distribution.

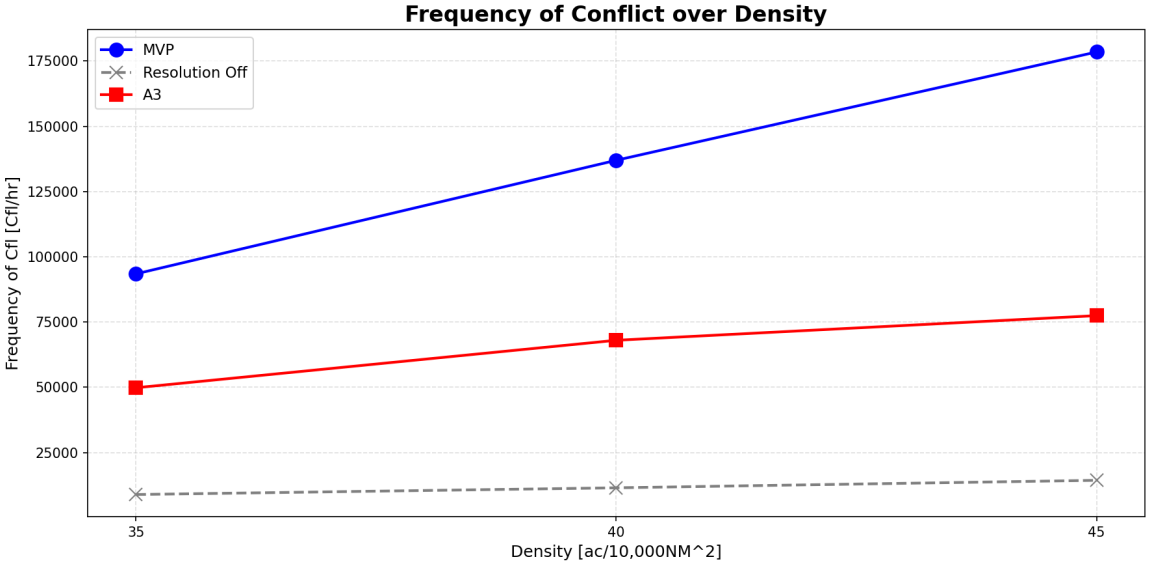



Figure 14: Frequency of conflict.

which may be due to its longer look-ahead time when using MTCR.

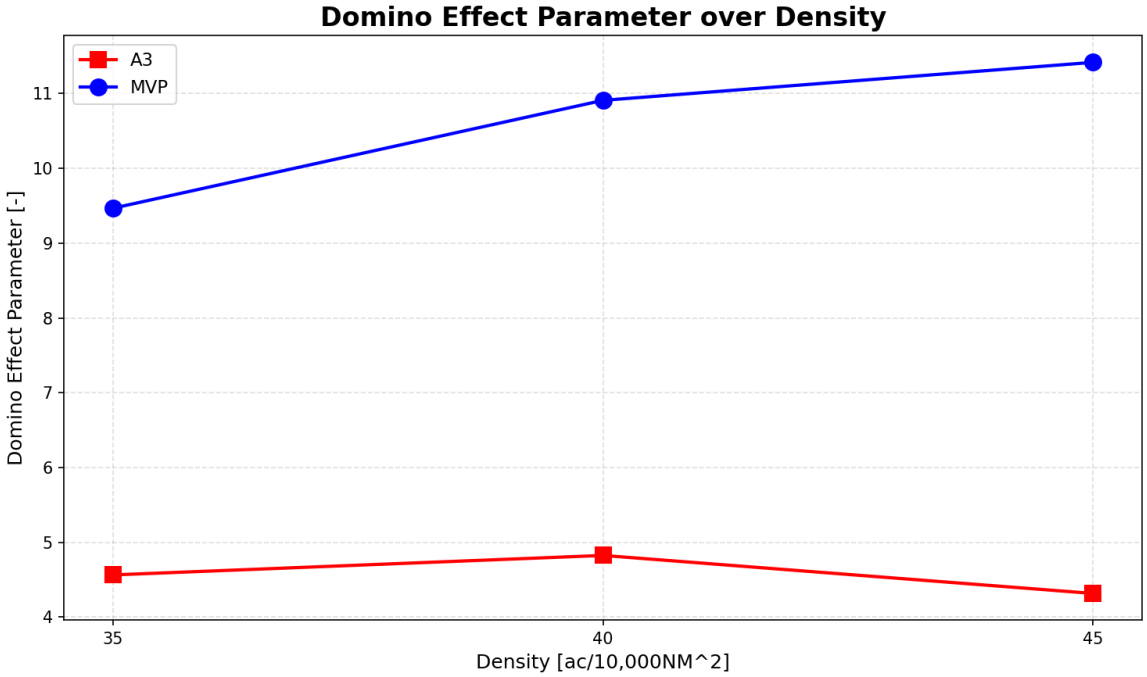



Figure 15: Domino effect parameter.

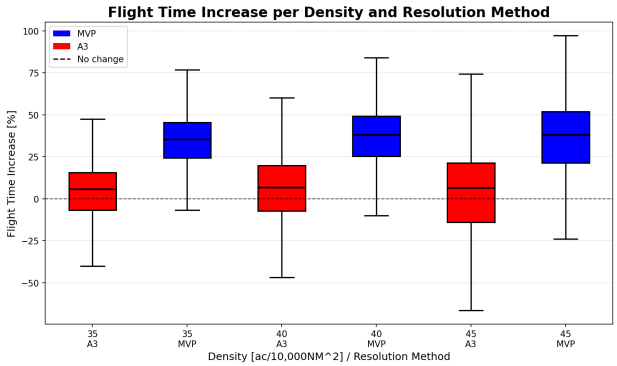
### 4.3.3 Efficiency

The efficiency analysis examines the operational costs of conflict resolution through flight time, distance, and work increases. Figure 16a shows that the MVP method incurs substantial flight time penalties across all density scenarios. In stark contrast, the A3 method maintains minimal flight time increase, indicating more time efficient conflict resolution. This paints a similar picture as the density results shown in section 4.3.1. As the MVP method increases the flight time significantly by slowing down, the density inside the measurement area increases. The A3 model finds conflict resolutions without a speed change, making more time efficient resolutions.

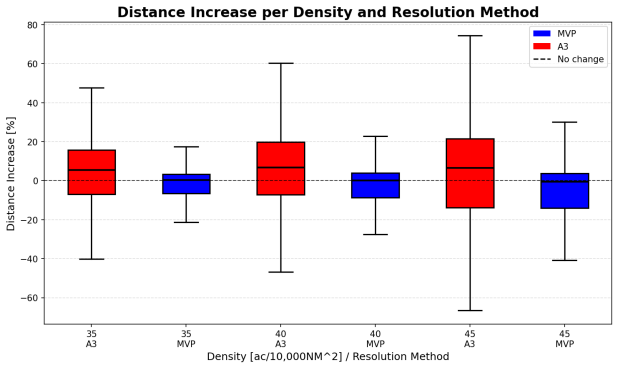
Distance increase per flight are shown in Figure 16b. This figure shows again shows similar results on the difference in behaviour of the A3 model and the MVP method. As the MVP method relies on speed change next to heading change, and flight time results showed a significant increase, it is to be expected that the covered distance does not increase as significantly. As the A3 model does not consider speed change, it is to be expected that the increase in distance is higher than for the MVP method.

Work increase analysis shown in Figure 17 shows that even though the flight time and distance increase results are opposite, the flight effort increases are similar. The A3 model does show much larger ranges, similar to the flight time and distance increase results. This variance might be due to the larger look-ahead time. Similar to the eight-aircraft scenario presented in section 4.2, aircraft with a lower priority tend to be forced to fly around high density area, increasing their flight time, distance, and flight effort significantly.

It is noted that in Figure 16a, Figure 16b, and in Figure 17 the boxplots also extend into the negative, showing a decrease in flight time, distance, or flight effort. This is due to the experiment design, where aircraft due to conflict resolution are forced out of the measurement area early. In the results, the time, distance, and flight effort of an aircraft in between 2 entries of the measurement area are taken into account. However, if an aircraft leaves early without re-entering the measurement area, this shows as a flight time, distance, or work decrease.




(a) Flight time increase.




(b) Distance increase.

Figure 16: Flight metrics increase comparison.

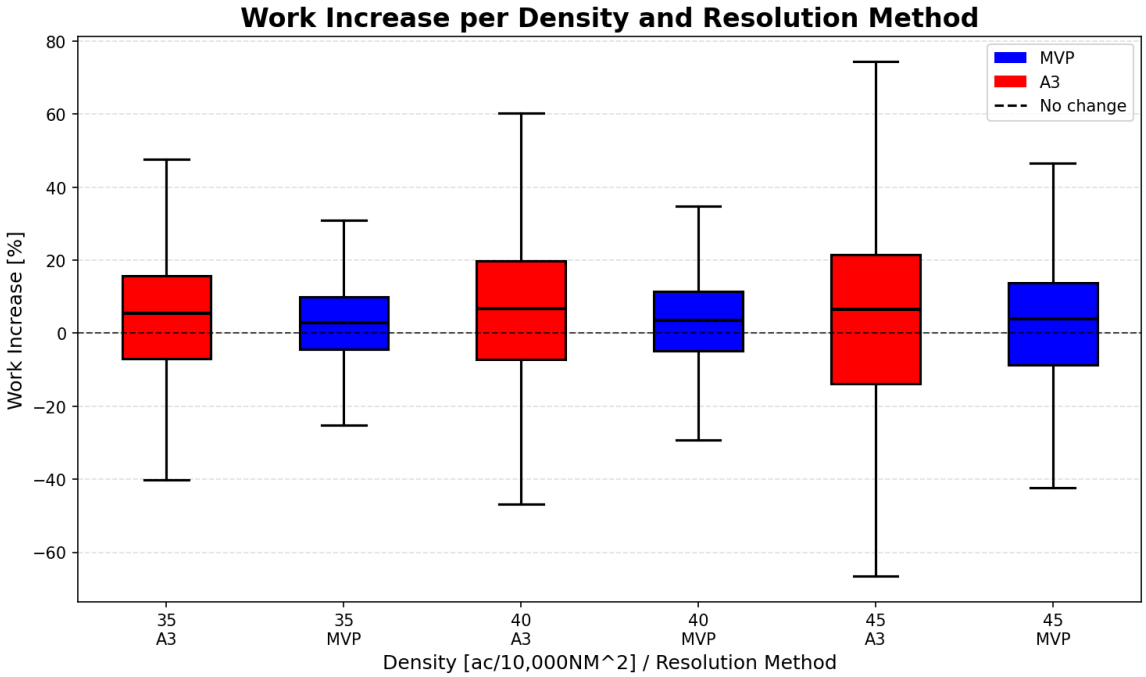



Figure 17: Work increase.

# 5 Discussion

In this discussion, based on the results presented in section 4 a comparison is made between the A3 model and the MVP method in section 5.1. Then, the impact of intent information in conflict resolution is discussed in section 5.2. Next, the limitations of this research are elaborated upon in section 5.3. Finally, recommendations for future research are given in section 5.4.

## 5.1 Performance Comparison

The random traffic simulation results show that the A3 model achieves substantially fewer conflicts and losses of separation than the MVP method across all tested traffic density levels. This is consistent with the longer look-ahead time of the A3 model and its intent-based conflict resolution strategy. By planning conflict-free trajectories up to 15 minutes ahead using MTCR, the model is able to resolve potential future conflicts in advance. The MVP method gets into conflicts more regularly by only reacting to the detected conflict, not taking into account its future trajectory. The domino effect parameter further illustrates this, with the MVP method generating substantially more conflicts than the A3 model.

However, when intrusions do occur using the A3 model, they tend to be larger. The intrusion distribution results show that the A3 model produces outliers with significantly larger horizontal intrusions into the PAZ, including violations below 2/3 separation, which are not observed at all with the MVP method. The MVP method exhibits a much narrower intrusion distribution. This suggests that while the A3 model is effective at preventing conflicts in the longer term, it performs worse when a loss of separation is unavoidable.

Regarding efficiency, the A3 model and MVP method show opposite trends in flight time and distance increase. The MVP method incurs substantial flight time increases across all density scenarios due to its reliance on speed reduction to resolve conflicts. The A3 model, which does not use speed changes, avoids this. However, the resolutions of the A3 model require aircraft to make a larger detour, resulting in greater distance increases than MVP. The variance in flight time and distance increases under the A3 model is also considerably larger. This is consistent with the eight-aircraft scenario results presented in section 4.2, where lower-priority aircraft are forced to fly larger detours around higher-priority aircraft. This asymmetric distribution of resolution costs is an inherent consequence of the priority-driven design of the A3 model. However, this effect may also spread aircraft over a larger area, preventing aircraft flying into high density areas, preventing more conflicts.

## 5.2 Impact of Intent Information in Conflict Resolution

The results of this study suggest that intent information provides meaningful advantages for CD&R performance. The primary benefit of the A3 model over the MVP method is a substantial reduction in the number of conflicts and losses of separation. By using the planned trajectories of surrounding aircraft, the A3 model is able to resolve conflicts earlier and compute resolutions that remain conflict-free for an extended horizon, which also reduces the domino effect.

However, the larger intrusions are concerning. According to Blom and Bakker (2013) the probability of a loss of 2/3 separation in controlled United Kingdom airspace is $10^{-4}$ per aircraft [18]. Knowing that the amount of aircraft in the measurement area per flight hour is approximately 1000 for the highest traffic density scenario, it can be estimated that this event is expected to happen 0.1 times per flight hour. However, the recorded frequency is more than 3. This was not seen in the safety evaluation by Blom and Bakker (2015), so it is expected this is due to differences with the numerical implementation of the model there and in this paper [12]. An overview of these differences can be found in Appendix A.

The efficiency results show that the intent-based approach reduces overall flight time increases by not using speed changes, but introduces larger and more variable distance increases. As the aim is to reduce air traffic flow management delays, this is a major benefit of the A3 model. Additionally, the distance increase might be worth it, as aircraft increase flight distance to avoid high density traffic areas, possibly preventing more losses of separation.

## 5.3 Limitations

Several limitations of this study should be noted. First, the different taxonomy of the two conflict resolution models. Where the intent-based A3 model finds a joint resolution using either a trajectory or heading resolution, the state-based MVP method resolves pairwise summed with both heading and speed alteration.

Secondly, the A3 model is based on a concept of operations, aiming to be realistic on real-world operations and limitations. The MVP method on the other hand is more idealized. This shows in the conflict detection, where the A3 model has update intervals in which conflict detection is performed, where state-based detection is performed every timestep. Also, the A3 model involves a pilot decision time, which is not considered in the MVP method. Finally, the A3 model limits the magnitude of the resolutions by limiting the heading change per TCP and the total amount of detour accepted. The MVP method on the other hand does not limit its resolutions.

third, only two conflict resolution methods are compared: the intent-based A3 model and the state-based MVP method. Without a broader comparative evaluation, using different intent-based and state-based conflict resolution models, it is unclear whether the observed performance differences are specific to the A3 versus MVP comparison or reflect more general properties of intent-based versus state-based conflict resolution.

Fourth, no uncertainty quantification is performed. Each scenario is run only once for each resolution method, meaning the variability inherent in the simulation, arising for example from the stochastic pilot decision times in the A3 model, is not characterized. It is therefore unclear whether the observed differences between A3 and MVP reflect consistent performance differences or are attributable to random variation. This is particularly interesting concerning the larger intrusion values found using the A3 model. Monte Carlo simulations with multiple replications would be required to establish statistical confidence in the results, as was done in the safety evaluation by Blom and Bakker (2015) [12].

Fifth, the scenarios considered are limited in scope. The verification scenarios are stylized conflict geometries that do not capture the full diversity of conflict types encountered in realistic traffic. The random traffic simulations are conducted at a single flight level and without wind or performance uncertainties, which is also a simplification compared to the simulations in [12].

## 5.4 Future Work

Based on the limitations identified above, several directions for future research are suggested. First, when performing a similar comparison study, setting equal degrees of freedom and limitations on both conflict resolution methods will lead to a better comparison. For example, a pilot decision time or maximum resolution heading might be included, or the A3 model might be adapted to include speed alterations.

Second, the comparison should be extended to a wider range of conflict resolution methods. This would provide a more complete understanding of how intent information impacts conflict resolution and when the advantages are most pronounced.

Second, proper uncertainty quantification should be introduced. Repeating each scenario with multiple random seeds and characterizing the distribution of performance outcomes would

substantially strengthen the conclusions. This is particularly important given the stochastic pilot decision timing in the A3 model, and the lack thereof in the MVP method.

Third, future work should focus on how scenarios should be constructed for the evaluation of CD&R methods. The choice of scenario design, including traffic density, airspace geometry, and the distribution of conflict types, can have an effect on the performance of a conflict resolution method. The density anomalies observed in section 4.3.1 illustrate how aspects of experiment design might influence results.

# 6 Conclusion

This research implemented and evaluated the A3 CD&R model in the BlueSky ATM simulator, comparing its performance against the MVP resolution method across multiple traffic density scenarios. The study provides insights into the effect of intent information on conflict resolution performance.

The A3 CD&R model compared to the MVP method demonstrates promising performance characteristics for autonomous conflict detection and resolution. Both in the frequency of conflict and loss of separation the intent-based A3 model performs better than the state-based MVP method. The intent-based resolutions allow for much less cascading conflicts. Additionally, the intent-based resolutions ensure significantly lower flight time increase, as well as avoid high density areas. However, the large intrusions when using the A3 model are concerning. Overall, intent information appears to be a valuable input for conflict resolution models.

Several limitations remain. There are more differences between the two methods than only using intent information or not. Also, limitations in the A3 model are not imposed on the MVP resolution method. Furthermore, only two methods are compared, and the scenarios are limited in scope: single flight level, no wind or performance uncertainties, and deterministic traffic generation without Monte Carlo replication.

Future work should address these limitations by extending the comparison to additional conflict resolution methods, imposing similar resolution limitations within the methods, introducing proper uncertainty quantification through Monte Carlo simulation, and investigating the scenario design choices that most fairly expose the advantages and disadvantages of intent-based conflict resolution.

This study contributes to the broader CD&R research by analysing the impact of intent information on the performance of conflict resolution. This is a step in the research on distributed conflict resolution, required to implement the concept of Free Flight in real-world operations to solve air traffic flow management delays and allow optimal flight routes.

# References


[1] *Deep Change*. IATA, June 2024, pp. 1–24. URL: https://www.iata.org/en/iata-repository/publications/economic-reports/global-outlook-for-air-transport-june-2024-report/.

[2] Richard McCausland. *Net zero carbon 2050 resolution - Fact Sheet*. Oct. 4, 2021. URL: https://www.iata.org/en/iata-repository/pressroom/fact-sheets/fact-sheet-iata-net-zero-resolution/.

[3] Xavier Fron, Piotr Ikanowicz, José Miguel de Pablo Guerrero, Ismail Hakki Polat, Michael Steinfurth, Daniel Weder, and Peter Whysall. *Performance Review Report: An Assessment of Air Traffic Management in Europe*. Performance Review Report. Eurocontrol, May 2024, pp. 1–134. URL: https://www.eurocontrol.int/sites/default/files/2024-06/eurocontrol-performance-review-report-2023.pdf (visited on 11/18/2024).

