Conflict Detection and Resolution Considering Human–Machine and Air–Ground Interaction

Yutong Chen, Yumeng Ren, Zhi Jun Lim, Sameer Alam*
Air Traffic Management Research Institute, Nanyang Technological University, Singapore
{yutong.chen, yumeng.ren, zhijun.lim, sameeralam}@ntu.edu.sg

***Abstract*—With the continual growth of air traffic, controller workload has become a key bottleneck, particularly in high-density airspace where multi-aircraft conflict resolution strains cognitive and computational capacity. While optimisation- and AI-based methods have been proposed, they often assume advisories are executed instantly, overlooking delays and uncertainties in human–machine and air–ground interactions, which limits their practical applicability. Building on the SESAR HYPERSOLVER project and supporting its vision of advancing human–AI teaming in air traffic management (ATM), this paper proposes a conflict detection and resolution decision-support method that explicitly incorporates controller workload and the dynamics of human–machine and air–ground interactions. This work introduces three key innovations: a trajectory prediction method that captures uncertainty caused by human–machine and air–ground interactions, thereby enabling more realistic conflict detection under dynamic conditions; a robustness index integrated into conflict-free trajectory planning optimisation, allowing the generation of solutions that better withstand operational uncertainties while ensuring safety and efficiency; and an adaptive dynamic search algorithm alternating between global and local strategies, effectively balancing optimisation performance and computational efficiency. Fast-time simulations demonstrate that the proposed approach has the potential to effectively handle uncertainties in human–machine and air–ground interactions in real-time operations, while accommodating traffic densities beyond the current maximum observed in practice, thereby paving the way for more effective deployment in real-world ATM systems.**

***Keywords*—Air traffic management, conflict detection and resolution, controller workload, human–machine interaction, air-ground interaction, trajectory optimisation**

I. Introduction

With the continual growth of air traffic, controller workload has emerged as a critical bottleneck to enhancing airspace capacity, particularly in high-density airspace where multi-aircraft conflict resolution poses significant challenges to controllers' cognitive and computational capabilities [1]. Excessive workload not only increases the likelihood of errors but also constrains the scalability of air traffic management (ATM) systems in the face of rising demand [2].

To address this bottleneck, numerous conflict detection and resolution methods have been developed in recent years, leveraging advances in both optimisation and artificial intelligence (AI). Optimisation-based approaches, such as mixed-integer programming, heuristics, and hybrid search strategies, have demonstrated effectiveness in improving throughput and reducing conflict risk in dense traffic scenarios [3] [4] [5]. In parallel, AI-driven approaches are gaining momentum: deep learning-based trajectory prediction methods have improved accuracy in capturing aircraft intent under uncertainty, while reinforcement learning has shown promise in generating adaptive and scalable conflict resolution policies for multi-agent environments [6] [7] [8]. These advances provide valuable technological foundations for future intelligent ATM systems.

However, a major limitation of most existing methods is the assumption that recommended conflict resolution advisories can be executed immediately and accurately. This overlooks the fact that in real-world operations, human–machine interactions and air–ground communications inherently require time and are subject to uncertainty [9]. In practice, controllers must review, decide, and communicate instructions sequentially, while flight crews need time to interpret and execute them. These processes constitute essential components of controller workload and cannot be neglected in system design [10]. Ignoring these interactions creates a gap between algorithmic feasibility and operational applicability, restricting the deployment of many promising methods in current ATM systems.

This paper aims to bridge this gap by introducing a conflict detection and resolution decision-support framework that explicitly incorporates human–machine and air–ground interaction dynamics. The main contributions of this paper are summarised as follows:

- A trajectory prediction method is presented that incorporates the uncertainty of times in air–ground interactions, enabling more realistic conflict detection in dynamic operational scenarios.
- A robustness index model is developed and integrated into the conflict-free trajectory planning optimisation framework, providing solutions that can better withstand operational uncertainties while maintaining safety and efficiency.
- An adaptive dynamic search algorithm is introduced that flexibly alternates between global and local search depending on search phases, effectively balancing optimisation performance and computational efficiency.

II. Problem Description

When controllers use conflict resolution decision-support systems, if the underlying algorithms fail to account for controller workload, the generated solutions may not resolve conflicts effectively. This is because there is a necessary

process between the decision-support system generating a recommended solution and the flight starting to manoeuvre. As shown in Figure 1, after receiving the rerouting advisory generated by the decision-support system, the controller must sequentially complete the human–machine interaction (controller reviews and decides the solution for the flight) and the air–ground interaction (controller communicates with the flight's crew). Only thereafter can the flight start to perform the manoeuvre. By that time, however, the aircraft may no longer be able to follow the recommended trajectory, potentially leading to failure of conflict resolution. This problem can be attributed to the following factors:

- Controllers require decision-making time to review the recommended trajectory for safety, as human operators remain the final safeguard in today's ATM systems (i.e., human-machine interaction).
- Controllers require communication time to issue radar vectoring instructions to the crew, whether via voice or datalink (i.e., air-ground interaction).
- Uncertainty exists in the time spans of both human–machine and air–ground interactions.
- Controllers can only perform one task at a time—they cannot simultaneously make multiple decisions, or decide and communicate concurrently, nor can they communicate with two crews simultaneously.

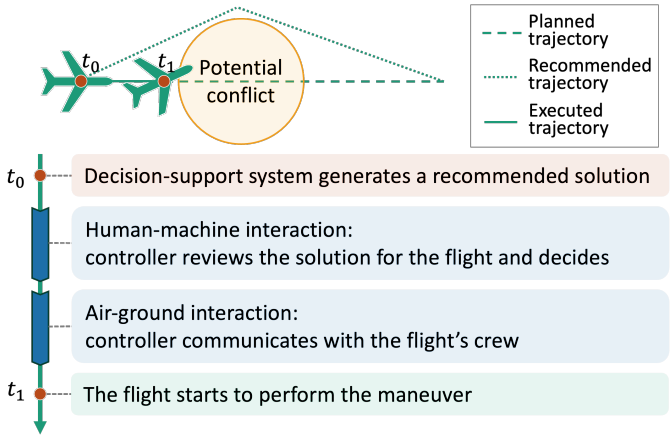


Figure 1. Why conflict resolution recommendations without considering controller workload may fail in practice?

Therefore, this paper aims to address the limitations of current conflict resolution decision-support systems by extending the conventional conflict detection and resolution framework to explicitly incorporate controller workload, which includes both human–machine and air–ground interactions, as well as the associated uncertainties, thereby enabling more practical and effective applications in real-world operations.