[4] J. M Hoekstra, R. N. H. W van Gent, and R. C. J Ruigrok. "Designing for safety: the free flight air traffic management concept". In: *Reliability Engineering & System Safety* 75.2 (Feb. 1, 2002), pp. 215–232. ISSN: 0951-8320. DOI: 10.1016/S0951-8320(01)00096-5. URL: https://www.sciencedirect.com/science/article/pii/S0951832001000965 (visited on 10/07/2024).

[5] J.K. Kuchar and L.C. Yang. "A review of conflict detection and resolution modeling methods". In: *IEEE Transactions on Intelligent Transportation Systems* 1.4 (Dec. 2000). Conference Name: IEEE Transactions on Intelligent Transportation Systems, pp. 179–189. ISSN: 1558-0016. DOI: 10.1109/6979.898217. URL: https://ieeexplore.ieee.org/document/898217 (visited on 10/07/2024).

[6] Marta Ribeiro, Joost Ellerbroek, and Jacco Hoekstra. "Review of Conflict Resolution Methods for Manned and Unmanned Aviation". In: *Aerospace* 7.6 (June 2020). Number: 6 Publisher: Multidisciplinary Digital Publishing Institute, pp. 1–37. ISSN: 2226-4310. DOI: 10.3390/aerospace7060079. URL: https://www.mdpi.com/2226-4310/7/6/79 (visited on 10/07/2024).

[7] Lee Yang and James Kuchar. "Using intent information in probabilistic conflict analysis". In: *Guidance, Navigation, and Control Conference and Exhibit*. Guidance, Navigation, and Control and Co-located Conferences. American Institute of Aeronautics and Astronautics, Aug. 10, 1998, pp. 797–806. DOI: 10.2514/6.1998-4237. URL: https://arc.aiaa.org/doi/10.2514/6.1998-4237 (visited on 11/20/2024).

[8] Inseok Hwang and Chze Eng Seah. "Intent-Based Probabilistic Conflict Detection for the Next Generation Air Transportation System". In: *Proceedings of the IEEE* 96.12 (Dec. 2008). Conference Name: Proceedings of the IEEE, pp. 2040–2059. ISSN: 1558-2256. DOI: 10.1109/JPROC.2008.2006138. URL: https://ieeexplore.ieee.org/document/4745649 (visited on 11/20/2024).

[9] Marco Porretta, Wolfgang Schuster, Arnab Majumdar, and Washington Ochieng. "Strategic Conflict Detection and Resolution Using Aircraft Intent Information". In: *The Journal of Navigation* 63.1 (Jan. 2010), pp. 61–88. ISSN: 1469-7785, 0373-4633. DOI: 10.1017/S0373463309990270. URL: https://www.cambridge.org/core/journals/journal-of-navigation/article/strategic-conflict-detection-and-resolution-using-aircraft-intent-information/326E85D05619AC5DA182E7EA07F561C7 (visited on 11/20/2024).

[10] Gustavo Adrián Mercado Velasco, Clark Borst, Joost Ellerbroek, M. M. van Paassen, and Max Mulder. "The Use of Intent Information in Conflict Detection and Resolution Models Based on Dynamic Velocity Obstacles". In: *IEEE Transactions on Intelligent Transportation Systems* 16.4 (Aug. 2015). Conference Name: IEEE Transactions on Intelligent Transportation Systems, pp. 2297–2302. ISSN: 1558-0016. DOI: 10.1109/TITS.2014.2376031. URL: https://ieeexplore.ieee.org/document/7063942 (visited on 10/07/2024).

[11] Gustavo Cuevas, Ignacio Echegoyen, José García García, Petr Cásek, Claudia Keinrath, Rosa Weber, Petr Gotthard, Frank Bussink, and Aavo Luuk. *iFly D1.3: Autonomous Aircraft Advanced (A3) ConOps*. D1.3. Jan. 30, 2010, pp. 1–130. URL: https://ifly.nlr.nl/documents/D1.3_Final_3.0_29_Jan_2010.pdf.

[12] Henk A. P. Blom and G. J. Bakker. "Safety Evaluation of Advanced Self-Separation Under Very High En Route Traffic Demand". In: *Journal of Aerospace Information Systems* 12.6 (2015). Publisher: American Institute of Aeronautics and Astronautics _eprint: https://doi.org/10.2514/1.I010243, pp. 413–427. ISSN: 1940-3151. DOI: 10.2514/1.I010243. URL: https://doi.org/10.2514/1.I010243 (visited on 10/07/2024).

[13] Emmanuel Sunil, Joost Ellerbroek, Jacco Hoekstra, Andrija Vidosavljevic, Michael Arntzen, Frank Bussink, and Dennis Nieuwenhuisen. "Analysis of Airspace Structure and Capacity for Decentralized Separation Using Fast-Time Simulations". In: *Journal of Guidance, Control, and Dynamics* 40.1 (Jan. 2017), pp. 38–51. ISSN: 0731-5090, 1533-3884. DOI: 10.2514/1.G000528. URL: https://arc.aiaa.org/doi/10.2514/1.G000528 (visited on 11/07/2024).

[14] Jacco M Hoekstra and Joost Ellerbroek. "BlueSky ATC Simulator Project: an Open Data and Open Source Approach". In: *International Conference on Research in Air Transportation* 7 (2016), pp. 1–9. URL: https://pure.tudelft.nl/ws/files/10083831/Hoekstra_BlueSky_project.pdf.

[15] G.J. Bakker, C.I.J. Nieskens, and H.A.P. Blom. *Accident Risk Assessment Model for the A3 CONOPS Considered within iFLY*. Technical Report. Version 1.0, NLR Company Confidential. Version accessed: 20 March 2023. National Aerospace Laboratory (NLR), 2010.

[16] G. Bakker. *BB_Source_iFLY_version_A7_CubeIPS_PRM_v4b*. Personal communication. Source code received 8 September 2025. 2025.

[17] Martin S. Eby. "A Self-Organizational Approach for Resolving Air Traffic Conflicts". In: *The Lincoln Laboratory Journal* 7 (1994), pp. 239–254. URL: https://archive.ll.mit.edu/publications/journal/pdf/vol07_no2/7.2.6.selforganizing.pdf.

[18] Henk Blom and G.J. Bakker. "In search of positive emergent behaviour in Trajectory Based Operations". In: *SIDs 2013 - Proceedings of the SESAR Innovation Days* (Jan. 1, 2013).

# Appendices

## A Appendix A

The A3 CD&R model used in this paper is originally implemented in the TOPAZ-iFLY Pascal codebase, which is based on the specification document [15, 16]. During the implementation of the BlueSky plugin described in section 2, several differences were identified between the specification document and the TOPAZ implementation. These differences fall into two categories: functions present in the TOPAZ implementation that are not described in the specification described in section A.1, and parameter values that differ between the specification and the code described in section A.2. Furthermore, some mistakes were made during the implementation of the A3 model in BlueSky, leading to differences. These differences are described in section A.3.

### A.1 Functions Not Described in the Specification

The first difference is that the specification document is incomplete concerning the functions used in the TOPAZ implementation. The missing functions are mentioned below.

#### A.1.1 MTCR Resolution Algorithm

The specification document defines the ASAS MTCR Advisory LPN in appendix U.4 and references the resolution function $f_{EvalResHor}^{MTCR,VO,Q,New}$ by name, but does not describe the algorithm itself. The iterative angular search, the back-to-goal turn logic, and the separation reduction loop are all implementation details present only in the TOPAZ implementation.

### A.1.2 Divergence Mechanism

The TOPAZ implementation implements a divergence mechanism in the STCR algorithm that is not described in the specification document. When the computed STCR resolution heading is nearly parallel to the aircraft's current heading, there is a risk that multiple aircraft involved in the same conflict execute similar heading changes, potentially maintaining parallel trajectories or generating new conflicts. To prevent this, the divergence mechanism is activated when the cosine of the angle between the resolution heading and the current heading exceeds 0.9994, corresponding to an angular difference of approximately 2° or less.

When the divergence mechanism is activated, three candidate heading adjustments are evaluated: the current parallel resolution heading, and the current heading offset by $\pm\Delta\phi_{div}$, where the code default is $\Delta\phi_{div} = 5°$. For each candidate, the minimum separation distance to all other aircraft is predicted one second ahead. The candidate that maximizes this minimum separation is selected as the divergence heading, which is then used as the starting offset for the STCR resolution search. The divergence criterion is only applied to aircraft within a vertical bound of $H_{div} - 100ft$ of the own aircraft, where $H_{div} = 300ft$.

## A.2 Parameter Differences

Several parameter values differ between the specification document and the TOPAZ implementation. The most notable differences are listed below.

### A.2.1 STCR Horizontal Separation Parameter

The specification document implies a horizontal separation distance for STCR of $R_{sep,STCR} = 5NM$, consistent with the PAZ radius. The code uses a default of $R_{sep,STCR} = 3NM$, which is smaller than the PAZ radius, but this is overridden to $R_{sep,STCR} = 5NM$ in the input parameters, bringing it in line with the specification. The simulations in this paper therefore use the specification value.

### A.2.2 MTCR Vertical Separation Parameter

The specification document specifies a vertical separation height for MTCR of $H_{sep,MTCR} = 950ft$. The code default is $H_{sep,MTCR} = 900ft$, and the input parameters set this to $H_{sep,MTCR} = 1000ft$. The simulations in this paper use the input parameter value of $1000ft$, which is in line with the input parameters.

### A.2.3 Pilot Decision Time for STCR

The specification document specifies a pilot decision time for STCR approximated by a Rayleigh distribution with $\mu_{TD1} = 5.7s$. The code default matches this value. However, the input parameters override this to $\mu_{TD1} = 18s$. This is a deliberate scenario choice that increases the pilot response delay for STCR. The simulations in this paper use $\mu_{TD1} = 5.7s$, in line with the specification document.

### A.2.4 Intent Update Interval

The specification document specifies an intent-based conflict detection update interval of $T_{I,update} = 90s$, i.e., the interval at which the ASAS CD&Management LPN re-evaluates conflicts using intent trajectories on the medium term horizon. The code default is $T_{I,update} = 65s$, which is below this value, and no override is applied in the input parameters. The simulations in this paper use $90s$, in line with the specification document.

## A.3 Differences Between the BlueSky Implementation and the Codebase

Several differences exist between the BlueSky plugin implementation used in this paper and the TOPAZ implementationbase. The most notable differences are described below.

### A.3.1 Divergence Mechanism Applied to Both STCR and MTCR

As described in section A.1, the divergence mechanism in the TOPAZ implementation is only applied within the STCR algorithm. In the BlueSky implementation, the divergence mechanism is also applied to MTCR resolutions. When an MTCR resolution heading is found to be nearly parallel to the current heading (i.e., the cosine of the angle between the two headings exceeds 0.9994), the same divergence routine is invoked to select an initial heading offset that maximizes the minimum separation to nearby aircraft. This extension was motivated by the same concern as the original STCR divergence mechanism: without it, MTCR resolutions can result in aircraft flying parallel courses that do not sufficiently separate.

### A.3.2 10 Second STCR Delay

In the TOPAZ implementation, STCR is triggered when a state-based short-term conflict is detected, i.e., when two aircraft are predicted to violate separation based on their current state vectors rather than their intent trajectories. The specification document defines $T_{STC,res} = 10s$ as the duration of a state-based conflict that must elapse before the system switches to STCR resolution mode. This delay is therefore tied to the state-based detection mechanism.

In the BlueSky implementation, a fixed 10-second delay is added to all STCR resolutions in addition to the Rayleigh-distributed pilot decision time, regardless of how the conflict was detected. Since the BlueSky implementation uses only intent-based conflict detection and state-based detection is not considered in this paper, this 10-second delay is not appropriate. It is applied unconditionally to all STCR resolutions, meaning that STCR resolutions in the simulations are systematically delayed by 10 seconds beyond what the specification and codebase intend for intent-detected short-term conflicts. Combined with the pilot decision time of $\mu_{TD1} = 5.7s$, the effective STCR response time in the BlueSky implementation is approximately $15.7s$, compared to the specification value of $\mu_{TD1} = 5.7s$.

# B Appendix B

The random traffic scenarios used in the simulations are generated using a BlueSky plugin. Before any simulation is run, the plugin generates a scenario file for each traffic density configuration. This appendix describes how these scenario files are created. The scenario generation consists of three steps: defining the spawn and destination positions, generating origin-destination (OD) pairs, and writing the scenario file.

## B.1 Spawn and Destination Positions

Aircraft are spawned on the edge of the experiment area, which has a side length of $L_{exp} = 500NM$. To prevent conflicts due to simultaneous spawning at the same location, a set of discrete spawn positions is pre-calculated along the four edges of the experiment area. The number of spawn positions is determined by dividing the total perimeter of the experiment area by an effective separation distance, given by:

$$d_{eff} = R_{\text{PAZ}} \cdot (1 + \sigma), \tag{20}$$

where $R_{\text{PAZ}} = 5NM$ is the PAZ radius and $\sigma = 4$ is a separation margin. This results in an effective separation of $d_{eff} = 25NM$ between adjacent spawn positions. The positions are distributed evenly across the four edges of the experiment area, with an equal number of positions on each edge. A spawn position is marked as unavailable for a cooldown period after an aircraft is spawned there. This cooldown is the time required for the aircraft to travel at least one PAZ radius away from the spawn point, plus a 60-second buffer.

Destination positions are calculated in the same way, but on the edge of a larger destination area with a side length of $L_{dest} = 707NM$. Placing destinations outside the experiment area ensures that aircraft do not converge on the same destination without being able to resolve the conflict.

## B.2 Origin-Destination Pair Generation

OD pairs are generated sequentially, starting from $t = 0$. For each aircraft, a true airspeed is drawn uniformly from $v \in [450, 500]kts$, and a spawn and destination position are selected at random from the pre-calculated edge positions. The time the aircraft will spend inside the measurement area is then computed by finding the intersection of the straight-line flight path between origin and destination with the boundary of the measurement area, which has a side length of $L_{meas} = 316.7NM$. If the resulting time inside the measurement area is less than the minimum required flight time of 30 min, the OD pair is rejected and a new one is generated.

Once a valid OD pair is accepted, the spawn time of the next aircraft is determined based on the predicted number of aircraft inside the measurement area at the time the next aircraft is expected to enter it. This expected count is computed from the entry and exit events of all previously generated aircraft. If the predicted count is below the target density, the next spawn time is set to $\Delta t_{spawn} \cdot \alpha$ after the current spawn time, where:

$$\Delta t_{spawn} = \frac{\bar{t}_{in}}{N_{target}}, \tag{21}$$

with $\bar{t}_{in}$ the mean time in the measurement area of all aircraft generated so far, $N_{target}$ the target number of aircraft in the measurement area, and $\alpha = 0.75$ a factor to accelerate spawning when below the target density. If the predicted count is already at or above the target density, the next spawn time is set such that the new aircraft enters the measurement area when a sufficient number of existing aircraft are expected to have exited, preventing the density from overshooting the target level.

This process is repeated until the end of the logging phase at $t = 9000s$. The resulting OD pairs, including spawn time, entry and exit times, airspeed, time and distance in the measurement area, and origin and destination coordinates, are saved to a log file. This log file is reused for all simulation configurations that share the same traffic density, ensuring that the traffic patterns are identical across different CD&R method configurations.

## B.3 Scenario File Creation

Using the generated OD pairs, a BlueSky scenario file is created for each simulation configuration. The scenario file first defines the simulation setup, including the center coordinates, the boundaries of the measurement area and experiment area, and the conflict detection and resolution methods to be used. The measurement and experiment areas are defined as rectangular regions using the `AREA` and `EXP` commands, respectively.

Subsequently, a `CRE` command is written for each aircraft in the OD pair log, scheduled at the aircraft's spawn time. This command creates the aircraft at its origin position with the correct heading toward its destination, at a cruise altitude of $H = 36{,}000$ ft and the corresponding

calibrated airspeed. An `ADDWPT` command is then written to assign the destination waypoint, followed by an `LNAV ON` command to activate lateral navigation. The aircraft will then fly a straight-line path from its spawn position to its destination at constant altitude and speed.

# II

# Literature Study

# 1 Introduction

During the past 20 years, the aviation sector has grown almost continuously. IATA expects that passenger numbers will hit 10 billion by 2050 [1, 2]. However, according to Eurocontrol, air traffic delays reached the second highest value in 20 years [3]. A significant part of this is attributed to the air traffic control (ATC) capacity.

The concept of Free Flight is one of the most promising developments for increasing the capacity in air traffic management (ATM). This concept allows optimisation of flight routes by airlines. The pilot is responsible for maintaining airborne separation. Aircraft ensuring self separation is thought to be more effective than a global optimisation of flight routes along airways by a centralised ATC [4]. To allow for safe operations within such a Self Separating Airspace (SSA), aircraft themselves have to be able to detect and resolve conflicts [5]. Airborne self separation research focuses on the development of Airborne Separation Assistance Systems (ASAS). It has seen a tremendous development worldwide since the invention of Free Flight in 1995. Examples are research programs like DAG/TM or SESAR [6, 7].

An example of a project studying the feasibility of SSA in high-density traffic conditions is the iFly project[1]. Through a sequence of studies within iFly an advanced airborne self separation concept of operations has been developed. This is called the Autonomous Aircraft Advanced (A3) Concept of Operations (ConOps) [5]. For this ConOps a model including Conflict Detection & Resolution (CD&R) using both state and intent information has been developed. This will be called the A3 CD&R model in this report. The state information of an aircraft contain its current position and velocity. The intent information includes future states of the aircraft such as position and velocity. Previous analysis of the A3 CD&R model have shown positive safety and flight efficiency related characteristics [8]. These findings encourage further analysis of the A3 CD&R model.

In this report, first a literature study is performed in chapter 2. Then a research gap is identified and research questions presented in chapter 3. Based on these gap and questions a preliminary methodology is given in chapter 4. Finally, the report is concluded in chapter 5.

[1] https://ifly.nlr.nl/

# 2

# Literature Study

This section provides an overview of the perfomed literature study. The literature study provides the relevant background for the identification of the research gap and research questions. This includes a definition of multiple CD&R models and how they can be categorised in Section 2.1. Following is a description of the A3 Model in Section 2.2. Next, the requirements for an ATM simulator and the chosen Open Source ATM simulator BlueSky are elaborated upon in Section 2.4. Finally an overview will be given of how CD&R models are simulated in Section 2.5.

## 2.1. CD&R Models

Before diving into the A3 model it is good to start with the development of CD&R models. To begin, it is necessary to have a clear definition of a conflict. A conflict is an event in which two or more aircraft experience a loss of separation (LoS) [9]. Separation minima are defined to be the protected aircraft zone (PAZ), which is, in en-route airspace, defined as a disc with diameter of 10 NM and a height of 2000 ft providing a horizontal separation of 5 NM and a vertical separation of 1000 ft, as is shown in Figure 2.1 [10]. The PAZ could also be defined as a much smaller region (e.g. a sphere of 500 ft in diameter) in the case of tactical collision alerting systems or in terms of parameters other than distance, for example time [9].