III. Methodology

To address the aforementioned problem, we propose a conflict detection and resolution method that explicitly accounts for controller workload, including human–machine and air–ground interactions. The framework consists of four interrelated modules (as illustrated in Figure 2):

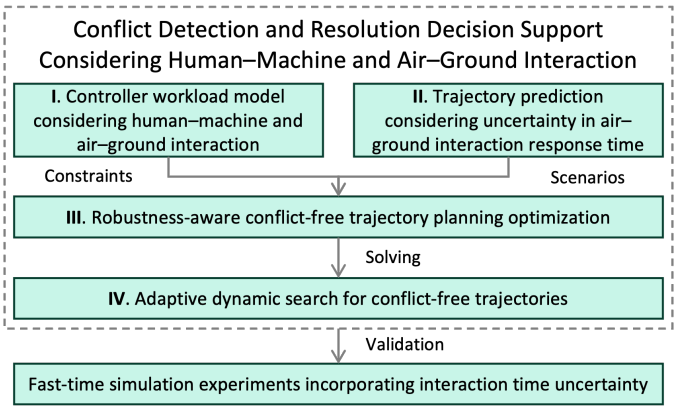


Figure 2. Research framework

I. Controller workload model considering human–machine and air–ground interaction,
II. Trajectory prediction considering uncertainty in air–ground interaction response time,
III. Robustness-aware conflict-free trajectory planning optimisation, and
IV. Adaptive dynamic search for conflict-free trajectories.

Modules I and II provide operational constraints and scenario inputs to module III, while module IV serves as the solver to obtain recommended solutions. The proposed method will be validated through fast-time simulation experiments incorporating interaction time uncertainty.

A. Controller workload model considering human–machine and air–ground interaction

As a preliminary study, this paper uses the most commonly employed radar vectoring manoeuvre model in practice as an example (refer to Figure 3). The model defines three key points: the current point $(\boldsymbol{p}^{\mathrm{C}}, t^{\mathrm{C}})$, the recommended 1st rerouting point (RP1) $(\boldsymbol{p}_1^{\mathrm{RR}}, t_1^{\mathrm{RR}})$, and the recommended 2nd rerouting point (RP2) $(\boldsymbol{p}_2^{\mathrm{RR}}, t_2^{\mathrm{RR}})$. Each symbol consists of two elements, representing the spatial coordinates of the point and the corresponding time.

This paper makes a simplified assumption based on practical operations: The change in motion state is instantaneous. For recommended RP1, the controller issues a heading instruction to the crew, which directs the aircraft to turn either left or right by an angle α from its current heading. For recommended RP2, the controller issues an instruction for the aircraft to fly toward a designated waypoint, which is typically the exit point of the current sector, where the yaw angle is β. To ensure that the controller has sufficient time to review and issue the trajectory recommended by the decision-support system, a reserved decision-making time is introduced, along with a reserved communication time assigned to recommended RP1 and RP2, respectively.

Therefore, for flight i, the times of the three key points are denoted as t_i^{G}, $t_{i,1}^{\mathrm{RR}}$, and $t_{i,2}^{\mathrm{RR}}$, respectively. The duration

of the reserved decision-making time is denoted by T_i^{D}, with its start and end times represented by t_i^{DS} and t_i^{DE}, that is, $T_i^{\mathrm{D}} = t_i^{\mathrm{DE}} - t_i^{\mathrm{DS}}$. The durations of the reserved communication times are denoted by $T_{i,1}^{\mathrm{C}}$ and $T_{i,2}^{\mathrm{C}}$, with $t_{i,1}^{\mathrm{CS}}$, $t_{i,1}^{\mathrm{CE}}$ and $t_{i,2}^{\mathrm{CS}}$, $t_{i,2}^{\mathrm{CE}}$ as their respective start and end times, that is, $T_{i,1}^{\mathrm{C}} = t_{i,1}^{\mathrm{CE}} - t_{i,1}^{\mathrm{CS}}$ and $T_{i,2}^{\mathrm{C}} = t_{i,2}^{\mathrm{CE}} - t_{i,2}^{\mathrm{CS}}$.

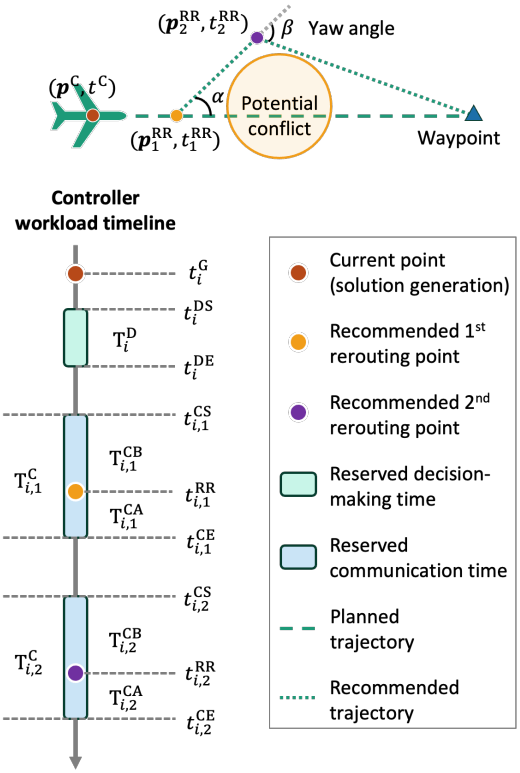


Figure 3. Controller workload timeline.

In practice, on the one hand, the controller needs to communicate with the crew some time before the flight passes the recommended rerouting point (i.e., $t_{i,1}^{\mathrm{CS}} < t_{i,1}^{\mathrm{RR}}$ and $t_{i,2}^{\mathrm{CS}} < t_{i,2}^{\mathrm{RR}}$), so that the aircraft can manoeuvre at a time as close as possible to the recommended point. On the other hand, it cannot be guaranteed that the aircraft will manoeuvre exactly at the recommended time; it may occur earlier or later. For conservatism, the end of the reserved communication time is set after the recommended rerouting point (i.e., $t_{i,1}^{\mathrm{CE}} > t_{i,1}^{\mathrm{RR}}$ and $t_{i,2}^{\mathrm{CE}} > t_{i,2}^{\mathrm{RR}}$). Therefore, each reserved communication time is divided into two parts: one before the rerouting point and one after the rerouting point. The lengths of these two parts are denoted as $T_{i,1}^{\mathrm{CB}}$ and $T_{i,1}^{\mathrm{CA}}$ for the RRP1 (i.e., $T_{i,1}^{\mathrm{CB}} = t_{i,1}^{\mathrm{RR}} - t_{i,1}^{\mathrm{CS}}$ and $T_{i,1}^{\mathrm{CA}} = t_{i,1}^{\mathrm{CE}} - t_{i,1}^{\mathrm{RR}}$), and $T_{i,2}^{\mathrm{CB}}$ and $T_{i,2}^{\mathrm{CA}}$ for the RRP2 (i.e., $T_{i,2}^{\mathrm{CB}} = t_{i,2}^{\mathrm{RR}} - t_{i,2}^{\mathrm{CS}}$ and $T_{i,2}^{\mathrm{CA}} = t_{i,2}^{\mathrm{CE}} - t_{i,2}^{\mathrm{RR}}$).

The values of parameters T_i^{D}, $T_{i,1}^{\mathrm{C}}$, $T_{i,2}^{\mathrm{C}}$, $T_{i,1}^{\mathrm{CB}}$, $T_{i,1}^{\mathrm{CA}}$, $T_{i,2}^{\mathrm{CB}}$ and $T_{i,2}^{\mathrm{CA}}$ can be determined adaptively, depending on the specific requirements of real-world operations.

B. Trajectory prediction considering uncertainty in air–ground interaction response time

Conflict detection relies on real-time trajectory prediction. Since the uncertainty in air–ground interaction response time is taken into account, once a flight receives a newly recommended rerouting trajectory, its trajectory prediction becomes more complex compared with the direct-flight case. This is because uncertainties may occur at both recommended RP1 and RP2, i.e., in reality, it cannot be guaranteed that the flight will manoeuvre exactly at the scheduled times of recommended RP1 and RP2.

In the proposed trajectory prediction method, when a flight follows a recommended rerouting trajectory, its predicted trajectory is divided into ten possible cases, corresponding to sub-figures (a)–(j) in Figure 4. The prediction is based on the following **assumptions**:

1. The flight speed remains constant.
2. The yaw angle for the manoeuvre at recommended RP1 is always α.
3. The time of predicted RP2 is consistent with the time of recommended RP2.
4. The manoeuvre heading for RP2 always points toward the designated waypoint.
5. If a flight passes a rerouting point without performing the corresponding manoeuvre, it is assumed to manoeuvre immediately thereafter.

The **ten prediction cases** are described as follows:

(a) If the flight has not reached the recommended RP1 and has not completed the manoeuvre, the predicted trajectory is identical to the recommended trajectory.

(b) If the flight performs the manoeuvre before reaching recommended RP1, and before reaching RP2, has not yet completed the second manoeuvre, then based on assumptions 2 and 3, a predicted RP2 is generated, with $\angle \boldsymbol{p}_2^{\mathrm{PR}} \boldsymbol{p}_1^{\mathrm{AR}} \boldsymbol{p}^{\mathrm{W}} = \alpha$. Further from assumption 1, $|\boldsymbol{p}_1^{\mathrm{AR}} \boldsymbol{p}_2^{\mathrm{PR}}| = |\boldsymbol{p}_1^{\mathrm{AR}} \boldsymbol{p}_1^{\mathrm{RR}}| + |\boldsymbol{p}_1^{\mathrm{RR}} \boldsymbol{p}_2^{\mathrm{RR}}|$, where $|\cdot\cdot|$ denotes the length of the line segment connecting the two points. The predicted flight trajectory passes through the predicted RRP2 toward the waypoint.

(c) Based on case (b), if the flight completes the manoeuvre before reaching the predicted RP2, then, according to assumption 4, the predicted trajectory proceeds directly to the waypoint.

(d) Based on case (b), if the flight passes the predicted RP2 without manoeuvring, then by assumption 5, the predicted trajectory proceeds immediately toward the waypoint.

(e) Based on case (d), if the flight subsequently manoeuvres for the RP2, then according to assumption 4, the predicted trajectory proceeds directly to the waypoint.

(f) If the flight passes the recommended RP1 without manoeuvring, then by assumption 5, a predicted RP2 is generated. From assumptions 1-3, $\angle \boldsymbol{p}_2^{\mathrm{PR}} \boldsymbol{p}^{\mathrm{C}} \boldsymbol{p}^{\mathrm{W}} = \alpha$, $|\boldsymbol{p}^{\mathrm{C}} \boldsymbol{p}_2^{\mathrm{PR}}| = |\boldsymbol{p}_1^{\mathrm{RR}} \boldsymbol{p}_2^{\mathrm{RR}}| - |\boldsymbol{p}_1^{\mathrm{RR}} \boldsymbol{p}^{\mathrm{C}}|$. The predicted trajectory immediately proceeds toward the predicted RP2, passes