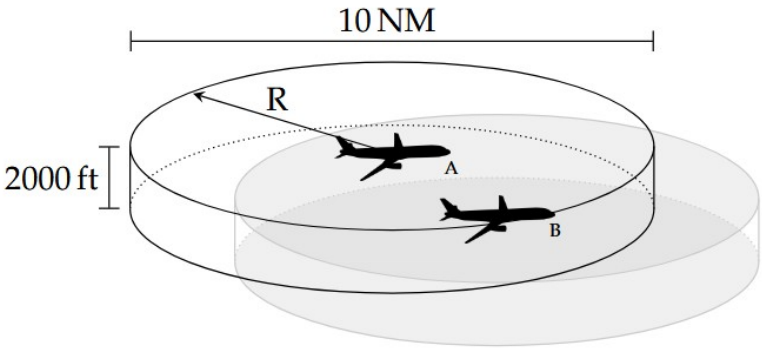



Figure 2.1: The International Civil Aviation Organization's (ICAO) separation minima: 5 NM horizontal separation and 1000 ft vertical separation [10].

The goal of the CD&R system is to predict that a conflict is going to occur in the future, communicate the detected conflict to a human operator, and assist in the resolution of the conflict situation [9]. To provide insight into different methods of conflict detection and resolution, methods can be evaluated according to ten characteristics [10]:

1. The type of surveillance
2. Trajectory propagation
3. Predictability assumption
4. Centralised or distributed control
5. Method category
6. Multi-actor conflict resolution
7. Timescale of avoidance planning
8. Type of resolution manoeuvre
9. Type of obstacle
10. Optimisation objective

In Section 2.1.1 an overview is given of some CD&R models and their categorisation. Next, the first three items are characteristics used in the conflict detection phase and will be elaborated upon in Section 2.1.2. The last seven characteristics are on the conflict resolution phase and will be elaborated upon in Section 2.1.3.

### 2.1.1. Overview

To summarise Sections 2.1.2 and 2.1.3, CD&R methods can be categorised by defining ten characteristics. The ten characteristics and their categories are visualised in Tables 2.1 and 2.2.

Table 2.1: Taxonomoy of conflict detection categories [10].

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

A lot of conflict detection and resolution methods are shown in Table 3 in [10]. Additions to Table 3 in [10] are introduced in Table 2.3. The used abbreviations can be seen in Table 2.4.

Table 2.2: Taxonomoy of conflict resolution categories [10].

| Conflict Resolution Categories | | | Applicable For All Conflict Resolution Categories | | | |
|---|---|---|---|---|---|---|
| Control | Method Categories | Multi-Actor Conflict Resolution | Avoidance Planning | Avoidance Manoeuvre | Obstacle Types | Optimization |
| Centralised ↦ | Exact | Sequential | Strategic | Heading | Static | Flight Path |
| | Heuristic → | Concurrent | Tactical | Speed | Dynamic | Flight Time |
| Distributed ↦ | Prescribed | Pairwise Sequential | Escape | Vertical | All | Fuel/Energy Consumption |
| | Reactive → | Pairwise Summed | | Flight Plan | | |
| | Explicitly Negotiated | Joint Solution | | | | |

Table 2.3: Taxonomy of CD&R methods, additions to Table 3 in [10]

| Met | Sur | Tra | PAs | Con | Mul | Plan | Obs | AvMan | Source |
|---|---|---|---|---|---|---|---|---|---|
| H | C | S | N | C | S | T | D | H+S+V | Sui et al. [11] |
| H | C | I | N | C | S | T | D | FP | Sui and Zhang [12] |
| R | C | S | N | D | PSE | T | D | S | Brittain and Wei [13] |
| R | D | S | N | D | J | T | A | H+S | Piccinelli et al. [14] |
| R | D | S | P | D | J | T | A | H+S+V | Naga Jyotish et al. [15] |
| R | D | S | N | D | J | T | D | FP | Bonanni et al. [16] |
| R | I | S | N | D | J | T | A | H | Fan et al. [17] |
| R | D | I | N | D | J | T | D | FP | Blom and Bakker [8] |

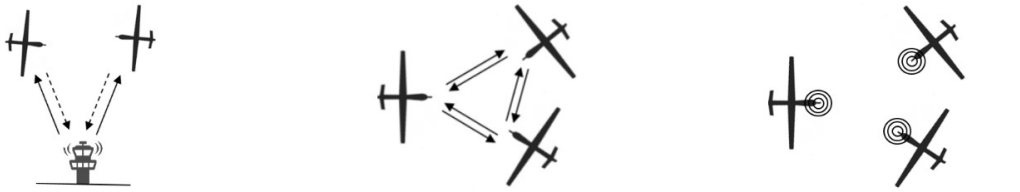
Table 2.4: Abbreviations used for taxonomy of conflict resolution methods [10].

| Category | Abbrev. | Meaning |
|---|---|---|
| Method Categories (Met) | E | Exact |
| | H | Heuristic |
| | P | Prescribed |
| | R | Reactive |
| | EN | Explicitly Negotiated |
| Surveillance (Sur) | C | Centralised Dependent |
| | D | Distributed Dependent |
| | I | Independent |
| Trajectory Propagation (Tra) | S | State-based |
| | I | Intent-based |
| Predictability Assumption (PAs) | N | Nominal |
| | P | Probabilistic |
| | WC | Worst-case |
| Control | C | Centralised |
| | D | Distributed |
| Multi-Actor Conflict Resolution (Mul) | S | Sequential |
| | C | Concurrent |
| | PSE | Pairwise Sequential |
| | PSU | Pairwise Summed |
| | J | Joint Solution |
| Avoidance Planning (Plan) | S | Strategic |
| | T | Tactical |
| | E | Escape |
| Obstacles (Obs) | S | Static |
| | D | Dynamic |
| | A | All |
| Avoidance Manoeuvre (AvMan) | H | Heading |
| | S | Speed |
| | V | Vertical |
| | H+V | Horizontal and Vertical |
| | H/V | Horizontal or Vertical |
| | FP | Flight Plan |

### 2.1.2. Conflict Detection Characteristics

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) [10]. Within the dependent surveillance systems, an additional distinction can be made based on the origin of the data: a centralised system receives data from a common station (e.g. ATC), whereas a distributed system processes information broadcast and received to and from the surrounding traffic (e.g. via ADS-B). Independent surveillance is more commonly referred to as Sense and Avoid and uses on-board sensors to detect static and dynamic obstacles (e.g. [17]) [10]. Independent surveillance is not employed in manned aviation, however, as aircraft are ex-

pected to cooperate through the ADS-B system [10]. In Figure 2.2 the three different types of surveillance are visualised.



(**a**)Centralised dependent surveillance. (**b**)Distributed dependent surveillance. (**c**)Independent surveillance.

Figure 2.2: Different types of surveillance used in CD&R models [10].

The propagation of an aircraft's trajectory can be considered based on their current state (i.e. state-based) or their future intent (i.e, intent-based) [10]. State-based trajectory propagation assumes a straight line as continuation from the current state, whereas intent-based trajectory propagation assumes turns and changes in heading and speed based on the future waypoints of the aircraft [10]. Using a state-based model is naturally simpler and faster computationally, as intent-based models require data transmission and heavier computational processing [10]. False alarms in state-based systems can occur, or potential conflicts overlooked, if future trajectory changes of all aircraft are not taken into account [10]. However, false alarms are also introduced when aircraft diverge from their initially intended trajectory due to conflict resolutions when using intent-based systems [10]. Some models using state-based conflict detection can for example be found in [18, 19, 20]. Intent in the form of future waypoints can be found in [21, 22, 23, 24, 12]

To find a conflict between two aircraft, it is to be identified whether those two aircraft will be closer than the minimum required separation at a future point in time [10]. This thus requires an estimation of the future positions of all aircraft, and it differs on whether uncertainties are added to the trajectory propagation [10]. A nominal assumption (Figure 2.3a) does not consider uncertainties (i.e. uncoordinated behaviour from other traffic, unknown wind, or state variation), which is often used for its simplicity and good computational performance [10]. This assumption is mostly used with shorter look ahead times (i.e. a few minutes) and can be quite accurate in an environment where aicraft have a steady behaviour [10]. A worst-case assumption (Figure 2.3b) considers all possible trajectory changes resulting from uncertainties [10]. However, this is impractical in a real environment as its complexity results in a heavy computational load [10]. The middle ground is the probabilistic assumption (Figure 2.3c) [10]. In this case the likelihood of each possible trajectory change is taken into account based on the current position and maximum turn and climb rates [10]. Incorporating uncertainties may improve accuracy; as more potential trajectories are considered, the more likely it is that one will resemble the real observed position in the future [10]. However, this is at the cost of more false alarms and reduces the available manoeuvring space to resolve conflicts [10].

### 2.1.3. Conflict Resolution Characteristics

Separation management, or control, can either be centralised or distributed [10]. Centralised control means that decisions regarding future trajectories and conflict resolutions are computed in a centralised location for multiple aircraft in a certain area for which the

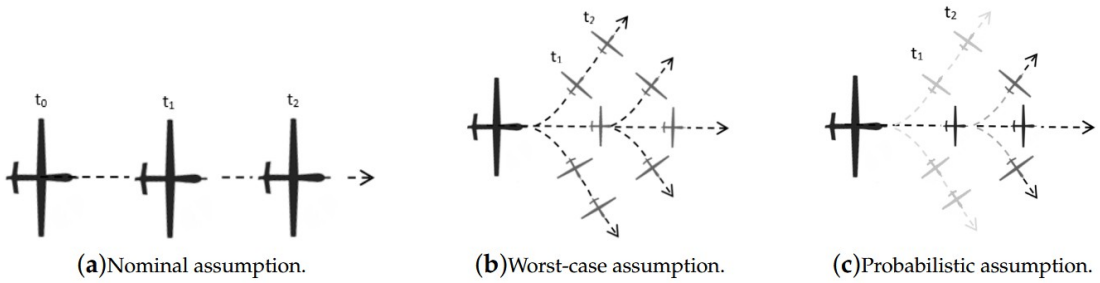



Figure 2.3: Different types of predictability assumption methods [10].

controller is responsible [10]. A centralised system is capable of providing a global solution to complex multiple-actor systems, at the cost of being computationally heavy and having a single point of failure [10]. Distributed control means that individual aircraft are responsible for assuring separation from other aircraft [5]. Each aircraft only needs to solve a fraction of the global problem solved by the centralised system, as each aircraft only considers neighbouring aircraft, incurring computational benefits [5]. A comparison between distributed and centralised systems concluded that, at the loss of some performance, distributed systems can successfully solve complex multiple-aircraft problems in real time with very low computational effort compared to a centralised system [25]. Furthermore, the Free Flight concept illustrates that in a distributed system aircraft are free to decide upon their optimal route, versus following the route received from a centralised system [4, 20].

Next, five conflict resolution method categories are described [10]. For centralised systems an exact and a heuristic approach can be defined. As mentioned above, using a centralised system can find a global solution for a complex multiple-actor systems. This global solution of all aircraft trajectories or avoidance manoeuvres can be optimised for either performance (e.g. fuel consumption, flight path, or flight time) or safety. When using an exact method (e.g. a multi integer linear programming approach [26]) a global optimum is to be found. Otherwise a heuristic method (e.g. Variable Neighborhood search [27], Ant Colony optimization [28], Evolutionary Algorithms [29], or Monte Carlo Tree Search [12]) attempts to yield a good, not necessarily global optimum, solution at the benefit of lower computational time. For distributed systems the prescribed, reactive, and explicitly negotiated categories are defined. Both prescribed and reactive resolution methods are implicitly coordinated, meaning no explicit communication takes place between the aircraft resolving a conflict. Prescribed resolution methods use a set of rules or determined manoeuvres to resolve conflicts (e.g. Right-of-Way rules [30] or fixed climbing-turn manoeuvres [31]). Reactive resolution methods perform manoeuvres based on the conflict geometry, where each aircraft not only reacts to the conflict but can also react to the other aircraft's reaction to the conflict (e.g. eby's self-organisational approach [18]). Explicit coordination resolves conflicts based on explicit communication between aircraft. This can be done by setting a negotiation mechanism where aircraft communicate towards an agreement and/or prioritisation (e.g. safe separation constraints [32], cost negotiated [33], or priority-based [11]).

Centralised and distributed systems have different approaches to multi-actor conflicts [10]. Centralised systems will either use a sequential algorithms according to the prioritisation of the aircraft (e.g. [34, 11]) or concurrent resolution where all trajectories are com-

puted simultaneously (e.g. [35]). Sequential algorithms do need an adequate priority determination method (e.g. [36]). Distributed systems use either a joint solution, a pairwise sequential solution, or a pairwise summed solution. A joint solution resolves multiple conflicts simultaneously in a single solution (e.g. [14, 15, 16]). To avoid too complex situations the look ahead time is limited. Pairwise sequential methods focus on individual conflict pairs and computing a resolution manoeuvre per conflict (e.g. [13]). Pairwise summed (used in [20]) sums all resolution manoeuvres from the pairwise sequential method to perform a single manoeuvre resolving all conflicts. It is good to note that the pairwise sequential method ensures implicit coordination as described above, as implicit coordination uses the reaction of the other aircraft to the conflict. Using a joint solution or pairwise summed resolution manoeuvre thus also had consequences on the implicit coordination between aircraft. It is shown that the summing of the avoidance vectors does have a beneficial emergent global effect of distributing the available airspace between different vehicles [20].

The planning of a resolution manoeuvre can be defined as per the look ahead time and the state of the aircraft after the avoidance manoeuvre is performed [10]. Strategic is a long-range action in the order of 20 minutes to hours before the conflict which changes the flight-path significantly, whereas tactical is a mid-range action in the order of 3 to 5 minutes before the conflict that changes a small part of the flight path [10]. An escape is a short-term manoeuvre in the order of seconds before a collision that brings the aircraft to safety with no additional consideration regarding the flight path [10]. As illustrated in Figure 2.4(a), a strategic resolution manoeuvre affects the planned flight considerably as future waypoints are modified to avoid conflict [10]. A tactical resolution manoeuvre (Figure 2.4(b)) avoids the conflict but does not consider a new flight plan or recovery to the original flight plan [10]. Some sort of coordination can still be utilised to perform the conflict resolution [10]. An escape manoeuvre as shown in Figure 2.4(c) is a last resort where no flight plan or coordination is used due to the lack of time, this is for collision avoidance [10]. In manned aviation, escape manoeuvres are not usually employed [10]. Due to the large PAZ around aircraft, a LoS does not necessarily represent a collision [10].

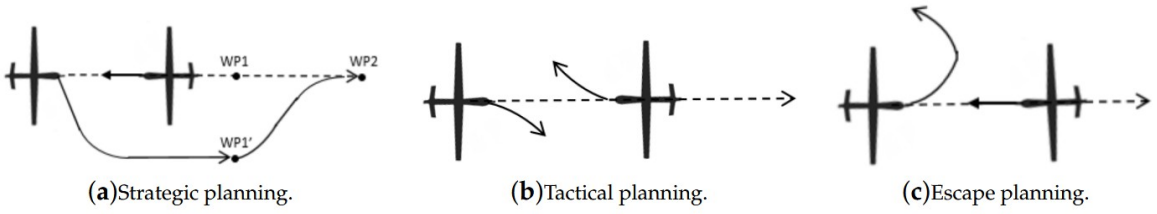



Figure 2.4: Different types of avoidance planning [10].

There are also multiple types of avoidance manoeuvres [10]. These can be based on changing the aircraft's current state (e.g. heading, speed, or altitude) or changing the aircraft's future intent by changing its flight plan [10]. The different state-based avoidance manoeuvres are visualised in Figure 2.5. Most methods use horizontal (thus heading and speed) manoeuvres, some also consider vertical manoeuvres. The added degree of freedom added by vertical manoeuvres results in a greater variety of possible conflict resolution manoeuvres, but also results in more complex optimal route calculation [10]. When at cruise altitude, the manoeuvring space for speed manoeuvres is small as the stall speed at such altitudes is high [10]. However, research done on using so-called 'subliminal' speed control,

i.e. modifying the aircraft speed within a small range around their original speeds, can be used to resolve some conflicts. Its efficiency does depend on the minimal separation and the available time to LoS, and is only possible with non-(near-)head-on conflicts [37, 38]. Other methods have shown that combining heading and speed manoeuvres have potential results [20, 39]. Intent-based resolution manoeuvres change the waypoints the aircraft is intended to follow. This way of avoiding conflicts has gained new attention with the development of the four-dimensional trajectory-based operation (4DTBO) concept, waypoints in three dimensional space with specific timestamps [40]. New flight plans can then be constructed in case of a conflict, either by changing the waypoint or the timestamp at the waypoint, or both. An example of intent-based avoidance can be found in [24, 12]. Lastly, resolution manoeuvres are either discrete, the environment is assumed to remain constant until the manoeuvre is terminated, or continuous, the environment is observed continuously and can adapt the manoeuvre while it is being executed [10]. In practice, it can be a combination where a manoeuvre is computed using an assumed discrete state but can be updated when for example a new conflict changes the existing traffic situation and thus the discrete state [10].

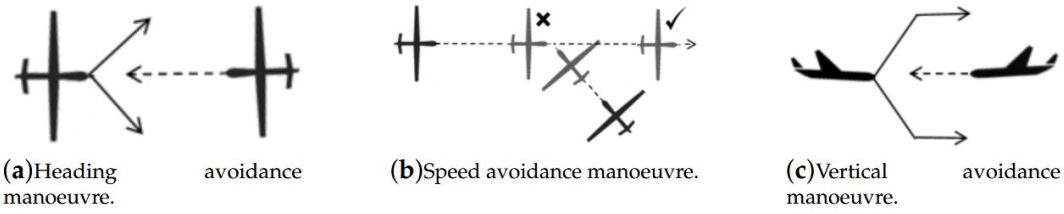



Figure 2.5: Different types of state-based avoidance manoeuvres [10].

A CD&R method may prevent collision with only static obstacles, dynamic obstacles, or with a combination of both [10]. Manned aviation CD&R models will naturally be directed at dynamic obstacles as these models are mostly used when aircraft are flying at cruise altitude [10]. An exception is the Ground Proximity Warning System (GPWS) which detects the aircraft's proximity to the ground as a static obstacle and warns the crew in case of a conflict [41].

Finally, CD&R methods can be categorised by the chosen optimisation objective [10]. For CD&R methods, safety is paramount; however, there is a preference for methods which do not significantly alter the initially planned trajectory or heavily increase the costs of an operation [10]. The efficiency of CD&R methods can be evaluated regarding its effect on the time and/or path of the flight or even fuel/energy consumption [10]. It is to be noted that a CD&R 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. In the Free Flight concept, the choice of optimisation is up to the operator of the aircraft [4].

## 2.2. iFly

As described in the introduction, the iFly project has lead to the development of the A3 Model, which is derived from the A3 ConOps. The A3 ConOps describes the overall concept of operations within the autonomous en-route ATM environment as well as a high

level specification for the required equipment [5]. The A3 ConOps is further elaborated upon in Section 2.2.1. Using the A3 ConOps, the iFly Work Package WP9 descibes an Operational Services and Environment Description (OSED) and the airborne system design requirements of airborne self-separation procedure [42, 43]. The OSED is elaborated upon in Section 2.2.2 and finally the system design requirements are reviewed in Section 2.2.3.