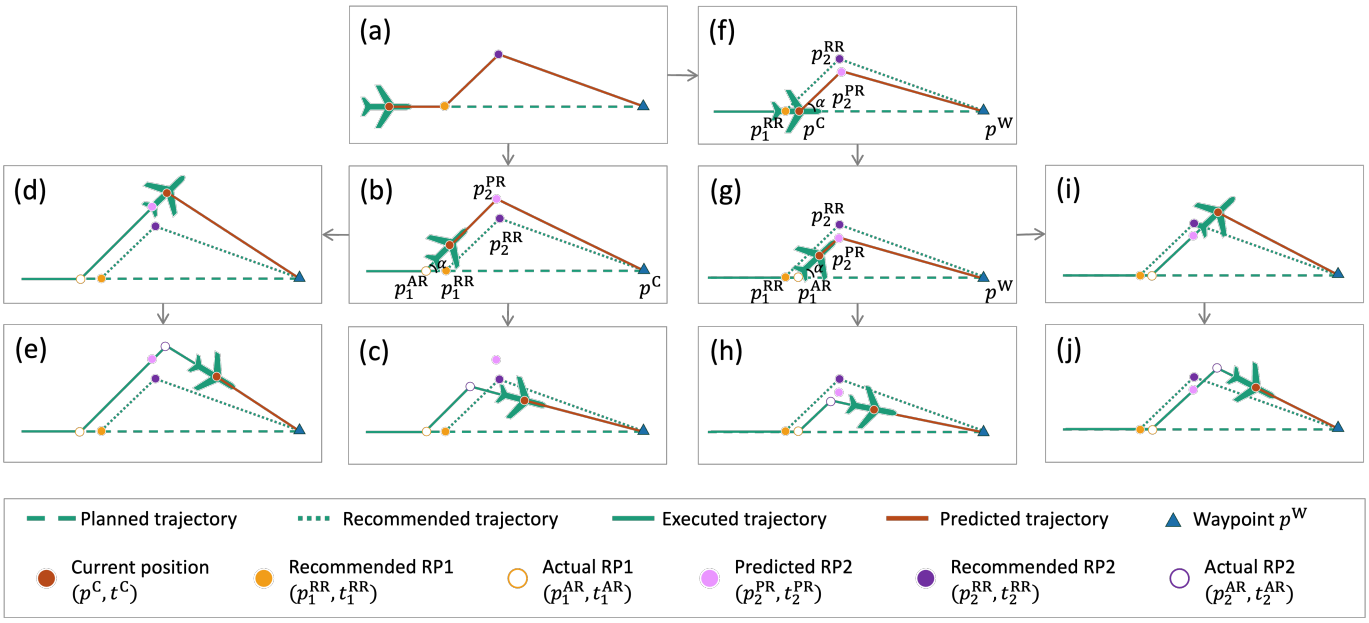


Figure 4. Trajectory prediction considering the uncertainty of air–ground interaction response time.

through it, and subsequently continues toward the waypoint.

(g) Based on case (f), if the flight subsequently manoeuvres for the RP1, but has not reached the recommended RP2 and has not completed the manoeuvre, based on assumptions 2 and 3, a predicted RP2 is generated, with $\angle \boldsymbol{p}_2^{\mathrm{PR}} \boldsymbol{p}_1^{\mathrm{AR}} \boldsymbol{p}^{\mathrm{W}} = \alpha$. Further from assumption 1, $|\boldsymbol{p}_1^{\mathrm{AR}} \boldsymbol{p}_2^{\mathrm{PR}}| = |\boldsymbol{p}_1^{\mathrm{RR}} \boldsymbol{p}_2^{\mathrm{RR}}| - |\boldsymbol{p}_1^{\mathrm{RR}} \boldsymbol{p}_1^{\mathrm{AR}}|$. The predicted trajectory passes through the predicted RP2 toward the waypoint.

(h) Based on case (g), if the flight manoeuvres before reaching the predicted RP2, then by assumption 4, the predicted trajectory proceeds directly to the waypoint.

(i) Based on case (g), if the flight passes the predicted RP2 without manoeuvring, then by assumption 5, the predicted trajectory proceeds immediately toward the waypoint.

(j) Based on case (i), and according to assumption 4, the predicted trajectory proceeds directly to the waypoint.

C. Robustness-aware conflict-free trajectory planning optimisation

In this paper, a conflict-free trajectory planning optimisation model incorporating robustness is developed to support the generation of conflict resolution advisories that can effectively cope with operational uncertainties. The constraints mainly include controller workload constraints, safety separation constraints, and aircraft performance constraints, while the optimisation objectives primarily consider flight cost and robustness. The decision variables are the start of the reserved decision-making time, and the locations of the recommended RP1 and RP2, that is, t_i^{DS}, $\boldsymbol{p}_{i,1}^{\mathrm{RR}}$ and $\boldsymbol{p}_{i,2}^{\mathrm{RR}}$ for flight i. As introduced in subsection III-A, since this paper assumes a constant flight speed, $t_{i,1}^{\mathrm{RR}}$ and $t_{i,2}^{\mathrm{RR}}$ can be computed from $\boldsymbol{p}_{i,1}^{\mathrm{RR}}$ and $\boldsymbol{p}_{i,2}^{\mathrm{RR}}$; subsequently, $t_{i,1}^{\mathrm{CS}}$, $t_{i,1}^{\mathrm{CE}}$, $t_{i,2}^{\mathrm{CS}}$, and $t_{i,2}^{\mathrm{CE}}$ can be obtained, and t_i^{DE} can also be derived from t_i^{DS}. The details are introduced as follows.

With respect to controller workload, the three reserved time intervals (the reserved decision-making time and the two reserved communication times) must follow a sequential order without overlap. Moreover, these reserved intervals must not overlap with the workload periods already assigned to the controller. Therefore, the **controller workload constraint** is expressed as:

$$t^{\mathrm{C}} \le t_i^{\mathrm{DS}} \tag{1}$$

$$t_i^{\mathrm{DE}} \le t_{i,1}^{\mathrm{CS}} \tag{2}$$

$$t_{i,1}^{\mathrm{CE}} \le t_{i,2}^{\mathrm{CS}} \tag{3}$$

$$([t_i^{\mathrm{DS}}, t_i^{\mathrm{DE}}] \cup [t_{i,1}^{\mathrm{CS}}, t_{i,1}^{\mathrm{CE}}] \cup [t_{i,2}^{\mathrm{CS}}, t_{i,2}^{\mathrm{CE}}]) \cap \boldsymbol{W} = \varnothing \tag{4}$$

where $\boldsymbol{W}$ denotes the set of workload periods already assigned to the controller.

Conflict management follows the first-come-first-served principle in this study, meaning that each flight only avoids those flights that enter the sector earlier than itself [5]. The **safety separation constraint** for flight i can be expressed as:

$$\begin{gathered} |\boldsymbol{p}_i(t) \boldsymbol{p}_j(t)| \le D^{\mathrm{SS}}, \\ \boldsymbol{p}_i(t) \in \boldsymbol{T}_i^{\mathrm{C}}(\boldsymbol{p}_{i,1}^{\mathrm{RR}}, \boldsymbol{p}_{i,2}^{\mathrm{RR}}), \boldsymbol{p}_j(t) \in \boldsymbol{T}_j^{\mathrm{P}}, j \in J_i \end{gathered} \tag{5}$$

where $\boldsymbol{T}_i^{\mathrm{C}}(\boldsymbol{p}_{i,1}^{\mathrm{RR}}, \boldsymbol{p}_{i,2}^{\mathrm{RR}})$ denotes the candidate trajectory of flight i determined by the recommended rerouting points $\boldsymbol{p}_{i,1}^{\mathrm{RR}}$ and $\boldsymbol{p}_{i,2}^{\mathrm{RR}}$, and $\boldsymbol{p}_i(t)$ represents the position on this trajectory at time t. $\boldsymbol{T}_j^{\mathrm{P}}$ denotes the predicted trajectory of flight j, and $\boldsymbol{p}_j(t)$ represents the position on this trajectory at time t. D^{SS} denotes the minimum required safety separation distance. J_i denotes the set of flights that have entered the sector of interest prior to flight i and remain within the sector.