### 2.2.1. A3 ConOps

The basis of the A3 ConOps is that all aircraft are responsible for airborne self separation without ground support from ATC while meeting traffic flow constraints [5]. The A3 ConOps is limited to en-route flight only, defined to be from the departing Terminal Maneuvring Area (TMA) exit point to the entry point of the arrival TMA [5]. This en-route airspace is classified as SSA in which the route from exit point to entry point is defined to be the Reference Business Trajectory (RBT) [5]. The above mentioned traffic flow constraint is the predetermined Controlled Time of Arrival (CTA) at the entry point of the arrival TMA [5]. A CTA is not an exact time of arrival but rather a time window which decreases in size as the aircraft comes closer to the entry point of the arrival TMA [5]. To fly the RBT it is assumed all aircraft are equipped and certified for self separation and precision navigation is the standard [5]. To allow for self separation, the aircraft are equipped with airborne separation assistance systems (ASAS).

The A3 ConOps thus describes some ground rules all autonomous aircraft have to adhere to when inside the SSA and performing self separation. These autonomous flight rules (AFR) are:

- Autonomous aircraft are responsible for maintaining separation with all other aircraft [5].
- Autonomous aircraft are required to maintain separation from designated areas and no-fly zones [5].
- Autonomous aircraft are required to adhere to flow management constraints. Renegotiation will have to take place if these constraints cannot be met [5].
- Lower priority autonomous aircraft involved in a medium term Intent based conflict ruled by priority are required to manoeuvre to solve it sufficiently in advance, so that the conflict does not continue until the conflict resolution becomes a short term cooperative conflict [5].
- Autonomous aircraft shall not manoeuvre in a way that creates a short term (3 to 5 minutes) conflict [5].
- The trajectory of autonomous aircraft shall at no time place the aircraft in a 2 minutes state vector conflict (blunder protection) [5].
- Autonomous aircraft shall not enter managed airspace (for example a TMA) without the approval of the controlling entity of that airspace [5].

A definition of a conflict is given in Section 2.1. In the A3 ConOps additionally a subset within the PAZ is made; the reduced separation minima of 3 NM instead of 5 NM horizontally and 900 ft instead of 1000 ft vertically. The formal PAZ is then called the comfort

separation zone (CSZ), whereas the reduced separation minima form the minimum separation zone (MSZ). An illustration of this is shown in Figure 2.6.

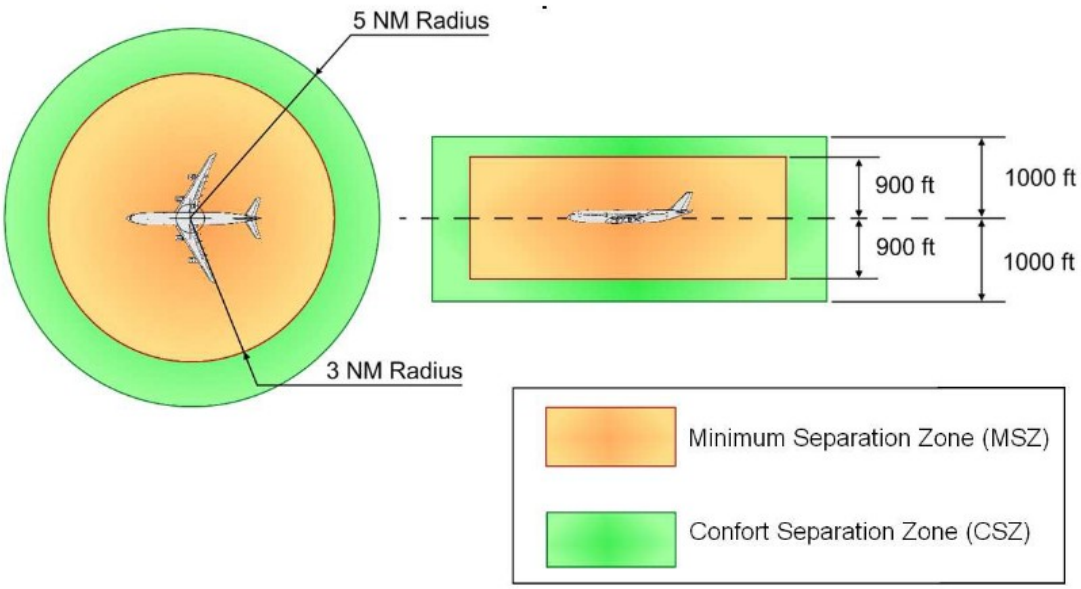



Figure 2.6: Protected Airspace Zone, divided into a comfort separation zone and a minimum separation zone [5].

Next to the AFR, some systems are required to enable the A3 ConOps allowing safe operations within the SSA [5]. These systems include:

- A system wide information management system (SWIM) [5].
- Air-Ground and Air-Air data link communications and surveillance broadcast
- On-board decision support tools (including ASAS) [5].
- Advanced airborne automated applications (e.g. weather data fusion applications, warning functions and guidance algorithms) [5].
- Advanced ground surveillance support (available via SWIM) which allows for:
  - Communicating the presence of other aircraft in the aircraft's awareness zones [5].
  - Detecting complex and/or congested areas [5].
- Advanced human machine interfaces (HMI) [5].
- New procedures and rules (e.g. AFR) [5].
- Flight Management System integrated with the on-board decision support tools [5].
- Airborne Collision Avoidance System (ACAS) [5].

The A3 ConOps uses a combination of long term area conflict detection (LTACD), medium term and short term conflict detection and resolution (MTCD&R, STCD&R), and ACAS to increase the maximum airspace capacity while allowing safe operations [5]. To be able to predict the future position and velocity of surrounding aircraft the A3 ConOps uses not only state information but also intent information. The intent is the planned flight trajectory of the aircraft broadcast up to a medium term time frame and allows surrounding aircraft to predict whether a conflict will occur. In the long term awareness zone (LTAZ) the aircraft, with help of SWIM, knows its own RBT, weather hazards, restricted airspace areas, and other areas to avoid. Within the medium term awareness zone (MTAZ) the aircraft will know its own state and intent and that of the surrounding aircraft [5]. Within the short term awareness zone (STAZ) the aircraft will know its own state and first level of intent as well as that of the surrounding aircraft [5]. The awareness zones are defined to enable processing of relevant information [5].

Table 2.5: Overview of CD&R and ACAS modules considered within A3 ConOps [5].

| | **Look ahead time for CD** | **Coord.** | **Principle of use** | **Priority rules** | **Create secondary conflict?** | **Type of resolution algorithm** |
|---|---|---|---|---|---|---|
| LTACD | >30 min | - | RBT | - | - | - |
| MTCD&R | Up to 15 to 20 min | Not required | Intent | Yes | Do not | Intent based |
| STCD&R | Up to 3 to 5 min | Implicit | State (first level of intent) | No | Do not | 1 on N |
| ACAS | <1 min | Explicit | Pure state | No | Try not | 1 on 1 |

The LTACD uses its own RBT information to detect conflicts with weather hazards or restricted airspaces [5]. If a conflict is detected, trajectory management is activated, however this is not an ASAS function.

The MTCD&R uses intent information only and is priority based [5]. The aircraft with the lowest priority has to make a resolution manoeuvre, whereas the aircraft with the highest priority is allowed to fly its planned trajectory. Priority can be based on multiple factors like the distance to destination, the aircraft manoeuvrability, or its mission statement (commercial, military, governmental, etc.). The resolution manoeuvre performed has to solve all known conflicts and shall not create new conflicts up to a to be determined time [5]. The medium term conflict resolution (MTCR) is in the form of an updated RBT, which will be broadcast to the surrounding aircraft as the new intent of the aircraft.

The STCD&R uses state and first level intent information to resolve conflicts within the STAZ [5]. If first level intent information is not available, the state information of surrounding aircraft is nominally extrapolated, which is considered to be representative. The short term conflict resolution (STCR) will enable a quick execution of conflict resolution due to a focus on conflict resolution instead of trajectory management [5]. The STCR makes use of implicit coordination between conflicting aircraft to resolve conflicts. The STCR will have to be able to resolve conflicts which involve several other aircraft and not create new conflicts [5]. The resolution is in the form of a manoeuvre. An update of the RBT and broadcast

of the new intent is performed next to this resolution but is not part of the resolution performed.

The ACAS only uses state information, as it requires a fast resolution of the potential collision [5]. Explicit coordination is used in this conflict pair solving the conflict only for the involved aircraft. It does consider surrounding aircraft within the STAZ, but can ignore secondary conflicts if no other option is available.

Conflict detection and conflict resolution modules work in parallel, using a conflict processing module to assign a detected conflict to the appropriate conflict resolution module [5]. For example, if a conflict is detected using medium term intent-based conflict detection, but requires a short term resolution, STCR is activated.

The A3 ConOps does not in detail describe operations involved in the transition in or out of SSA, but only mentiones some guidelines [5]. The most important is that the aircraft is free of conflict for a to be determined timeframe when entering SSA and is free of conflict when exiting SSA. Furthermore, the A3 ConOps describes military, head-of-state, non-normal and emergency operations.

### 2.2.2. Operational Services and Environment Description

The operational services and environment description is meant to provide a more detailed description of the A3 ConOps by elaborating on the environment in which the A3 ConOps will operate and describing the process of conflict detection and resolution [42].

Inside the environment, aircraft know about each others existence via both Air-Air datalink and SWIM [42]. Other means of communication like voice communication are not considered. SWIM provides the aircraft with the state and intent information of all aircraft within the medium term time horizon (MTTH) [5, 42]. The Air-Air datalink provides the aircraft with the state, and if possible first level intent, information of all aircraft within the short term time horizon (STTH). It is possible to have an SSA with different performance requirements or level of ground support like SWIM. In the OSED three service levels (SL) are considered. A SL is the level of ground support or information available to the autonomous aircraft in SSA. The three service levels are:

- SL1: All autonomous aircraft are broadcasting only state information.
- SL2: All autonomous aircraft conform to SL1 and in addition they broadcast intent information allowing a prediction of the trajectory planned by other aircraft for the MTTH.
- SL3: All autonomous aircraft conform to SL2 and in addition there is a ground information sharing (SWIM) support. This level corresponds to the complete system described in the A3 ConOps [5].

For this study, it is assumed the system is continuously in SL3.

Within the environment, different responsibilities are applied to the different actors [42]. It is the autonomous aircraft's responsibility to broadcast the state and intent information and immediately announce any changes to the RBT due to conflict resolution to SWIM. For the flight crew, the OSED explicitly states: 'all actions which are suggested by onboard support tools and which would directly influence the actually flown trajectory shall at least be approved by the flight crew'. This also means that the flight crew has a proper level of situational awareness provided by the HMI. Next to this, flight operation centres (FOC) are

defined to be airline operation centres and responsible for the planning of flights of their fleet. In the A3 ConOps FOCs are not essential for the safety and feasibility of operations in SSA, but it is envisioned they will play a role for the effectiveness of the A3 ConOps for maximising user benefits. Finally, service providers for SWIM, advanced ground surveillance support, and air navigation (ANSP) are defined. Next to SWIM, which is defined in Section 2.2.1, the advanced ground surveillance support is responsible for providing additional information such as traffic complexity prediction, increasing flight performance, whereas the ANSP controls a TMA and thus the CTA of the aircraft. The A3 ConOps assumes: 'no ANSP intervening during the self separation part of flight (even if self-separation fails)'. If a CTA is to be renegotiated, this is done with the ANSP.

To process the detected conflict and resolve it, certain parameters have to be defined [42]. For a start, the urgency of the conflict at any moment of conflict processing in terms of the time span between the actual time and the predicted LoS. This time span is defined as the time to predicted loss of separation (TTL). Now, to separate MTCR and STCR processes the short term time threshold (STT) is defined as a TTL threshold. Due to the absence of any additional communication among conflicting aircraft, the start of a resolution manoeuvre execution is the first point of onboard conflict processing detectable by surrounding aircraft. Therefore, the operational requirements are refined as follows:

- The resolution manoeuvre which starts at TTL < STT shall fulfill the implicit coordination requirements with respect to the conflicting aircraft. Such conflict is referred as a Short Term Conflict (STC).
- The resolution manoeuvre which starts at TTL > STT shall respect the set priority rules. Such conflict is referred as a Medium Term Conflict (MTC).

Furthermore, to decide whether the detected conflict will be resolved using MTCR or STCR, a conflict processing logic (CPL) is used as described below. In the CPL, open and closed manoeuvres are introduced. A closed manoeuvre is defined to be a conflict resolution manoeuvre provided in the form of an updated RBT within the MTTH. An open manoeuvre on the other hand is a conflict resolution manoeuvre that does not consider a consistent continuation of the existing RBT, but updates its RBT after the conflict resolution has been executed. The conflict resolution in this case is thus a conflict free trajectory within the STTH. The CPL also uses the remaining time to LoS (RTTL), which is the time remaining between the predicted LoS and the moment the resolution manoeuvre is executed. The difference between the TTL and RTTL is the time required to process the detected conflict and come to a decision on which resolution manoeuvre is to be executed. This conflict processing performance (CPP) consists of:

- Surveillance Performance: The delay between the time the information for conflict detection is available and the time the conflict is actually detected;
- Logic Performance: The time taken to assess the situation and deciding on a suitable resolution type;
- Conflict Resolution Performance: The time it takes to generate (possibly multiple) resolution manoeuvres and present them to the flight crew;
- Execution Delay: The time used by the crew to evaluate the resolution manoeuvres presented to them and start execution.

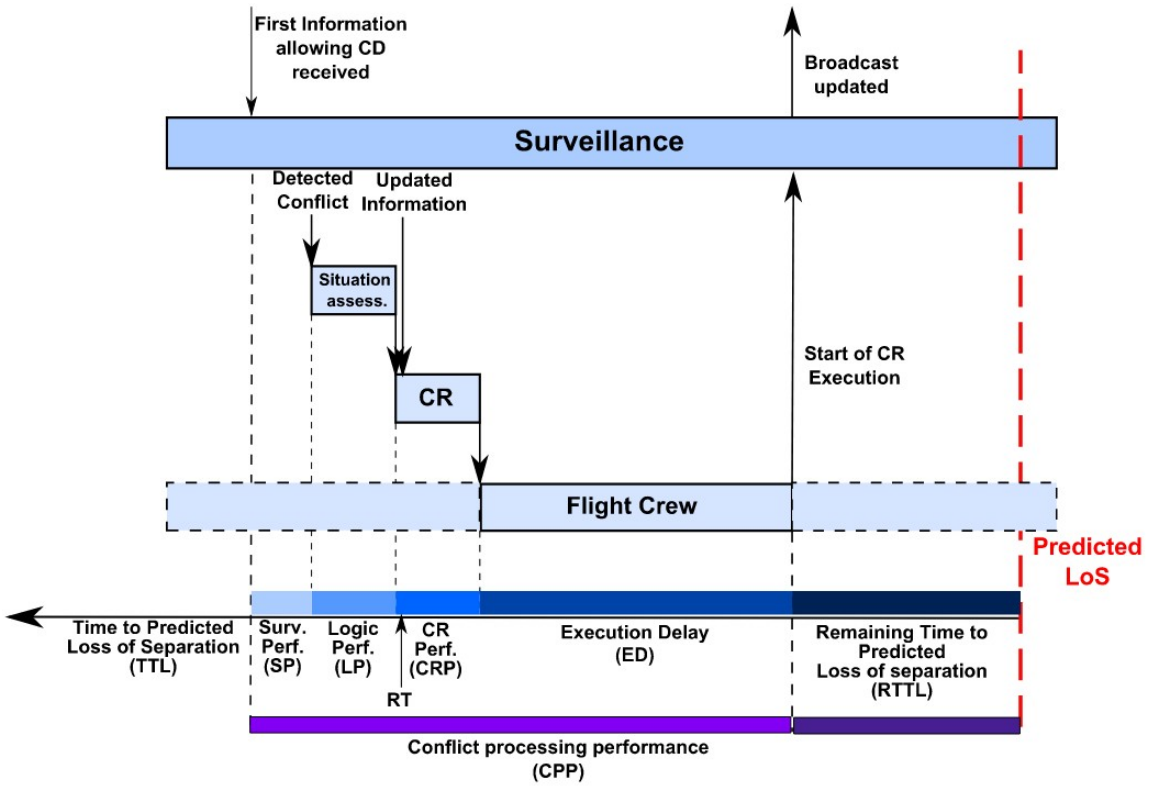



Figure 2.7: On-board conflict processing [42].

A visualisation of the CPP is given in Figure 2.7 [42]. The shown RT (Reference Time) is the time of the traffic situation 'snapshot' used in the conflict resolution. As solutions are presented for acceptance to the flight crew, continuously updating the given resolutions might not be acceptable for the pilots.

Using all these parameters and definitions the CPL will look like Figure 2.8 [42]. When the conflict process ends at enhanced monitoring, this means the conflicting aircraft which is supposed to manoeuvre is monitored to check whether it is actually manoeuvring.

### 2.2.3. System Design

Now that the A3 ConOps and its conflict processing logic is known, the design of the system can be elaborated upon. The system design requirements have been elaborated upon in [43]. The system is shown in Figure 2.9 as a high-level architecture. The contents of the different functional blocks (FB) are elaborated upon below. A functional block is a set of functionalities assuring a group of tasks, for example navigation or conflict resolution.

First, the information management system groups functions dealing with information to and from external systems and agents other than the flight crew [43]. The information management consists of ADS-B communication, SWIM communication, navigation communication, traffic data, and airborne weather data management. The traffic data management fuses information from both ADS-B and SWIM, checks the quality of the information, and serves this information to other ASAS functions. The airborne weather data focuses on long term area detection which can be used to determine areas to avoid.

The application management system is responsible for the overall function orchestra-

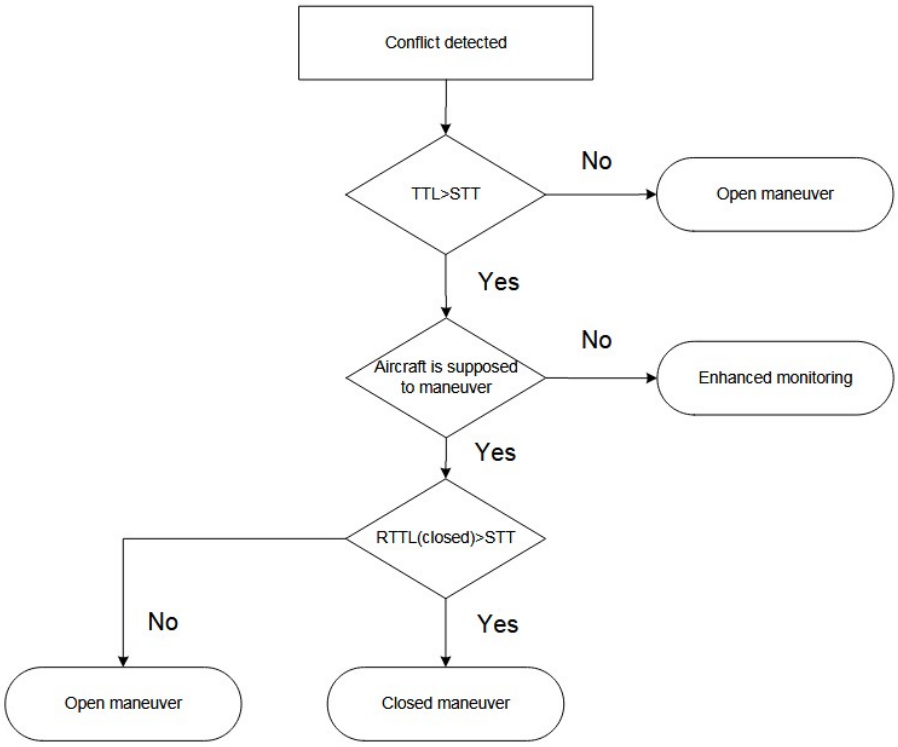



Figure 2.8: Conflict Processing Logic (CPL) [42].

tion, global parameter determination, and operational status supervision [43]. The application management system contains the priority level selection function, service level management, conflict management, system operational status management, and user-preferred trajectory modification management. The conflict management function coordinates other functions related to CD&R, prioritises conflicts, initiates and executes the conflict processing logic and potentially the conflict resolution manoeuvres. The user-preferred trajectory modification management is used to alter the RBT outside of resolution manoeuvres. Finally, the system operational status management checks whether all necessary functions are working as required, and if not manages potential degradation of certain functions.

Next is the conflict detection system, which is responsible for the detection of all types of conflicts and their reporting to the conflict management function in the application management system [43]. This system includes short term conflict detection, medium term conflict detection, complexity conflict detection, and area-to-avoid conflict detection. Conflict detection is intent-based using nominal trajectories.

The conflict resolution system is responsible for computing conflict resolution manoeuvres, and monitoring their execution, whenever solicited by the conflict management function [43]. This system includes functions like the computation of a tactical manoeuvre, the computation of a trajectory modification, and monitoring functions for the execution of either a tactical manoeuvre or a trajectory modification.

The last system is the human machine interface (HMI) [43]. The HMI displays traffic and conflict information to the flight crew, presents proposed conflict resolution manoeuvres, and includes a system operational status interface which displays for example the current priority level and provides information about the ASAS. Finally, the HMI allows the

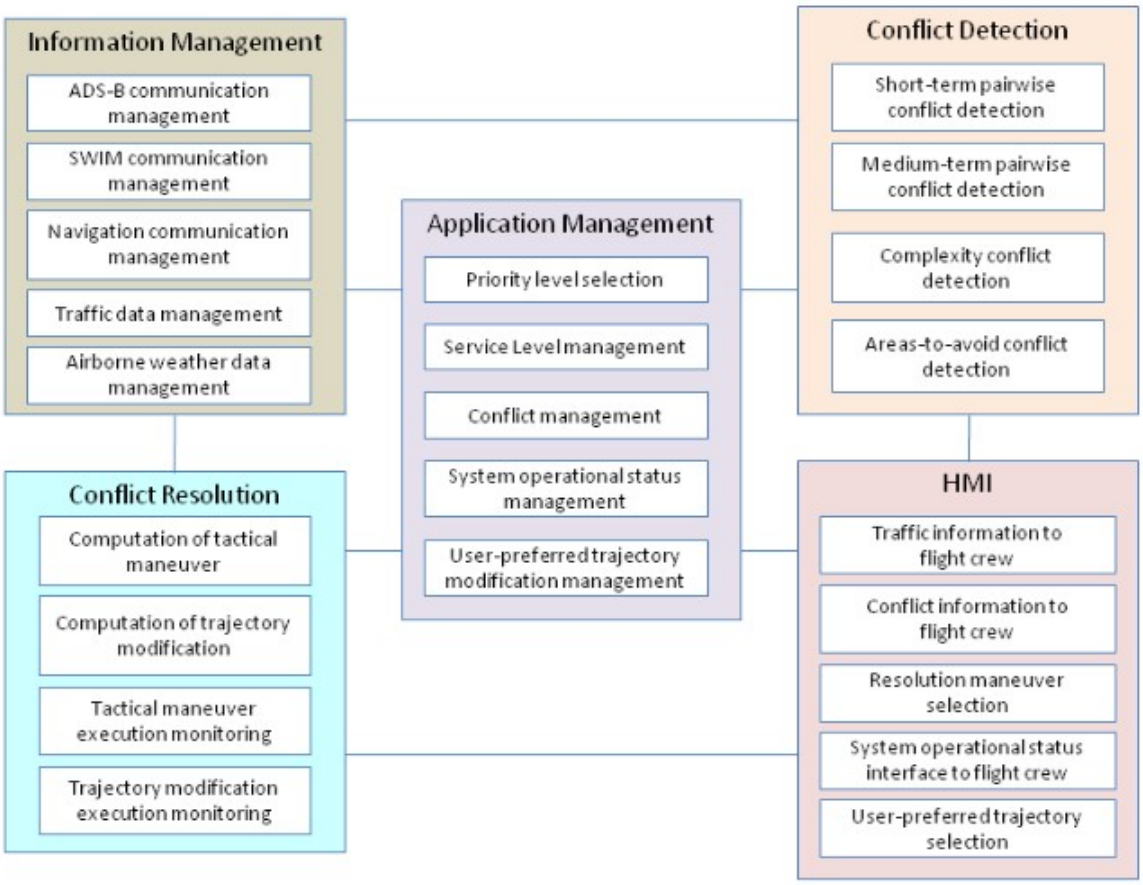



Figure 2.9: High level view of ASAS architecture [43].

flight crew to select a user-preferred trajectory, e.g. for optimisation.

## 2.3. A3 Model

Using the A3 ConOps, a safety evaluation has been performed [8]. In the safety evaluation, the TOPAZ modelling approach has been used to develop a mathematical model of the A3 ConOps [44, 45]. As the first step, an agent-based model of the A3 ConOps has been developed which allows for rare-event Monte Carlo (MC) simulation. Previous research done on rare-event MC simulations can be found in [46, 47]. Using this, it is required that 'the agent-based model satisfies specific mathematical conditions [48]' [8]. In the safety evaluation, these conditions are satisfied by developing the A3 model in the framework of stochastically and dynamically coloured petri nets [49]. An elaboration on Petri nets is given in Appendix 6.1. The A3 model is further elaborated upon in Section 2.3.1. In the safety evaluation, the dynamic velocity obstacle (DVO) method is chosen to perform conflict resolution. The velocity obstacle method is described in Section 2.3.2, and the dynamic velocity obstacle in Section 2.3.3.

### 2.3.1. A3 Model

Within the agent-based formulation of the A3 model, the following agents are considered [50]:

- Aircraft state
- Pilot-Flying (PF)
- Pilot-Not-Flying(PNF)
- Airborne guidance, navigation, and control (GNC)
- ASAS
- communication, navigation, and surveillance systems

It should be noted that the A3 model does not incorporate environment/weather, ACAS, or FOCs [50]. Furthermore, the A3 model ASAS agent is restricted to horizontal CD&R only, thus assuming all aircraft fly at the same flight level.

For the A3 model, the relevant ASAS elements of the A3 ConOps are [50]:

- All aircraft are supposed to be A3 equipped, and their ADS-B periodically broadcasts own aircraft state and intent information, and periodically receives the state and intent information messages broadcasted by other aircraft.
- All aircraft are supposed to use the same resolution algorithm, and all crews are assumed to use ASAS and to collaborate in line with the procedures.
- ASAS related information is presented to the crew through a Cockpit Display of Traffic Information (CDTI) [5].
- Following [5], the aim is to work with a vertical separation minimum of 900 ft and with a horizontal separation minimum of 3 NM (which is referred to as Minimum Separation Zone (MSZ)). A conflict is detected if these separation minima will be violated within medium term or short term horizon. Minimum separation between centre lines of intents are 1000 ft and 5Nm in vertical and horizontal direction respectively (which is referred to as Comfort Separation Zone (CSZ)).
- The conflict resolution process consists of two phases: MTCR and STCR. During the MTCR phase, one of the aircraft crews should make a resolution manoeuvre. If this does not work, then during the STCR phase, both crews should make a resolution manoeuvre.
- Both STCR and MTCR are intent-based, i.e. available intent information of own and other aircraft is taken into account when identifying a conflict free RBT. A key difference between MTCR and STCR is that the former uses priority rules and the latter not.
- During STCR, coordination does not take place explicitly, i.e., there is no communication on when and how a resolution manoeuvre will be executed.

Using these relevant elements, the implementation principles of the MTCR and STCR are given as in [8]:

- MTCR implementation principles:

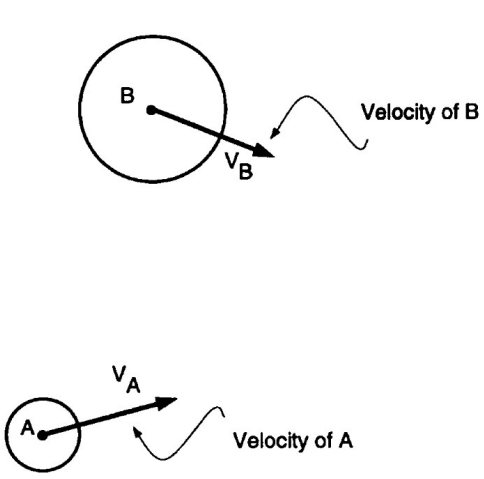
  - MTCR detects planning conflicts 10 min ahead, and it resolves 15 min ahead.
  - An aircraft nearest to the destination has priority over others.
  - An aircraft with the lowest priority has to make its 4-D plan conflict free (15 min ahead) with all other plans.
  - If there is no feasible conflict-free plan, then rather than doing nothing, it is better to identify a plan that has a minimal undershooting of the 5 NM/1000 ft criterion and does not create a short-term conflict.
  - Upon approval by the crew, the aircraft uses automatic dependant surveillance-broadcast (ADS-B) to send this 4-D plan to all other aircraft; if applicable, together with a message of being handicapped (which is priorityincreasing, and it forces other aircraft to help resolve the initial undershooting).

- STCR implementation principles:
  - STCR detects tactical conflicts 3 min ahead and resolves 3 min plus 10 sec ahead through course changes.
  - When an aircraft detects a short-term conflict, it is obliged to resolve the conflict without waiting for any of the other aircraft.
  - If there is no feasible alternative, then rather than doing nothing, it is better to choose a tactical manoeuvre that minimizes the undershooting of the minimum tactical separation criterion.
  - Upon approval of the crew, the aircraft broadcasts its new course, which will trigger other aircraft to help with resolving the conflict in case of an initial undershooting.
  - STCR triggers MTCR to produce a new consistent RBT taking into account the pilot confirmed STCR.

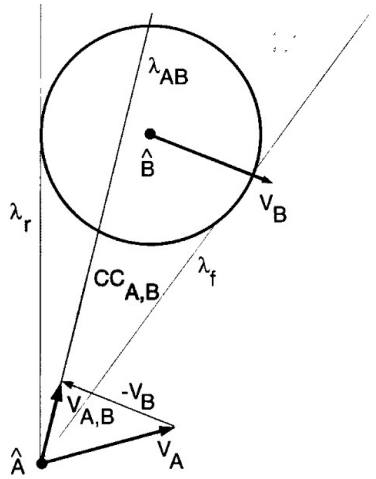
### 2.3.2. Velocity Obstacles

As the A3 ConOps does not describe a conflict resolution algorithm for either MTCR or STCR, for both a specific approach needed to be adopted [8]. As computational load is a severe issue in rare-event MC simulation, the velocity obstacle based conflict resolution algorithm was selected.

The conventional velocity obstacle approach is described in [24]. The velocity obstacle description is restricted to circular agents, thus considering a planar problem with no rotations. It is also assumed that agents move along arbitrary trajectories and that their instantaneous state (position and velocity) is known. Now consider the own aircraft as agent $A$ and another aircraft with circular PAZ as agent $B$, moving with respective velocities $v_A$ and $v_B$, as shown in Figure 2.10a. The collision cone $CC_{A,B}$ is defined as the set of colliding relative velocities between $\hat{A}$ and $\hat{B}$, where $\hat{A}$ is agent $A$ reduced to a point and $\hat{B}$ is obstacle $B$ enlarged by the radius of $A$. This cone is the planar sector with apex in $\hat{A}$, bounded by tangents from $\hat{A}$ to the edge of the PAZ of $\hat{B}$. Any relative velocity outside of $CC_{A,B}$ is guaranteed to be collision free provided that $\hat{B}$ maintains shape and speed. The collision cone is illustrated in Figure 2.10b.

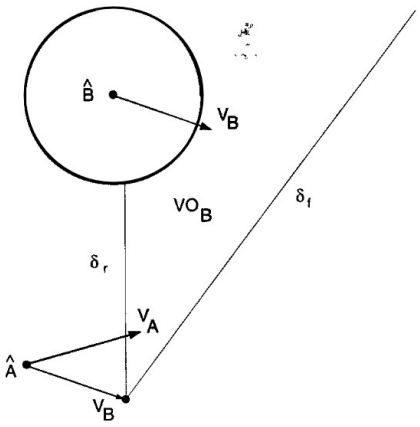



(a) Moving agent $A$ and a moving obstacle $B$.

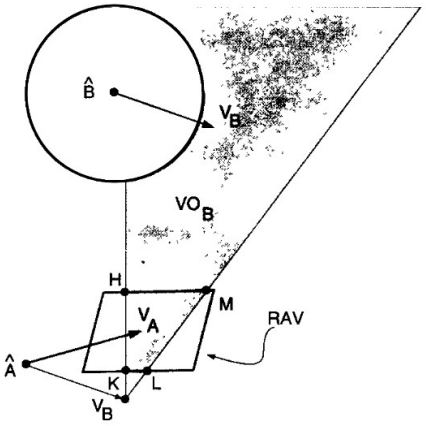



(b) Relative velocity $v_{A,B}$ and collision cone $CC_{A,B}$.

Figure 2.10: Construction of the collision cone [24].




(a) Velocity obstacle $VO_B$.




(b) Reachable avoidance velocities (RAV).

Figure 2.11: Construction of velocity obstacle and consequent reachable avoidance velocities [24].

To consider multiple obstacle agents it is useful to establish an equivalent condition on the absolute velocities of $A$ [24]. This is done by translating $CC_{A,B}$ by $v_B$. This velocity obstacle (VO) partitions the absolute velocities of $A$ into avoiding and colliding velocities. Selecting $v_A$ outside of the VO would avoid collision with $B$. To avoid multiple agents the union of all individual VOs is considered. The avoidance velocities then consist of those velocities $v_A$ that are outside all VOs.

To decide on the avoidance manoeuvre, which consists of a one-step change in velocity to avoid a future collision, a set of reachable velocities is to be defined [24]. Reachable velocities are velocities within the kinematic (e.g. turning radius) and dynamic (e.g. accelerations) constraints of the agent. The set of reachable avoidance velocities (RAV) is then defined as the difference between the set of reachable velocities and the VOs. An avoidance manoeuvre can then be computed by selecting any velocity in the set of RAVs. An application of this theory for solving multi-aircraft conflicts using VOs is the solution space diagram [51], a method to show the VOs to the pilot which combines all VOs into a set of forbidden reachable velocities (FRV) and its complement set of allowed reachable velocities

(ARV) [52]. The construction of the SSD is illustrated in Figure 2.12.

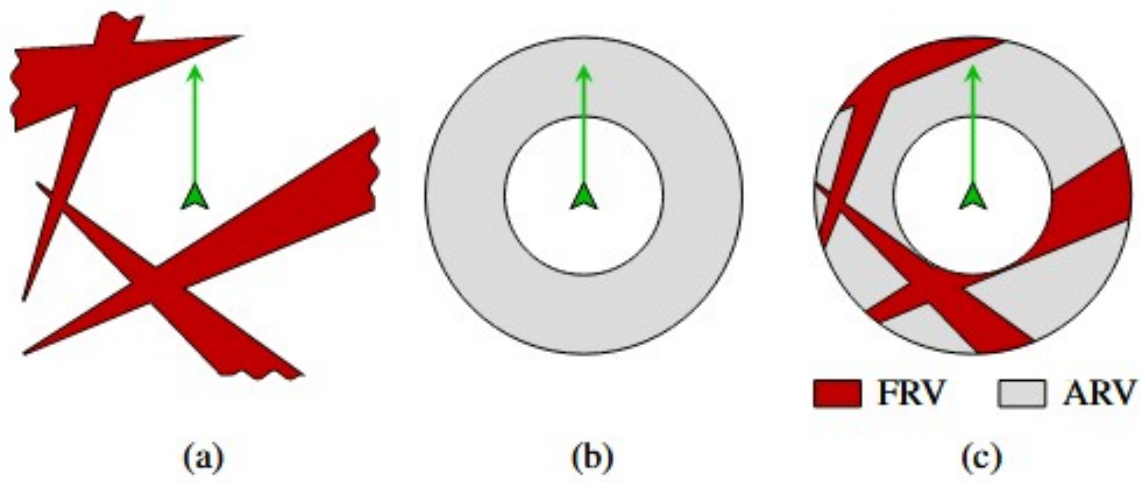



Figure 2.12: Construction of SSD in (c) by combining set of forbidden velocities (or VOs) in (a) and the reachable velocities in (b) [52].

When resolving a conflict using VOs, a choice has to be made by the pilot on which manoeuvre is executed. In Figure 2.13 an example is shown of different coordination rules and their location within an SSD [52]. As can be seen, the different coordination rules are:

1. Shortest way out
2. Clockwise turning
3. Rules of the air
4. Shortest from target heading
5. Only heading change
6. Only speed change

Notice that the resolution point using 'rules of the air' remains within the FRV as the conflicting aircraft approaches from the left, and the depicted aircraft thus has the right of way [52]. Different rules can lead to the same resolution point. Additionally, rules five and six do not garantuee a solution and thus have to be used in conjuncture with the other rules. The forbidden resolution velocities (FRV) are given in red, whereas the available resolution velocities (ARV) are the white space.

These 6 coordination rules solve the conflicts in the SSD in a joint solution. However the conflicts can also be solved in a sequential manner, making use of priority to determine which aircraft should solve which conflict [52].