In the rerouting model designed in this paper (refer to subsection III-A), two heading change manoeuvres are considered, corresponding to RP1 and RP2, with yaw angles denoted by α and β, respectively. Since this study assumes instantaneous changes in motion states, it is necessary to constrain the values of α and β to approximate real-world conditions. Moreover, imposing such a constraint on them also helps to limit the deviation of the rerouted trajectory from the planned trajectory. Given the assumption of constant speed, the **aircraft performance constraint** is therefore considered only in terms of yaw angle limitations, which can be expressed as:

$$\beta \leq \theta_{\max} \tag{6}$$

where $\theta_{\max}$ represents the upper limit of the yaw angle, determined based on practical operational needs. It is evident that α is always smaller than β; therefore, it is sufficient to impose an upper bound only on β.

In this paper, the optimisation objective is defined as the weighted minimisation of delay time cost and the robustness index. Since the flight speed is assumed constant in this study, delay time can be regarded as proportional to additional flight distance and extra fuel consumption, which are common measures of flight cost. A larger value of the robustness index indicates a stronger capability of coping with uncertainties, and its design methodology will be introduced in subsection III-D. The **objective function** is expressed as:

$$\min Z = t_i^{\mathrm{D}}(\boldsymbol{p}_{i,1}^{\mathrm{RR}}, \boldsymbol{p}_{i,2}^{\mathrm{RR}}) - \omega r_i(\boldsymbol{p}_{i,1}^{\mathrm{RR}}, \boldsymbol{p}_{i,2}^{\mathrm{RR}}) \tag{7}$$

where $t_i^{\mathrm{D}}(\boldsymbol{p}_{i,1}^{\mathrm{RR}}, \boldsymbol{p}_{i,2}^{\mathrm{RR}})$ denotes the delay time of flight i under the given decision variables, $r_i(\boldsymbol{p}_{i,1}^{\mathrm{RR}}, \boldsymbol{p}_{i,2}^{\mathrm{RR}})$ denotes the robustness index of flight i under the given decision variables, and ω is the weighting coefficient used to balance the two objectives.

D. Adaptive dynamic search for conflict-free trajectories

Since dynamic responsiveness to traffic state changes is required, the proposed decision-support method must update its recommendations dynamically, i.e., every T^{U} seconds (typically a few seconds). Therefore, the solution algorithm is required to be computationally efficient. Considering the problem to be solved is a non-linear optimisation problem, this paper proposes a grid-based heuristic adaptive dynamic search for conflict-free trajectories. The grid model is adopted to transform the continuous solution space into a discrete one, thereby limiting the number of search operations to a finite set. The heuristic-based dynamic search is employed to further balance optimisation performance and computational efficiency. In each update, flights are processed sequentially according to their entry order into the sector. For each flight, conflict detection is performed, and if a conflict is identified, an attempt is made to generate a recommended conflict-free trajectory. The detailed procedure is as follows.

First, a candidate set of RP1 is constructed through the following steps:

1) On line segment $\boldsymbol{p}^{\mathrm{C}}\boldsymbol{p}^{\mathrm{W}}$, between the current time and the closest point of approach (CPA) time, select the earliest feasible RP1 that satisfies the controller workload constraint for RP1, denoted as $\boldsymbol{p}_1$. Then, select the latest feasible RP1 that satisfies the controller workload constraints for both RP1 and RP2 (i.e., this RP1 ensures that the time interval to the CPA is at least $T_{i,1}^{\mathrm{CA}} + T_{i,2}^{\mathrm{C}}$), denoted as $\boldsymbol{p}_2$.
2) Using $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ as the start and end points, respectively, identify all feasible RP1 points on line segment $\boldsymbol{p}_1\boldsymbol{p}_2$ at intervals of D^{P}.
3) These feasible RP1 points constitute the candidate set of RP1, denoted as $\boldsymbol{P}^{\mathrm{RP1}}$.

Then, based on a feasible RP1, the optimal RP2 is determined, resulting in a solution pair (RP1, RP2). Considering that the proposed decision-support method dynamically updates the solution, one way to avoid repeated exhaustive searches in each update is as follows: if a recommended solution for a given flight has already been generated in the previous update, the current search primarily performs local checking and adjustment rather than a full global search. Based on this idea, in each update, the search process for RP2 is carried out as follows (refer to Figure 5):

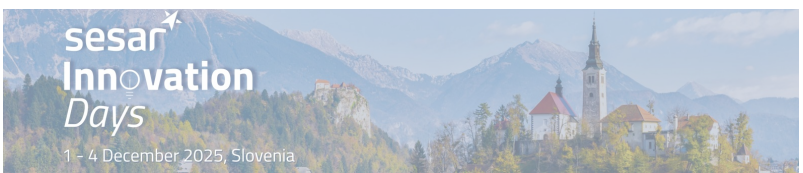
(1) **Initial search**. If this is the first search for the flight, the initial search begins with the first candidate RP1 in $\boldsymbol{P}^{\mathrm{RP1}}$. In this search, a grid coordinate system is constructed with RP1 as the origin, the direction from RP1 to the waypoint as the X-axis, and the perpendicular direction as the Y-axis. The grid has a length L^{G}, and its centre points are taken as candidate RP2s. Based on planar geometry, RP2s satisfying the yaw-angle constraint must lie within the intersection of two symmetric circles of radius $r = |\boldsymbol{p}_{\boldsymbol{1}}^{\mathrm{RR}}\boldsymbol{p}^{\mathrm{W}}|/(2\sin\theta_{\max})$ passing through RP1 and the waypoint. This defines the initial feasible grid.

(2) **Feasible solution check**. If the initial search yields feasible solutions that satisfy the safety separation constraints, a robustness index value is assigned to each feasible solution. For candidate RP2 with the coordination of (X, Y), the robustness index is defined as Equation 8, where $\boldsymbol{N}$ denotes the set of natural numbers and $\boldsymbol{G}^{\mathrm{IF}}$ denotes the set of infeasible grids. The optimal solution is then selected according to the optimisation objective to generate the recommended solution.

$$\begin{aligned} &r_i(X, Y) = \\ &\arg\max_{r \in \boldsymbol{N}} \{\nexists (X \pm x, Y \pm y) \in \boldsymbol{G}^{\mathrm{IF}}, x, y \leq r\} \end{aligned} \tag{8}$$

(3) **Exhaustion of RP1 candidates**. If no feasible solution is found in the initial search, the next candidate RP1 is examined, repeating steps (1) and (2). If no feasible RP2 can be found for any RP1 candidate, the output is reported as infeasible.