### 2.3.3. Dynamic Velocity Obstacles

Now the above described VOs are state-based, but considering the intent-based nature of the A3 model it is interesting to look at applications considering intent information for VOs. An example of this is the Dynamic VO (DVO) method [8, 53]. The DVO method takes the VO

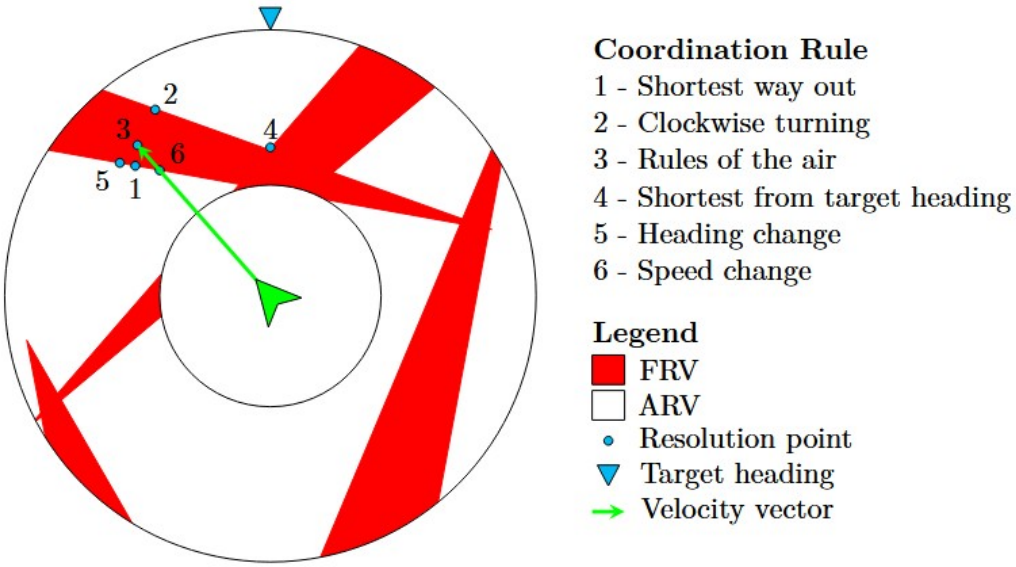
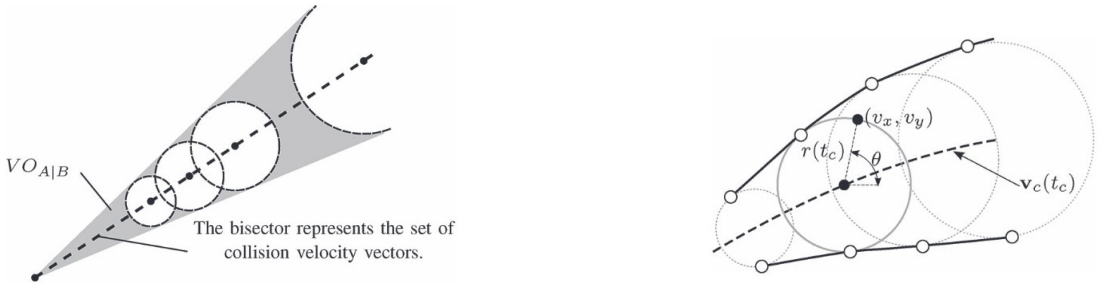



Figure 2.13: Resolution points under different coordination rules [52].

to be a family of circles with their centres at $\vec{v}_c(t_c)$ and radii $r(t_c)$ as illustrated in Figure 2.14. $\vec{v}_c(t_c)$ is the exact collision course from agent $A$ to agent $B$.




(a) VO as a family of cicles.

(b) Family of circles with centres on $\vec{v}_c(t_c)$ and radii $r(t_c)$.

Figure 2.14: VO described as a family of circles [53].

Another example is to consider that a complex trajectory can be approximated as a series of straight paths [8, 54]. A curved trajectory can be approximated using virtual vehicle positions, or 'ghosts', which can be extrapolated backward along each of the approximated straight sections. The virtual positions can be used to generate a VO for that part of the moving obstacle's trajectory. This is illustrated in Figure 2.15.

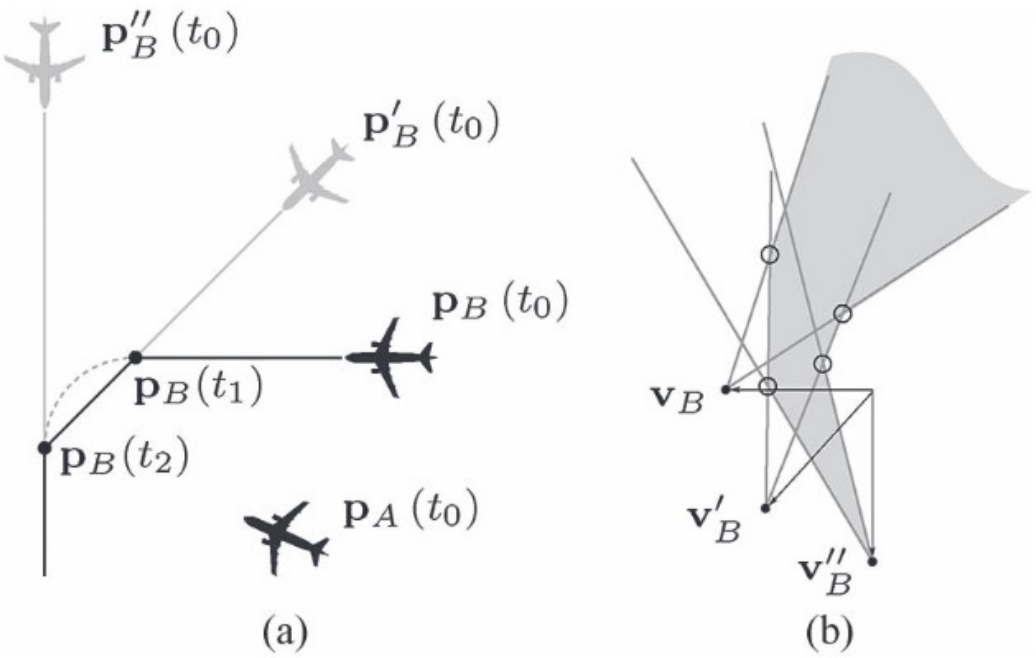



Figure 2.15: Approximation of the curved trajectory of obstacle *B* using 'ghosts' in (a), and consequent VO for agent *A* in (b) [53].

The safety evaluation of Blom and Bakker (2015) also uses such a DVO approach [8]. In this case, the DVOs follow from the RBTs, each of which consists of a centerline and a 3D volume around this centerline. This means that the 3D volumes of RBTs define the DVOs. The technical details of the RBT-DVO approach used by Blom and Bakker (2015) have been specified in the unpublished NLR Technical Report by Bakker, Nieskens and Blom (2010) [55].

## 2.4. Simulator

When looking at different papers on CD&R models and their analysis, it is clear that simulation of air traffic with ASAS is the main way to verify and validate the CD&R model and its performance. This is shown in for example [8] or [18]. In these cases the model is verified and validated using a simulator. The different simulators have different fidelity levels, or use different aircraft performance models. To test the performance of the A3 model and be able to compare it to other CD&R models one simulator has to be used.

The chosen simulator for this research is the BlueSky ATM simulator [56]. This simulation tool is open-data, using for example self-generated text files containing Aeronautical Information Services (AIS) as navigation data. Furthermore, the simulator is open-source which allows the community to contribute to the development. To allow for the simulator to be open-source, it is build using Python, which is a popular open-source programming language used by many academia. Furthermore, used data files are plain text files which contributes to its ease of use. Simulation scenarios can also be described in text files, using the traffic program manager used by for example NLR and NASA as part of the ASTOR simulation [57]. This program uses plain text scenario files with time stamped commands, that

can also be entered during runtime by the user, in the console of the simulation program. Next to this, the scenario files can easily be shared to be used in future research.

Next to the scenario files, plugins can be built and implemented inside the simulator. Such plugins add functionalities to the simulator which are not available at that moment. For this research, the A3 model will be implemented using one (or maybe more) plugins in BlueSky. Other conflict resolution algorithms have already been implementend using such a plugin[1]. An overview using the taxonomy described in Section 2.1 is given in

Table 2.6: Conflict resolution methods in BlueSky and their taxonomy.

| Method | Surv | Traj | PAsm | Cont | Multi | Plan | Obst | AvMan | Source |
|---|---|---|---|---|---|---|---|---|---|
| MVP | D | S | N | D | PSU | T | D | H+S+V | [20, 18] |
| SSD | D | S | N | A | J | T | A | H+S | [52] |

## 2.5. Simulation Scenarios

To evaluate a CD&R method in a simulator, certain scenarios can be used to allow for a fair comparison. In this section, scenario variables and how they are used in different papers are described. First, different traffic scenarios are introduced in Section 2.5.1. Next, independent variables are elaborated upon in Section 2.5.2. Finally, some dependent variables are discussed in Section 2.5.3.

### 2.5.1. Traffic Scenarios

As a start, a simulation has a certain amount of aircraft or agents within the simulated space. This number can vary a lot. For example, when looking at validation of a CD&R method, simulations can contain two aircraft flying towards the same point in space, illustrated in Figure 2.16 [18, 19]. Other two aircraft conflicts are used as case studies for the DVO, for example by including manoeuvres, and are shown in Figure 2.17.

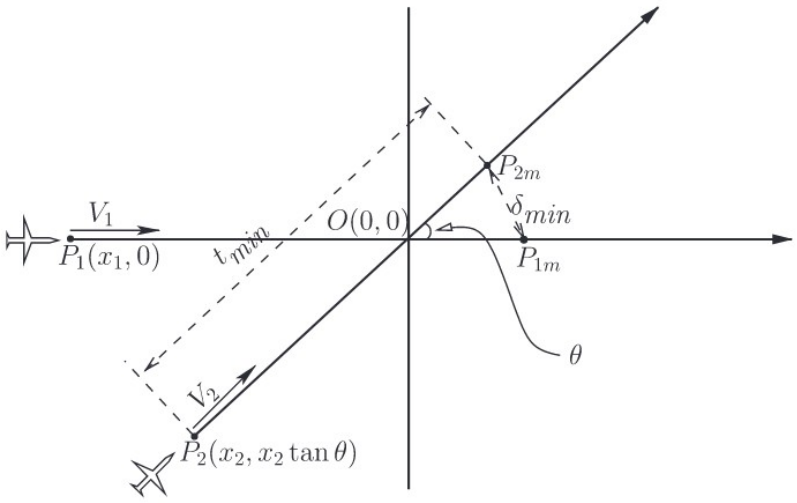
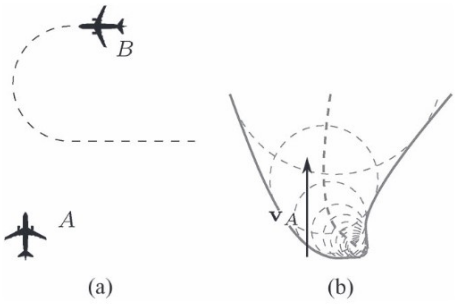
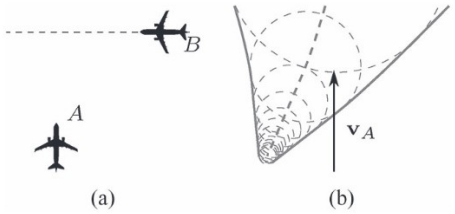



Figure 2.16: Example of a two aircraft conflict [38].

[1] https://github.com/TUDelft-CNS-ATM/bluesky/tree/master/bluesky/plugins/asas



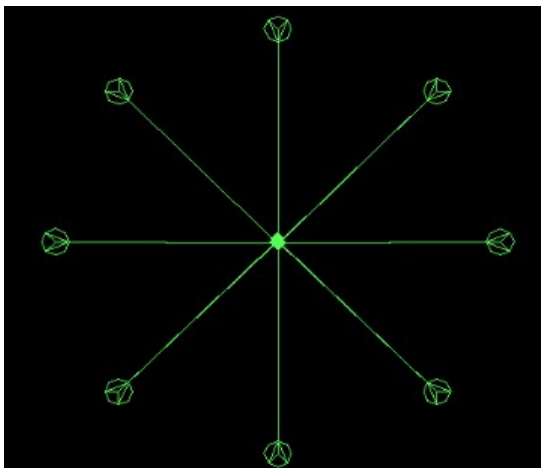

(a) Basic heading change manoeuvre (a) and corresponding DVO (b).



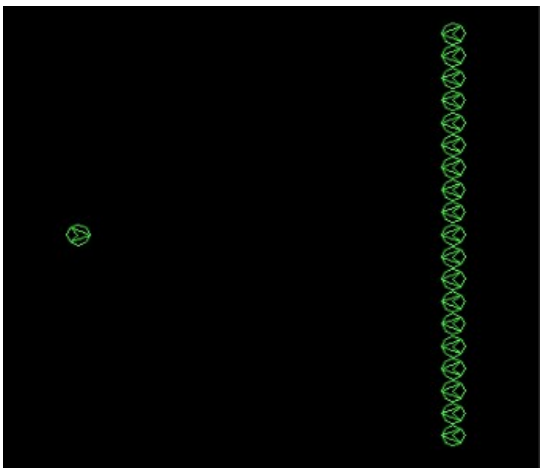

(b) Basic acceleration manoeuvre (a) and corresponding DVO (b).

Figure 2.17: Two aircraft conflicts including manoeuvres [53].

With a few more aircraft, more complex conflict geometries can be made, e.g. the super conflict with eight aircraft visualised in Figure 2.18a [18, 4, 23] or the wall conflict visualised in Figure 2.18b [4, 23].



(a) Example of a superconflict.



(b) Example of a wall conflict.

Figure 2.18: Examples of special conflict geometries [4].

Now when looking at large scale traffic scenarios, an area has to be defined in which the simulation will take place. For example, Balasooriyan et al. uses a square area of $455,625 NM^2$ [52], whereas Ribeiro et al. define a square area of $202,500 NM^2$ [10]. The measurement area does not need to be chosen arbitrarily. This can be done using the average true air speed (TAS) and the average flight time, giving a square area [10]. However, this is the measurement area. Both define an experiment area larger than the measurement area to account for edge effects, for example aircraft leaving the measurement area due to resolution manoeuvres. The aim is to keep the traffic density constant within the measurement area. In these simulations, aircraft are spawned continuously.

Another way of defining a simulation area is described in [8]. Here, the area is packed with rectangular boxes. Each box contains a fixed number of seven aircraft which fly at an arbitrary position and arbitrary direction at a certain ground speed. One aircraft aims to fly straight through a sequence of connected boxes at the same speed. Per box, the aircraft within it behave the same, and for aircraft that pass the boundary of a box a periodic

boundary condition applies. This means that all aircraft have to be simulated in one box only. By changing the size of the boxes, the traffic density can be varied. The boxes should not become too small to avoid that an aircraft experiences a conflict with its own copy in a neighbouring box. In such a simulation, all aircraft are spawned immediately.

Both types of simulations may need a build-up period. Either to reach a steady state of traffic volume and traffic pattern [10], or to solve serious short and medium term conflicts which appear due to the immediate nature of spawning [8].

Finally, the aircraft in the simulation can be defined in certain different ways. For example, the aircraft can be based on a real aircraft (e.g. B747 [52, 10] or B737-500 [23]). Another option is to simulate more anonymous agents, giving a speed and altitude normal for manned aviation purposes [8]. More reality based simulation scenarios use a mix of aircraft types and thus performance values [39].

### 2.5.2. Independent Variables

In this Section, the independent variables are described. These variables are independent because they are not influenced by other parameters, and can thus be varied to make different analysable scenarios. These variables do have influence on the dependent variables explained below.

1. Conflict detection & resolution methods

Simulations in BlueSky use the already implemented state-based conflict detection method [52, 10]. Another possibility is to use intent-based conflict detection [23, 8].

Furthermore, the resolution method can also be varied. For example, implementations in BlueSky can lead to a comparison of different resolution methods [52, 10]. Within the resolution method, the type of manoeuvre can also be varied, for example by giving a range of speed controls in subliminal speed control or defining certain coordination rulesets when using VOs [39, 52].

2. Traffic volume

To increase pressure on the CD&R models during simulations, the traffic volume can be increased which will lead to an increase in the amount of conflicts. Here a wide range of traffic densities per flight level have been considered

- 2.5 to 7.5 $AC/10,000NM^2$ [52]
- 20.8 $AC/10,000NM^2$ and 41.6 $AC/10,000NM^2$ [8]
- 32 to 45 $AC/10,000NM^2$ [10]

3. Manoeuvring space

The performance of the CD&R models will be dependent on how constrained the agents in the simulation are in their manoeuvres. This can include restrictions on turn rate, speed, and altitude. A speed range for manned aviation between 450 and 500$kts$ can be used [52, 10], but a larger speed range of 300 to 500$kts$ is also possible and increases the available manoeuvring space [52].

Most simulations encountered use a horizontal resolutions only approach [8, 52, 10]. Others evaluate horizontal and vertical separately, where in theory the resolution method can choose [23].

A final way to limit the manoeuvring space is to implement static obstacles within the experiment area, creating corridors in which the agents have to manoeuvre [32].

4. Wind

Next, an independent variable is the wind. Some simulations included no wind [52, 10]. Some do include an invariant wind vector field [58]. Others include a wind prediction error trying to measure the effect of unexpected local wind or weather disturbances [8].

5. Various disturbances

Finally, other independent variables can be set as a probability of failure of for example the enabling systems for the CD&R [8]. Such baseline values of key dependability parameters can cause various disturbances with which the CD&R model has to be able to deal with in the real world. Below a list is given of the dependability parameters used in the safety evaluation [8]:

- Probability of global navigation surveillance system down
- Probability of global ADS-B down (frequency congestion/data transfer overload)
- Probability of aircraft ADS-B receiver down
- Probability of aircraft ADS-B transmitter down
- Probability of aircraft ASAS performance corrupted
- Probability of aircraft ASAS down

### 2.5.3. Dependent Variables

3 categories of dependent variables can be used to compare different CD&R methods, which are loss of separation, stability, and efficiency [58].

1. Loss of Separation

The obvious aim of a CD&R method is to allow safe self separation. This can be measured in the number conflicts. A conflict is defined as a predicted intrusion. The safety can also be measured as the number of intrusions. An intrusion is defined as a violation of separation minima. These metrics can either be shown as the total number found during the simulation or as a frequency per flight hour [8, 10].

Using these numbers, the intrusion prevention rate (IPR) can be computed. The IPR can be computed as in Equation 2.1. Here $n_{cft}$ is the number of conflicts and $n_{int}$ is the number of intrusions.

$$IPR = \frac{n_{cft} - n_{int}}{n_{cft}} \tag{2.1}$$

Other metrics can be the severity of LoS. This can be measured as a fraction of the intrusion compared to the horizontal separation minima. This is computed in Equation 2.2 [52]. $R$ is the horizontal separation minimum and $d_{CPA}$ the closest distance between two aircraft.

$$LoS_{sev} = \frac{R - d_{CPA}}{R} \tag{2.2}$$

Furthermore, another metric to measure the severity of the losses of separation is the number of intrusions going under 2/3 of the separation minima [8]. This metric can be compared to the number of intrusions to find whether the intrusions were severe, or the closest point of approach was just inside the PAZ. Here, the number of intrusions going under 2/3 of the separation minima can be shown as the number during a simulation or the frequency per flight hour.

2. Stability

When at very high traffic densities, resolving conflicts may cause new conflicts [58]. The stability of the airspace as a direct result of conflict resolution manoeuvres can be measured using the domino effect parameter (DEP). The DEP is visualised in Figure 2.19. The DEP is computed using Equation 2.3.