(4) **Subsequent search**. If the current search is not the first search, a subsequent search is performed. Taking the solution (RP1, RP2) from the previous update as the reference, the subsequent search only examines the grid points within the R^{S}-neighbourhood of RP2 (i.e., $\{(x, y) | x = X \pm \Delta x, y = Y \pm \Delta y, \Delta x, \Delta y \leq R^{\mathrm{S}}\}$). The



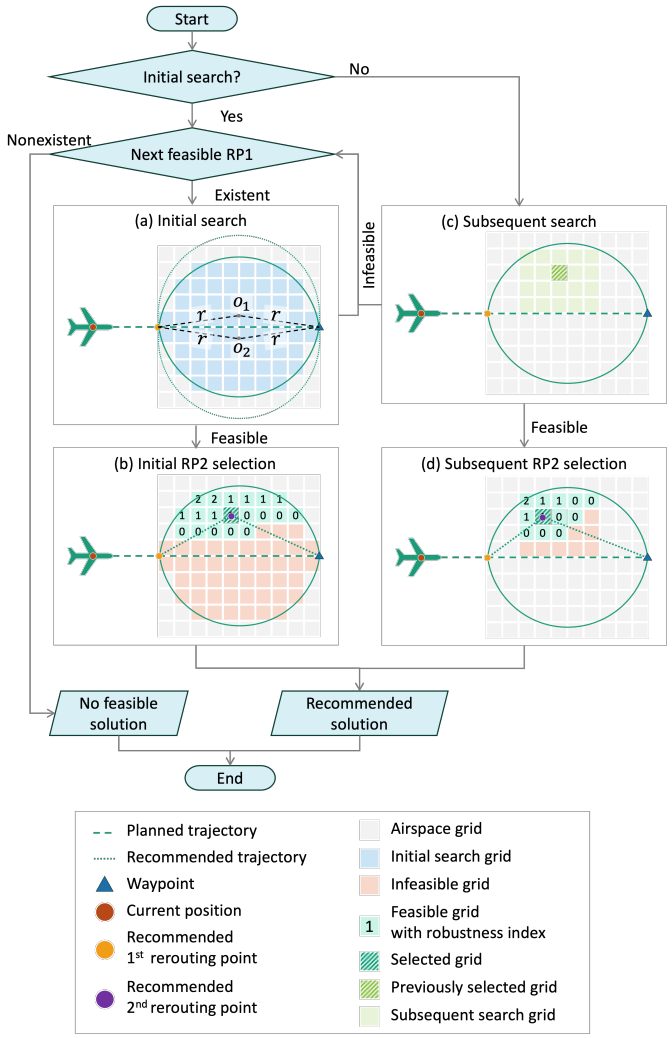


Figure 5. Recommended 2nd rerouting point search method.

optimal solution is then obtained using the same procedure as in step (2) and output as the new recommendation.

(5) **Fallback to initial search**. If the subsequent search fails to find a feasible solution, the process reverts to the initial search, i.e., steps (1)–(3) are executed again.

IV. Experiments

The experimental environment for this study is Python 3.12, running on macOS with an Apple M4 Pro chip and 24GB of memory. The proposed method is validated through purpose-built fast-time simulation experiments incorporating time uncertainty arising from human–machine and air–ground interactions.

A. Scenario and parameter setup

In this study, a simplified operational scenario was constructed (refer to Figure 6). The scenario was designed as a square single-flight-level airspace of 100×100 NM^2. Considering the semicircular flight level allocation rule, four routes were established, with two oriented horizontally and two vertically. Seven traffic density scenarios (measured by the number of flights entering the sector per hour) were considered: 40, 60, 80, 100, 120, 140 and 160 flights per hour, with 100 random instances generated for each density level. The entry time of each flight was determined as the reciprocal of the traffic density, with an additional uncertainty of ± 10 s, i.e., the entry time of the n-th flight was randomly generated within the interval $[n/d–10, n/d + 10]$ s, where d denotes traffic density. Flight routes were randomly selected, and flight speeds were randomly assigned as integer values between 450 and 490 knots. Each simulation covered one hour of traffic, meaning that all flights generated within the one hour were simulated until they exited the sector.

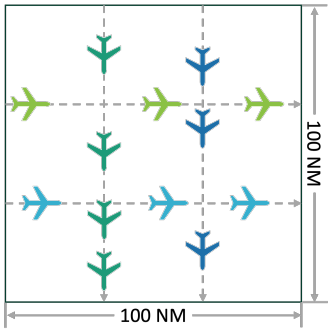


Figure 6. Illustration of the experimental scenario.

TABLE I. Parameter set-up

Parameter	Symbol	Value
Minimum safe separation distance	D^{SS}	5 NM
Grid length	L^{G}	5 NM
Maximum yaw angle	$\theta_{\max}$	$\pi/3$ [8]
Advisory update interval	T^{U}	5 s
Reserved decision-making time	T_i^{D}	2.76 s [11]
Minimum decision-making time	$T_i^{\mathrm{D,min}}$	1.91 s [11]
Reserved communication time	$T_{i,1}^{\mathrm{C}}, T_{i,2}^{\mathrm{C}}$	17.74 s [12]
Reserved communication time before rerouting point	$T_{i,1}^{\mathrm{CB}}, T_{i,2}^{\mathrm{CB}}$	9.9 s [12]
Reserved communication time after rerouting point	$T_{i,1}^{\mathrm{CA}}, T_{i,2}^{\mathrm{CA}}$	7.84 s [12]
Minimum communication time	$T_{i,1}^{\mathrm{C,min}}, T_{i,2}^{\mathrm{C,min}}$	4 s [12]
Weighting coefficient	ω	50

Table I presents the key parameter settings used in the experiments. It should be emphasised that, according to previous research findings, which indicate that traffic density and complexity have no significant effect on controller communication time [13], this study assumes that human–machine interactions and air–ground communications follow the same distributions across all scenarios. In the simulation, interaction times are randomly generated as follows: Decision-making time and communication time are modelled using the distributions and parameters reported in [11] (lognormal) and [12] (normal), respectively. In the controller workload model, the reserved communication time and the minimum decision-making time are set to the upper and lower bounds of the 95% confidence intervals of the distribution. The reserved communication time before the rerouting point is set to the mean of the distribution,

while the reserved communication time is set to the upper bound of the 95% confidence interval; the difference between the two yields the reserved communication time after the rerouting point. Minimum communication time is provided by [12].