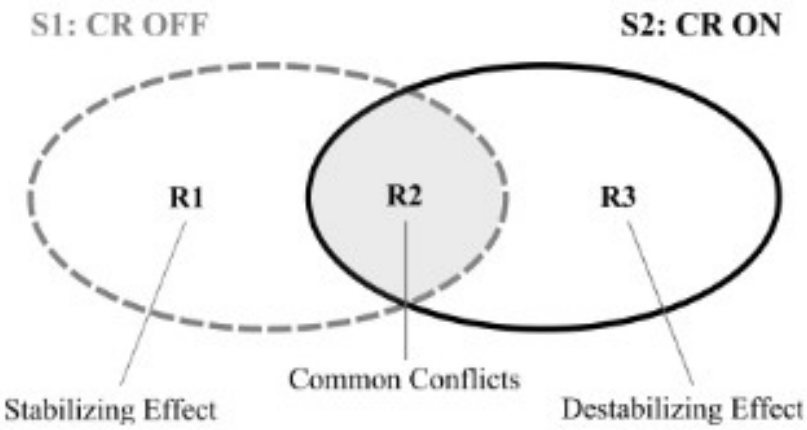



Figure 2.19: The DEP compares simulations with and without conflict resolution to measure airspace stability [58].

$$DEP = \frac{R3 - R1}{S1} = \frac{S2}{S1} - 1 \tag{2.3}$$

3. Efficiency

Finally, it is interesting to know whether a model can not only allow safe separation, but also do so in an efficient manner. A first metric is the distance travelled. This can be compared to the minimum distance required to find the loss in effective distance travelled. Summed for the total of all flights in the scenario, the efficiency can be described by the mean loss in effective distance travelled [8]. This metric can also be expressed in time.

Another efficiency metric is the mean lateral deviation at the endpoint of the simulation period. This provides a measure of the net effect of the detours made due to conflict resolutions [8].

The efficiency in a simulation can also be analysed using the work done metric [58]. This metric considers the optimality of the aircraft's trajectory, having a strong correlation with fuel/energy consumption. The work is computed using the thrust vector $\vec{T}$ and the displacement vector $\vec{s}$, as shown in Equation 2.4. This can then be compared to the minimum work required for the flight.

$$W = \int_{path} \vec{T} \cdot d\vec{s} \tag{2.4}$$

Finally, the efficiency of a CD&R method can be analysed by looking at the pilot activity. Blom and Bakker (2015) assessed pilot activity in terms of number of pilot-flying and pilot-not-flying activities per flight hour [8].

# 3

# Research Gap & Questions

After the given overview of relevant literature, the identified research gap is defined. Additionally, resulting research questions are provided. The research gap can be found in section 3.1 and the research questions in section 3.2.

## 3.1. Research Gap

When looking at the literature study, it is noticed that a lot of research has been performed on CD&R methods. Both the development of CD&R methods and the analysis of their performance has seen a lot of research over the past years. However, some gaps can be identified.

First, the comparability of different CD&R methods, both state-based and intent-based, is lacking. It is unclear how CD&R methods perform compared to each other. This is due to different ATM simulators and scenarios used in different papers. This includes a comparison between state-based and intent-based CD&R methods. This is required to analyse the impact of using intent-information in conflict detection and resolution.

Next, how to set up simulation scenarios to study intent-based CD&R methods is missing. Furthermore, this is needed to be able to compare them fairly to other CD&R methods. How is intent information incorporated in simulation scenarios? How to measure the impact of intent information on conflict detection and resolution? And what variables describe the performance of an intent-based CD&R method?

This research aims to fill the gaps defined above by implementing the A3 model in BlueSky. Then use this implementation to do a comparative study to other existing CD&R models.

## 3.2. Research Questions

In this section the main research question is presented, which is followed by the subquestions and some elaboration on how together they will answer the main research question.

### Main Research Question

*How are traffic scenarios developed which include intent information and explore a range of independent variables and rare events to research the advantages and disadvantages of different CD&R models?*

## Research Subquestions

1. *What is the impact of using intent information when analysing CD&R methods?*

2. *Which performance indicators should be chosen to perform a comparative study between different intent-based CD&R models in BlueSky?*

# 4

# Methodology

In this section, a brief description is given on how the thesis research will progress from this proposal. A preliminary methodology based on the research questions given in Section 3.2 is given in Section 4.1

## 4.1. Methodology

The steps needed to be taken to perform this research are the following:

1. Implement intent information in BlueSky.

As a first step intent information has to be implemented into the vectorised operations of the BlueSky ATM simulator. The required intent information of aircraft is key to implement and simulate the A3 CD&R model.

2. Implement A3 intent-based conflict detection algorithm in BlueSky.

Now, aircraft in BlueSky have to be able to detect potential conflicts using intent information. Current conflict detection in BlueSky is state-based only [1].

3. Implement both A3 intent-based STCR and MTCR algorithms in BlueSky.

When a conflict is detected, either the MTCR or STCR is to resolve it. Both MTCR and STCR algorithms will develop an intent-based resolution, where MTCR will give a conflict free trajectory for the MTTH and STCR will give a conflict free trajectory for the STTH.

4. Develop traffic scenarios which include intent information.

Next, to analyse the performance of the A3 model and make a comparison with other conflict resolution algorithms implemented in BlueSky, traffic scenarios have to be developed. These traffic scenarios have to at least include a variety of traffic densities, time horizons for CD&R, and PAZ dimensions.

5. Create and analyse results using the developed scenarios for both the A3 model and other conflict resolution models implemented in BlueSky based on chosen performance indicators.

Using the developed traffic scenarios, both the A3 model and other conflict resolution models implemented in BlueSky can be simulated. Results on the chosen performance indicators are created and Analysed.

[1] https://github.com/TUDelft-CNS-ATM/bluesky/wiki/asas

# 5

# Conclusion

The global air traffic continues to grow. To increase the ATM capacity, the concept of Free Flight is a promising development. To allow for Free Flight, an Autonomous Aircraft Advanced (A3) concept of operations has been developed. This concept of operations uses intent information to combine strategic and tactical conflict detection and resolution. It allows safe operations in a self separating airspace.

Yet it is unclear how this A3 model performs compared to other CD&R models. Thus the impact of combining strategic and tactical CD&R using intent information is unclear.

This research aims to answer this question by implementing the A3 model in the BlueSky ATM simulator. This implementation is used to perform simulations. By doing so, it will improve understanding of the A3 model. Furthermore, it enables future research in a comparative simulator environment. This will ultimately provide a means of implementing the concept of Free Flight in air traffic management.

# 6

# Appendix 1

## 6.1. Petri Nets

To mathematically describe the A3 model and be able to use it within rare-event MC simulations, Dynamically Coloured Petri Nets (DCPN) are used [8]. The total A3 model is a composition of Petri nets per functional entity of an agent, defined as a Local Petri Net (LPN). Below DCPNs and LPNs are described.

**Dynamically Coloured Petri Net**

A petri net is a graph of circles (named places), rectangles (named transitions), and arrows (named arcs) [49]. The arcs exist between places and transistions and vice versa. The places represent discrete modes or conditions, the transitions represent possible actions: they indicate transitions between those conditions. A condition is current if a token (represented by a dot) is residing in the corresponding place.

The many advantages of petri nets and their extensions include their graphical representation which makes it possible to observe or model a system in all of its components at a higher level, and their applicability to dynamic process models [49]. The extension used for the A3 model is the DCPN. A DCPN is given by the following tuple: DCPN = $(P, T, A, N, S, C, V, G, D, F, I)$, where:

$P$ is a set of places

$T$ is a set of transitions which consists of a set of guard transitions ($T_G$), a set of delay transitions ($T_D$), and a set of immediate transitions ($T_I$)

$A$ is a set of arcs which consists of a set of ordinary arcs ($A_o$), a set of enabling arcs ($A_e$), and a set of inhibitor arcs ($A_i$)

$N$ is a node function which maps each arc to an ordered pair of one transition and one place; multiple arcs between the same place and transition are allowed

$S$ is a set of colour types for the tokens occuring in the net; a colour is the value of a token, a colour type is thus the state space of the colours a token can be

$C$ is a colour function which maps each place to a colour type in $S$

$V$ is a set of place-specific colour functions which describe what happens to (i.e. define the rate of change of) the colour of a token while it is residing in that place; each colour function is determined by a differential equation which is locally Lipschitz continuous

$G$ is a set of Boolean-valued transition guards associating each transition in $T_G$ with a guard function

$D$ is a set of transition delays associating each transition in $T_D$ with a delay function

$F$ is a set of (probabilistic) firing functions which for each transition describes the quantity and colours of the tokens produced by the transition at its firing; a transition will either fire 0 or 1 token per outgoing arc which together with its colour is according to a transition-specific probabilistic mapping rule that may depend on the colours of the input tokens

$I$ is an initial marking which defines the set of tokens initially present, at which place they initially reside, and what colour they initially have

Tokens can thus be removed from places by transitions that are connected to these places by incoming ordinary arcs [49]. However, two conditions have to be satisfied. First, the transition must have at least one token per ordinary arc and one token per enabling arc in each of its input places and no tokens in the input places to which it is connected by an inhibitor arc. If this condition is satisfied the transition is pre-enabled. The second condition differs per type of transition; the condition for guard transitions is specified by the set of transition guards, for delay transitions by the set of transition delays, and no second condition is specified for immediate transitions. The transition guards and transition delays are continuously evaluated if the transition is pre-enabled, and may depend on the colours of the tokens in the input places.

When both conditions are satisfied the transition removes a token from all input places connected by an ordinary arc [49]. It does not remove the tokens from input places connected by an enabling arc. The transition then produces a token for some or all of its output places, specified by the firing function. The colours of the tokens produced by the transition may depend on the colours of the tokens in the input places.

In order to avoid ambiguity, for a DCPN the following rules apply when two transitions are enabled simultaneously:

R0 The firing of an immediate transition has priority over the firing of a guard or delay transition.

R1 If one transition becomes enabled by two or more disjoint sets of input tokens at exactly the same time, then it will fire these sets of tokens independently, at the same time.

R2 If one transition becomes enabled by two or more non-disjoint sets of input tokens at exactly the same time, then the set that is fired is selected randomly.

R3 If two or more transitions become enabled at exactly the same time by disjoint sets of input tokens, then they will fire at the same time.

R4 If two or more transitions become enabled at exactly the same moment by non-disjoint sets of input tokens, then the transition that will fire is selected randomly, with the same probability for each transition.

**Local Petri Net**

The composition of a complete DCPN for a complex process with many different interacting agents such as in ATM is a bottom-up process. Each low level functional entity of each agent is to be defined as a Local Petri Net (LPN). All LPNs are then connected such that the high level architecture of the agent, and subsequently the complete process is defined. The LPNs are connected in such a way that the number of tokens residing in a LPN is not influenced by these interconnections.

The interconnections between different LPNs have to be specified in a way that allows to start at the lowest level and then step by step going up to the highest level, such that an interconnection at a higher level does not imply a significant change at a lower level. Two types of interconnections can be identified:

- Enabling (or inhibitor) arc from one place in one LPN to one transition in another LPN.
- Interaction Petri Net (IPN) from one (or more) transition(s) in one LPN to one (or more) transition(s) in another LPN.

When LPNs are interconnected by an enabling or inhibitor arc the transition in LPN B can only fire when LPN A is in a particular state. When the transition in LPN B fires, it may use the information existing in LPN A. For example, if the transition in LPN B depends on the colour of the token in LPN A.

An IPN consists of at least one place, and zero or more transitions. It connects LPNs by ordinary arcs, thus a token moves to and from the IPN. If there are any transitions in the IPN, and those transitions are connected with other LPNs, only enabling or inhibitor arcs can be used for those connections with other LPNs. IPNs are used when enabling or inhibitor arcs are insufficient to model interconnections. For example, it can hold on to state information from its input LPN while the state of the LPN itself evolves further. Next to this, an IPN connects a transition to a transition whereas an enabling or inhibitor arc connect a place to a transition.

# Bibliography


[1] *Deep Change*. IATA, June 2024, pp. 1–24. URL: https://www.iata.org/en/iata-repository/publications/economic-reports/global-outlook-for-air-transport-june-2024-report/.

[2] Richard McCausland. *Net zero carbon 2050 resolution - Fact Sheet*. Oct. 4, 2021. URL: https://www.iata.org/en/iata-repository/pressroom/fact-sheets/fact-sheet-iata-net-zero-resolution/.

[3] Xavier Fron, Piotr Ikanowicz, José Miguel de Pablo Guerrero, Ismail Hakki Polat, Michael Steinfurth, Daniel Weder, and Peter Whysall. *Performance Review Report: An Assessment of Air Traffic Management in Europe*. Performance Review Report. Eurocontrol, May 2024, pp. 1–134. URL: https://www.eurocontrol.int/sites/default/files/2024-06/eurocontrol-performance-review-report-2023.pdf (visited on 11/18/2024).

[4] JM Hoekstra, RCJ Ruigrok, and RNHW van Gent. "Free Flight in a crowded Airspace?" In: *Air Transportation & System Engineering*. Ed. by GL Donohue and AG Zellweger. American Institute of Aeronautics and Astronautics Inc. (AIAA), 2000, pp. 533–546. ISBN: 978-1-56347-474-3. URL: https://www.academia.edu/60698336/Free_flight_in_a_crowded_airspace?b=50_percent_vector.

[5] Gustavo Cuevas, Ignacio Echegoyen, José García García, Petr Cásek, Claudia Keinrath, Rosa Weber, Petr Gotthard, Frank Bussink, and Aavo Luuk. *iFly D1.3: Autonomous Aircraft Advanced (A3) ConOps*. D1.3. Jan. 30, 2010, pp. 1–130. URL: https://ifly.nlr.nl/documents/D1.3_Final_3.0_29_Jan_2010.pdf.

[6] Mark Ballin, Jacco Hoekstra, David Wing, and Gary Lohr. "NASA Langley and NLR Research of Distributed Air/Ground Traffic Management". In: *AIAA's Aircraft Technology, Integration, and Operations (ATIO) 2002 Technical Forum*. AIAA's Aircraft Technology, Integration, and Operations Technical Forum. Aviation Technology, Integration, and Operations (ATIO) Conferences. Los Angeles, California: American Institute of Aeronautics and Astronautics, Oct. 2002, pp. 1–23. DOI: 10.2514/6.2002-5826. URL: https://arc.aiaa.org/doi/10.2514/6.2002-5826 (visited on 11/18/2024).

[7] SESAR Joint Undertaking. *SESAR Joint Undertaking | MP 1 executive summary*. European ATM Master Plan. URL: https://www.sesarju.eu/MasterPlan2025 (visited on 12/19/2024).

[8] Henk A. P. Blom and G. J. Bakker. "Safety Evaluation of Advanced Self-Separation Under Very High En Route Traffic Demand". In: *Journal of Aerospace Information Systems* 12.6 (2015). Publisher: American Institute of Aeronautics and Astronautics _eprint: https://doi.org/10.2514/1.I010243, pp. 413–427. ISSN: 1940-3151. DOI: 10.2514/1.I010243. URL: https://doi.org/10.2514/1.I010243 (visited on 10/07/2024).

[9] J.K. Kuchar and L.C. Yang. “A review of conflict detection and resolution modeling methods”. In: *IEEE Transactions on Intelligent Transportation Systems* 1.4 (Dec. 2000). Conference Name: IEEE Transactions on Intelligent Transportation Systems, pp. 179–189. ISSN: 1558-0016. DOI: 10.1109/6979.898217. URL: https://ieeexplore.ieee.org/document/898217 (visited on 10/07/2024).

[10] Marta Ribeiro, Joost Ellerbroek, and Jacco Hoekstra. “Review of Conflict Resolution Methods for Manned and Unmanned Aviation”. In: *Aerospace* 7.6 (June 2020). Number: 6 Publisher: Multidisciplinary Digital Publishing Institute, pp. 1–37. ISSN: 2226-4310. DOI: 10.3390/aerospace7060079. URL: https://www.mdpi.com/2226-4310/7/6/79 (visited on 10/07/2024).

[11] D. Sui, Z. Zhou, and X. Cui. “Priority-based intelligent resolution method of multi-aircraft flight conflicts”. In: *The Aeronautical Journal* (Oct. 16, 2024), pp. 1–25. ISSN: 0001-9240, 2059-6464. DOI: 10.1017/aer.2024.75. URL: https://www.cambridge.org/core/journals/aeronautical-journal/article/prioritybased-intelligent-resolution-method-of-multiaircraft-flight-conflicts/DC433FD78EC1D7EC8F9714F7CAF91489 (visited on 11/20/2024).

[12] Dong Sui and Kai Zhang. “A Tactical Conflict Detection and Resolution Method for En Route Conflicts in Trajectory-Based Operations”. In: *Journal of Advanced Transportation* 2022.1 (2022). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1155/2022/9283143, pp. 1–16. ISSN: 2042-3195. DOI: 10.1155/2022/9283143. URL: https://onlinelibrary.wiley.com/doi/abs/10.1155/2022/9283143 (visited on 11/28/2024).

[13] Marc Brittain and Peng Wei. “One to Any: Distributed Conflict Resolution with Deep Multi-Agent Reinforcement Learning and Long Short-Term Memory”. In: *AIAA Scitech 2021 Forum*. AIAA SciTech Forum. Online: AIAA, 2021, pp. 1–10. DOI: 10.2514/6.2021-1952. URL: https://arc.aiaa.org/doi/epdf/10.2514/6.2021-1952 (visited on 12/04/2024).

[14] Nicola Piccinelli, Federico Vesentini, and Riccardo Muradore. “MPC Based Motion Planning For Mobile Robots Using Velocity Obstacle Paradigm”. In: *2023 European Control Conference (ECC)*. 2023 European Control Conference (ECC). Bucharest: EUCA, June 2023, pp. 1–6. DOI: 10.23919/ECC57647.2023.10178219. URL: https://ieeexplore.ieee.org/document/10178219/?arnumber=10178219 (visited on 11/28/2024).

[15] P. S. Naga Jyotish, Yash Goel, A. V. S. Sai Bhargav Kumar, and K. Madhava Krishna. “PIVO: Probabilistic Inverse Velocity Obstacle for Navigation under Uncertainty”. In: *2019 28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*. 2019 28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN). ISSN: 1944-9437. New Delhi: IEEE, Oct. 2019, pp. 1–6. DOI: 10.1109/RO-MAN46459.2019.8956406. URL: https://ieeexplore.ieee.org/abstract/document/8956406 (visited on 12/04/2024).

[16] Lorenzo Bonanni, Daniele Meli, Alberto Castellini, and Alessandro Farinelli. “Monte Carlo planning for mobile robots in large action spaces with velocity obstacles”. In: *The 10th Italian Workshop on Artificial Intelligence and Robotics*. 22nd International Conference of the Italian Association for Artificial Intelligence. 2023, pp. 1–7. URL: https://ceur-ws.org/Vol-3686/short8.pdf.

[17] Tangxiang Fan, Pinxin Long, Wenxin Liu, and Jia Pan. "Distributed multi-robot collision avoidance via deep reinforcement learning for navigation in complex scenarios". In: *The International Journal of Robotics Research* 39.7 (2020), pp. 856–892. DOI: 10.1177/0278364920916531. URL: https://journals.sagepub.com/doi/epdf/10.1177/0278364920916531?src=getftr&utm_source=sciencedirect_contenthosting&getft_integrator=sciencedirect_contenthosting (visited on 12/04/2024).

[18] Martin S. Eby. "A Self-Organizational Approach for Resolving Air Traffic Conflicts". In: *The Lincoln Laboratory Journal* 7 (1994), pp. 239–254. URL: https://archive.ll.mit.edu/publications/journal/pdf/vol07_no2/7.2.6.selforganizing.pdf.

[19] Y. Abe and M. Yoshiki. "Collision avoidance method for multiple autonomous mobile agents by implicit cooperation". In: *Proceedings 2001 IEEE/RSJ*. International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No.01CH37180). Vol. 3. Maui, Hawaii: IEEE, Oct. 2001, pp. 1207–1212. DOI: 10.1109/IROS.2001.977147. URL: https://ieeexplore.ieee.org/abstract/document/977147 (visited on 11/14/2024).

[20] J. M Hoekstra, R. N. H. W van Gent, and R. C. J Ruigrok. "Designing for safety: the free flight air traffic management concept". In: *Reliability Engineering & System Safety* 75.2 (Feb. 1, 2002), pp. 215–232. ISSN: 0951-8320. DOI: 10.1016/S0951-8320(01)00096-5. URL: https://www.sciencedirect.com/science/article/pii/S0951832001000965 (visited on 10/07/2024).

[21] Lee Yang and James Kuchar. "Using intent information in probabilistic conflict analysis". In: *Guidance, Navigation, and Control Conference and Exhibit*. Guidance, Navigation, and Control and Co-located Conferences. American Institute of Aeronautics and Astronautics, Aug. 10, 1998, pp. 797–806. DOI: 10.2514/6.1998-4237. URL: https://arc.aiaa.org/doi/10.2514/6.1998-4237 (visited on 11/20/2024).

[22] Inseok Hwang and Chze Eng Seah. "Intent-Based Probabilistic Conflict Detection for the Next Generation Air Transportation System". In: *Proceedings of the IEEE* 96.12 (Dec. 2008). Conference Name: Proceedings of the IEEE, pp. 2040–2059. ISSN: 1558-2256. DOI: 10.1109/JPROC.2008.2006138. URL: https://ieeexplore.ieee.org/document/4745649 (visited on 11/20/2024).

[23] Marco Porretta, Wolfgang Schuster, Arnab Majumdar, and Washington Ochieng. "Strategic Conflict Detection and Resolution Using Aircraft Intent Information". In: *The Journal of Navigation* 63.1 (Jan. 2010), pp. 61–88. ISSN: 1469-7785, 0373-4633. DOI: 10.1017/S0373463309990270. URL: https://www.cambridge.org/core/journals/journal-of-navigation/article/strategic-conflict-detection-and-resolution-using-aircraft-intent-information/326E85D05619AC5DA182E7EA07F561C7 (visited on 11/20/2024).

[24] Paolo Fiorini and Zvi Shiller. "Motion Planning in Dynamic Environments Using Velocity Obstacles". In: *The International Journal of Robotics Research* 17.7 (July 1998), pp. 760–772. DOI: 10.1177/027836499801700706. URL: https://journals.sagepub.com/doi/epdf/10.1177/027836499801700706 (visited on 11/14/2024).

[25] Karl Bilimoria, Hilda Lee, Zhi-Hong Mao, and Eric Feron. “Comparison of centralized and decentralized conflict resolution strategies for multiple-aircraft problems”. In: *18th Applied Aerodynamics Conference*. AIAA Guidance, Navigation, and Control Conference. Vol. 18. Fluid Dynamics and Co-located Conferences. Denver, Colorado: American Institute of Aeronautics and Astronautics, Aug. 14, 2000, pp. 1–11. DOI: 10.2514/6.2000-4268. URL: https://arc.aiaa.org/doi/10.2514/6.2000-4268 (visited on 11/20/2024).

[26] F. Borrelli, D. Subramanian, A.U. Raghunathan, and L.T. Biegler. “MILP and NLP Techniques for centralized trajectory planning of multiple unmanned air vehicles”. In: *2006 American Control Conference*. 2006 American Control Conference. ISSN: 2378-5861. Minneapolis, Minnesota: IEEE, June 2006, pp. 5763–5768. DOI: 10.1109/ACC.2006.1657644. URL: https://ieeexplore.ieee.org/document/1657644 (visited on 11/20/2024).

[27] Antonio Alonso-Ayuso, Laureano F. Escudero, F. Javier Martín-Campo, and Nenad Mladenovi. “A VNS metaheuristic for solving the aircraft conflict detection and resolution problem by performing turn changes”. In: *Journal of Global Optimization* 63.3 (Nov. 1, 2015), pp. 583–596. ISSN: 1573-2916. DOI: 10.1007/s10898-014-0144-8. URL: https://doi.org/10.1007/s10898-014-0144-8 (visited on 11/20/2024).

[28] Huaxian Liu, Feng Liu, Xuejun Zhang, Xiangmin Guan, Jun Chen, and Pascal Savinaud. “Aircraft conflict resolution method based on hybrid ant colony optimization and artificial potential field”. In: *Science China Information Sciences* 61.12 (Nov. 15, 2018), pp. 1–3. ISSN: 1869-1919. DOI: 10.1007/s11432-017-9310-5. URL: https://doi.org/10.1007/s11432-017-9310-5 (visited on 11/20/2024).

[29] Anoop Sathyan, Nicholas Ernest, Loic Lavigne, Franck Cazaurang, Manish Kumar, and Kelly Cohen. “A Genetic Fuzzy Logic Based Approach to Solving the Aircraft Conflict Resolution Problem”. In: *AIAA Information Systems-AIAA Infotech @ Aerospace*. AIAA SciTech Forum. American Institute of Aeronautics and Astronautics, Jan. 5, 2017. DOI: 10.2514/6.2017-1751. URL: https://arc.aiaa.org/doi/10.2514/6.2017-1751 (visited on 11/20/2024).

[30] Federal Aviation Administration. *14 CFR ğ 91.113 - Right-of-way rules: Except water operations*. LII / Legal Information Institute. 2004. URL: https://www.law.cornell.edu/cfr/text/14/91.113 (visited on 11/20/2024).

[31] Marvin C. Waller and Charles H. Scanlon. *Proceedings of the NASA Workshop on Flight Deck Centered Parallel Runway Approaches in Instrument Meteorological Conditions*. NTRS Author Affiliations: NASA Langley Research Center NTRS Report/Patent Number: NAS 1.55:10191 NTRS Document ID: 19970011096 NTRS Research Center: Langley Research Center (LaRC). Hampton, Virginia: NASA, Dec. 1, 1996. 208 pp. URL: https://ntrs.nasa.gov/citations/19970011096 (visited on 11/20/2024).

[32] Jian Yang, Dong Yin, Yifeng Niu, and Lincheng Shen. “Distributed Cooperative Onboard Planning for the Conflict Resolution of Unmanned Aerial Vehicles”. In: *Journal of Guidance, Control, and Dynamics* 42.2 (Feb. 2019). Publisher: American Institute of Aeronautics and Astronautics, pp. 272–283. ISSN: 0731-5090. DOI: 10.2514/1.G003583. URL: https://arc.aiaa.org/doi/10.2514/1.G003583 (visited on 11/20/2024).

[33] Amy R. Pritchett and Antoine Genton. “Negotiated Decentralized Aircraft Conflict Resolution”. In: *IEEE Transactions on Intelligent Transportation Systems* 19.1 (Jan. 2018). Conference Name: IEEE Transactions on Intelligent Transportation Systems, pp. 81–91. ISSN: 1558-0016. DOI: 10.1109/TITS.2017.2693820. URL: https://ieeexplore.ieee.org/document/7918615 (visited on 11/20/2024).

[34] Jianghai Hu, Maria Prandini, and Shankar Sastry. “Optimal Coordinated Maneuvers for Three-Dimensional Aircraft Conflict Resolution”. In: *Journal of Guidance, Control, and Dynamics* 25.5 (Sept. 2002). Publisher: American Institute of Aeronautics and Astronautics, pp. 888–900. ISSN: 0731-5090. DOI: 10.2514/2.4982. URL: https://arc.aiaa.org/doi/10.2514/2.4982 (visited on 11/20/2024).

[35] L. Pallottino, E.M. Feron, and A. Bicchi. “Conflict resolution problems for air traffic management systems solved with mixed integer programming”. In: *IEEE Transactions on Intelligent Transportation Systems* 3.1 (Mar. 2002). Conference Name: IEEE Transactions on Intelligent Transportation Systems, pp. 3–11. ISSN: 1558-0016. DOI: 10.1109/6979.994791. URL: https://ieeexplore.ieee.org/document/994791/?arnumber=994791 (visited on 11/20/2024).

[36] Georgios Chaloulos, Peter Hokayem, and John Lygeros. “Hierarchical Control with Prioritized MPC for Conflict Resolution in Air Traffic Control”. In: *IFAC Proceedings Volumes* 44.1 (Jan. 2011), pp. 1564–1569. ISSN: 14746670. DOI: 10.3182/20110828-6-IT-1002.01516. URL: https://linkinghub.elsevier.com/retrieve/pii/S1474667016438322 (visited on 11/20/2024).

[37] Gilles Gawinowski, Jean-Louis Garcia, Roger Guerreau, Rosa Weber, and Marc Brochard. “ERASMUS: a new path for 4d trajectory-based enablers to reduce the traffic complexity”. In: *2007 IEEE/AIAA 26th Digital Avionics Systems Conference*. 2007 IEEE/AIAA 26th Digital Avionics Systems Conference. ISSN: 2155-7209. Oct. 2007, 1.A.3–1–1.A.3–11. DOI: 10.1109/DASC.2007.4391819. URL: https://ieeexplore.ieee.org/document/4391819 (visited on 11/20/2024).

[38] Georgios Chaloulos, Eva Crück, and John Lygeros. “A simulation based study of subliminal control for air traffic management”. In: *Transportation Research Part C: Emerging Technologies*. Special issue on Transportation Simulation 18.6 (Dec. 1, 2010), pp. 963–974. ISSN: 0968-090X. DOI: 10.1016/j.trc.2010.03.002. URL: https://www.sciencedirect.com/science/article/pii/S0968090X10000288 (visited on 11/20/2024).

[39] David Rey, Christophe Rapine, Rémy Fondacci, and Nour-Eddin El Faouzi. “Subliminal Speed Control in Air Traffic Management: Optimization and Simulation”. In: *Transportation Science* 50.1 (Feb. 2016). Publisher: INFORMS, pp. 240–262. ISSN: 0041-1655. DOI: 10.1287/trsc.2015.0602. URL: https://pubsonline.informs.org/doi/10.1287/trsc.2015.0602 (visited on 11/20/2024).

[40] Laurence H Mutuel, Pierre Neri, and Erwan Paricaud. “Initial 4D Trajectory Management Concept Evaluation”. In: *USA/Europe Air Traffic Management Research and Development Seminar*. Air Traffic Management Research and Development Seminar. Vol. 1. Chicago, Illinois: The European Organisation for the Safety of Air Navigation (EUROCONTROL), 2013, pp. 1–8. ISBN: 978-1-5108-0173-8. URL: https://

citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=376adf0766be87bdf9e5311881d1a925ae46cb3c.

[41] Hanna Romanivna Plytus. "The ground proximity warning system of modern aircraft". Accepted: 2023-03-09T08:15:32Z Publisher: National Aviation University. Master. Kyiv: National Aviation University, Mar. 7, 2023. 96 pp. URL: https://er.nau.edu.ua/handle/NAU/58304 (visited on 11/22/2024).

[42] Petr Cásek and Eva Gelnarová. *iFly D9.1: Operational Services and Environment Description (OSED) of Airborne Self-Separation Procedure (SSEP)*. D9.1. Jan. 29, 2010, pp. 1–46. URL: https://ifly.nlr.nl/documents/D9.1_Final_1.2_29_Jan_2010.pdf.

[43] Ítalo Romani de Oliveira and Petr Cásek. *iFly D9.4: Airborne System Design Requirements of Airborne Self-Separation Procedure*. D9.4. June 30, 2011, pp. 1–60. URL: https://ifly.nlr.nl/documents/D9.4_Final_1.4_30_Jun_2011.pdf.

[44] H A P Blom, G J Bakker, P J G Blanker, J Daams, M H C Everdij, and M B Klompstra. "Accident Risk Assessment for Advanced Air Traffic Management". In: *Air Transportation Systems Engineering* (2001), pp. 1–27. URL: https://reports.nlr.nl/server/api/core/bitstreams/d85aca8c-c8ab-4037-838b-6350b8306f58/content.

[45] Henk Blom, Sybert Stroeve, and Hans de Jong. "Safety Risk Assessment by Monte Carlo Simulation of Complex Safety Critical Operations". In: *Proceedings of the 14th Safety-Critical Systems Symposium, SSS 2006*. Developments in Risk-Based Approaches to Safety -. Journal Abbreviation: Developments in Risk-Based Approaches to Safety - Proceedings of the 14th Safety-Critical Systems Symposium, SSS 2006. Bristol: NLR, Feb. 1, 2006, pp. 47–67. ISBN: 978-1-84628-333-8. DOI: 10.1007/1-84628-447-3_3. URL: https://reports.nlr.nl/server/api/core/bitstreams/27c8f5e7-ddf6-41de-94a3-20227e8ab699/content.

[46] Henk A.P. Blom, G.J. (Bert) Bakker, and Jaroslav Krystul. "Probabilistic reachability analysis for large scale stochastic hybrid systems". In: *2007 46th IEEE Conference on Decision and Control*. 2007 46th IEEE Conference on Decision and Control. New Orleans, LA, USA: IEEE, 2007, pp. 3182–3189. ISBN: 978-1-4244-1497-0. DOI: 10.1109/CDC.2007.4434095. URL: http://ieeexplore.ieee.org/document/4434095/ (visited on 12/05/2024).

[47] H A P Blom, G J Bakker, and J Krystul. *iFly D7.2e Rare event estimation for a large scale stochastic hybrid system with air traffic application -IPS extension to large hybrid systems-*. D7.2e. Jan. 30, 2009, pp. 1–129. URL: https://ifly.nlr.nl/documents/D7.2e_Final_0.6_28_Jan_2009.pdf.

[48] Henk A P Blom, Jaroslav Krystul, Bert Bakker, Margriet Klompstra, and Bart Klein Obbink. "Free Flight Collision Risk Estimation by Sequential Monte Carlo Simulation". In: *Stochastic Hybrid Systems*. Taylor & Francis / CPC Press, Jan. 2007, pp. 247–279. URL: https://www.researchgate.net/publication/255970355_Free_flight_collision_risk_estimation_by_sequential_Monte_Carlo_simulation.

[49] M H C Everdij and H A P Blom. *Petri-nets and hybrid-state Markov processes in a power-hierarchy of dependability models*. NLR-TP-2003-559. NLR, Oct. 2003, pp. 1–19. URL: https://reports.nlr.nl/server/api/core/bitstreams/15873cc0-f477-4747-9d05-fe09e997a9cf/content.

[50] H A P Blom and G J Bakker. *iFly D7.4 Final Report on Accident Risk Assessment of the A3 operation*. D7.4. Sept. 30, 2011, pp. 1–84. URL: https://ifly.nlr.nl/documents/D7.4_Final_0.7_19_Sep_2011.pdf.

[51] Gustavo Mercado-Velasco, Max Mulder, and Marinus Van Paassen. "Analysis of Air Traffic Controller Workload Reduction Based on the Solution Space for the Merging Task". In: *AIAA Guidance, Navigation, and Control Conference*. Guidance, Navigation, and Control and Co-located Conferences. American Institute of Aeronautics and Astronautics, Aug. 2, 2010, pp. 1–18. DOI: 10.2514/6.2010-7541. URL: https://arc.aiaa.org/doi/10.2514/6.2010-7541 (visited on 12/20/2024).

[52] Suthes Balasooriyan. "Multi-aircraft Conflict Resolution using Velocity Obstacles". Master. Delft: TU Delft, Oct. 2017. 126 pp. URL: https://repository.tudelft.nl/record/uuid:cf361aff-a7ec-444c-940a-d711e076d108 (visited on 11/28/2024).

[53] Gustavo Adrián Mercado Velasco, Clark Borst, Joost Ellerbroek, M. M. van Paassen, and Max Mulder. "The Use of Intent Information in Conflict Detection and Resolution Models Based on Dynamic Velocity Obstacles". In: *IEEE Transactions on Intelligent Transportation Systems* 16.4 (Aug. 2015). Conference Name: IEEE Transactions on Intelligent Transportation Systems, pp. 2297–2302. ISSN: 1558-0016. DOI: 10.1109/TITS.2014.2376031. URL: https://ieeexplore.ieee.org/document/7063942 (visited on 10/07/2024).

[54] P. Hermes, M. Mulder, M. M. Van Paassen, J. H. L. Boering, and H. Huisman. "Solution-Space-Based Complexity Analysis of the Difficulty of Aircraft Merging Tasks". In: *Journal of Aircraft* 46.6 (Nov. 2009). Publisher: American Institute of Aeronautics and Astronautics (AIAA), pp. 1995–2015. ISSN: 0021-8669, 1533-3868. DOI: 10.2514/1.42886. URL: https://arc.aiaa.org/doi/10.2514/1.42886 (visited on 07/14/2025).

[55] G.J. Bakker, C.I.J. Nieskens, and H.A.P. Blom. *Accident Risk Assessment Model for the A3G CONOPS Considered within iFLY*. Technical Report NLR-TR-2013-xxx. Version 1.0, NLR Company Confidential. National Aerospace Laboratory (NLR), 2010.

[56] Jacco M Hoekstra and Joost Ellerbroek. "BlueSky ATC Simulator Project: an Open Data and Open Source Approach". In: *International Conference on Research in Air Transportation* 7 (2016), pp. 1–9. URL: https://pure.tudelft.nl/ws/files/10083831/Hoekstra_BlueSky_project.pdf.

[57] Ding-Jen Liu and William Chung. "ASTOR: An Avionics Concept Test Bed in a Distributed Networked Synthetic Airspace Environment". In: *AIAA Modeling and Simulation Technologies Conference and Exhibit*. Guidance, Navigation, and Control and Co-located Conferences. Providence, Rhode Island: American Institute of Aeronautics and Astronautics, Aug. 16, 2004. DOI: 10.2514/6.2004-5259. URL: https://arc.aiaa.org/doi/10.2514/6.2004-5259 (visited on 11/14/2024).

[58] Emmanuel Sunil, Joost Ellerbroek, Jacco Hoekstra, Andrija Vidosavljevic, Michael Arntzen, Frank Bussink, and Dennis Nieuwenhuisen. "Analysis of Airspace Structure and Capacity for Decentralized Separation Using Fast-Time Simulations". In: *Journal of Guidance, Control, and Dynamics* 40.1 (Jan. 2017), pp. 38–51. ISSN: 0731-5090, 1533-3884. DOI: 10.2514/1.G000528. URL: https://arc.aiaa.org/doi/10.2514/1.G000528 (visited on 11/07/2024).