B. Comparison models

To validate the effectiveness of the key technical components of the proposed method, four comparison models were designed (refer to Table II). They mainly differ in three aspects: **workload constraint** means whether controller workload is considered; **heuristic search** means whether heuristic search is adopted, otherwise global search is applied; **uncertainty** means whether the time uncertainty of human–machine and air–ground interactions is taken into account. In the table, a check mark (✓) indicates inclusion, while a dash (–) indicates exclusion. The four models are named W, H, WH, and WHU, where the letters denote the features included. Model WHU is used to evaluate the performance of the proposed method under uncertainty scenarios close to real-world operations. In addition, a comparison between models W and WH is conducted to verify the effectiveness of the proposed heuristic search approach, while a comparison between models H and WH is carried out to explore the impact of controller workload constraints on conflict detection and resolution. It should be noted that since Models W, H, and WH do not account for uncertainty, their reserved decision-making time and communication time are set to the mean of the corresponding distributions (2.31 s [11] and 9.9 s [12], respectively) rather than the upper bound of the 95% confidence interval, and the corresponding interactions are assumed to be completed exactly at the end of the reserved period.

TABLE II. COMPARISON MODELS

Model	Workload constraint	Heuristic search	Uncertainty
W	✓	-	-
H	-	✓	-
WH	✓	✓	-
WHU	✓	✓	✓

C. Statistical analysis of experimental results

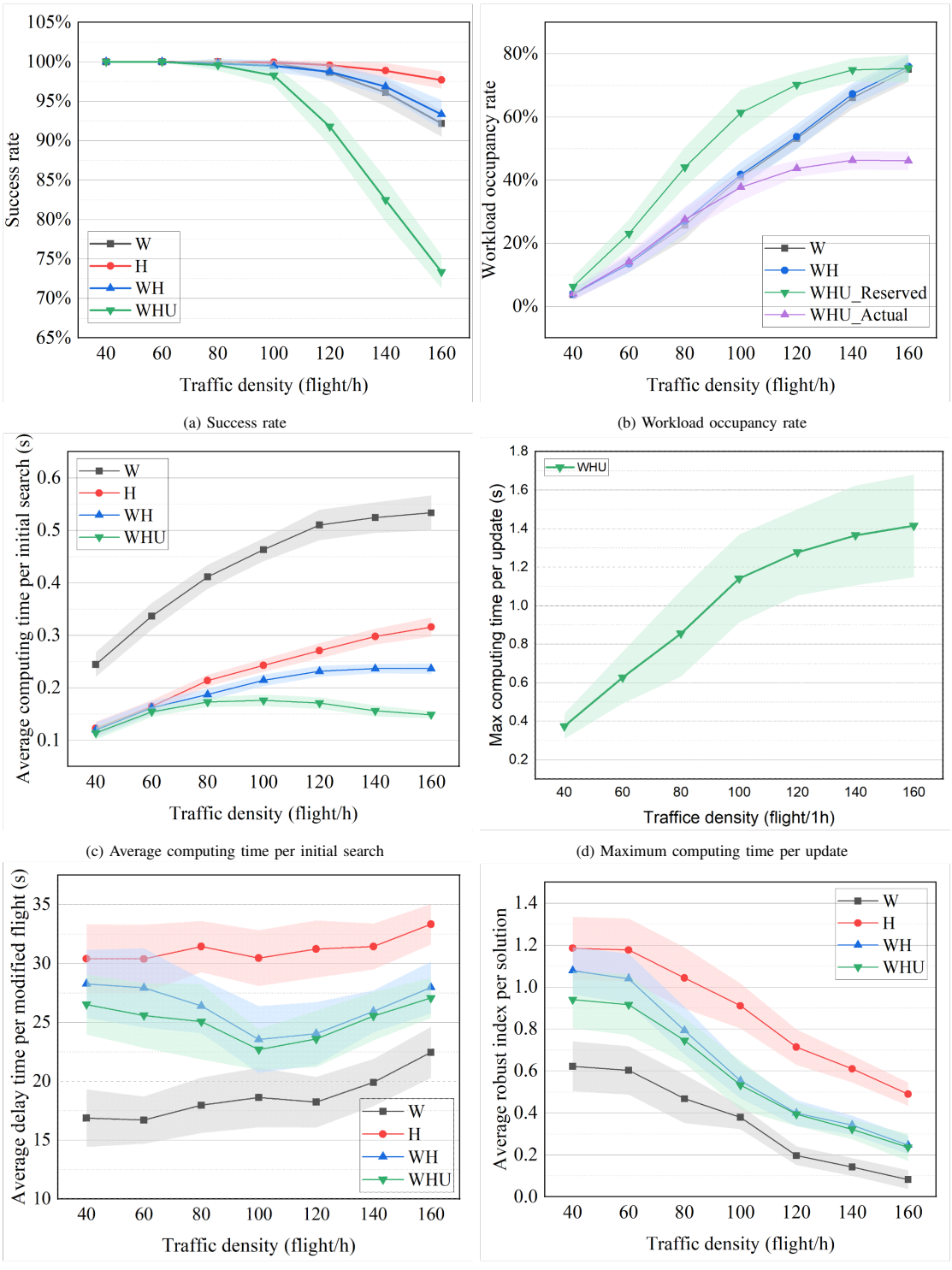
Figure 7 presents the experimental results. Sub-figure 7a illustrates the success rate of conflict resolution, defined as one minus the ratio of unresolved flights to the total number of flights. Sub-figure 7b shows the controller workload occupancy rate; since Model WHU accounts for uncertainty, both its reserved and actual values are presented separately. Sub-figure 7c presents the average computing time per initial search to evaluate computational performance. Sub-figure 7d reports the maximum computing time per update, including conflict detection for all flights and the necessary conflict-free trajectory searches. In this figure, only WHU is shown because it is the sole model that accounts for uncertainty and therefore requires continuous computation to respond to it, making this metric meaningful only for this model. Sub-figure 7e shows the average delay time per modified flight, measured as the difference between actual and planned arrival times at sector exit points. Sub-figure 7f presents the average robustness index per solution. The solid lines in the figures indicate the mean results of the corresponding models, while the shaded areas represent the standard deviations of the respective statistics. Based on these experimental results, the following key findings can be drawn:

Success rate (sub-figure 7a): Since Model H does not include controller workload constraints, it enjoys a larger solution space and therefore achieves the highest success rate. In contrast, Model WHU sets longer reserved decision-making and communication times than Models W and WH (see subsection IV-B and sub-figure 7b) to address uncertainty, resulting in the smallest solution space and thus the lowest success rate. In the 160 flights per hour scenario, the success rate of Model WHU is up to 24.3% lower than that of Model H, suggesting that conventional CDR methods that neglect workload constraints, even if applicable, may suffer from substantial performance degradation in practice. Nevertheless, under simulated uncertainty scenarios (Model WHU), the proposed method still achieves a 100% conflict resolution success rate at traffic densities of 60 flights per hour, twice as high as the maximum observed in real-world operations [14].

Computation time (sub-figures 7c and 7d). All models except Model W employ heuristic search, leading to significantly shorter solution times compared with Model W, which demonstrates the effectiveness of the proposed adaptive dynamic search in improving computational efficiency. The notable differences among Models H, WH, and WHU stem from the fact that, in the search algorithm, the relatively time-consuming process is the feasibility check of initial/subsequent search grids, and the number of grids to be checked depends on the size of the solution space. Because Model WHU has the smallest solution space, its computation time is the shortest. Even under the highest traffic density scenarios tested, the maximum computation time per update does not exceed 1.68 seconds, well below the update cycle of 5 seconds, confirming that the computational performance of the proposed method meets real-time operational requirements.

Optimisation objective (sub-figures 7e and 7f). These are the two optimisation objectives of the proposed models. Average delay time changes moderately across models, mainly because it is bounded by the yaw angle constraint. Even at the maximum of 35.0 seconds (model H), the delay accounts for only about 4.6% of flight time within the sector. The robustness index shows a decreasing trend with increasing traffic density, as higher densities aggravate conflicts, shrink feasible spaces, and reduce robustness values of solutions. Across different models, higher delay times are generally associated with higher robustness values. The detailed reasons behind this trade-off will be further investigated in future studies through sensitivity analysis of the weighting coefficients.

Controller workload occupancy (sub-figure 7b). Under the current models and parameter settings, reserved decision-



(a) Success rate

(b) Workload occupancy rate

(c) Average computing time per initial search

(d) Maximum computing time per update

(e) Average delay time per flight

(f) Average robust index per solution

Figure 7. Experimental results.

making and communication times occupy up to 79.8% of the total time. Two possible reasons are: (i) the remaining gaps are too short to accommodate a full reserved period, or (ii) the remaining time does not match the required operations. More critically, under uncertainty scenarios, the actual controller workload occupancy drops to below 49.1%. This suggests that the static workload allocation model adopted in this study may need to be improved into a dynamic one to enhance controller workload utilisation. For example, if a controller completes an operation earlier than expected, then it may be feasible to

reassign another operation to begin immediately.

V. Conclusion

The experiments demonstrate that the proposed method has strong potential to efficiently and reliably assist controllers in resolving conflicts under uncertainties in human–machine and air–ground interactions. It achieves a 100% success rate even at twice the maximum real-world traffic density while keeping computation times well below real-time operational limits, confirming both the effectiveness and efficiency of the proposed approach. However, controller workload utilisation falls below 49.1% in higher-density scenarios, which restricts further capacity improvements. Future work will therefore focus on developing dynamic workload allocation models to enhance utilisation and on validating the method in more realistic operational environments.

Acknowledgment

This research is under the SESAR HYPERSOLVER project, which has received funding from the SESAR Joint Undertaking under Grant Agreement No 101114820 under the European Union's Horizon 2020 research and innovation program for the partners (NeoMetSys, L'Ecole Nationale d'Aviation Civile, EUROCONTROL, LUFTFARTSVERKET) and National funding for the Associated Partners (Nanyang Technological University and University of Warwick).

This research has also been supported by the National Research Foundation, Singapore, and the Civil Aviation Authority of Singapore under the Aviation Transformation Programme (REQ0532039_FAA). Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Research Foundation, Singapore, or the Civil Aviation Authority of Singapore.

References

[1] J. Zhang, X. E, F. Du, J. Yang, S. Loft, The difficulty to break a relational complexity network can predict air traffic controllers' mental workload and performance in conflict resolution, Human factors 63 (2) (2021) 240–253.

[2] P. D. Mascio, R. Carrara, L. Frasacco, E. Luciano, A. Ponziani, L. Moretti, Influence of tower air traffic controller workload and airport layout on airport capacity, Journal of Airport Management 15 (4) (2021) 408–423.

[3] M. Ribeiro, J. Ellerbroek, J. Hoekstra, Review of conflict resolution methods for manned and unmanned aviation, Aerospace 7 (6) (2020) 79.

[4] E. Hernandez-Romero, A. Valenzuela, D. Rivas, Probabilistic multi-aircraft conflict detection and resolution considering wind forecast uncertainty, Aerospace Science and Technology 105 (2020) 105973.

[5] Y. Chen, M. Hu, L. Yang, Autonomous planning of optimal four-dimensional trajectory for real-time en-route airspace operation with solution space visualisation, Transportation Research Part C: Emerging Technologies 140 (2022) 103701.

[6] Z. Wang, W. Pan, H. Li, X. Wang, Q. Zuo, Review of deep reinforcement learning approaches for conflict resolution in air traffic control, Aerospace 9 (6) (2022) 294.

[7] Y. Guleria, D.-T. Pham, S. Alam, P. N. Tran, N. Durand, Towards conformal automation in air traffic control: Learning conflict resolution strategies through behavior cloning, Advanced Engineering Informatics 59 (2024) 102273.

[8] Y. Chen, M. Hu, L. Yang, Y. Xu, H. Xie, General multi-agent reinforcement learning integrating adaptive manoeuvre strategy for real-time multi-aircraft conflict resolution, Transportation Research Part C: Emerging Technologies 151 (2023) 104125.

[9] M. Jameel, L. Tyburzy, I. Gerdes, A. Pick, R. Hunger, L. Christoffels, Enabling digital air traffic controller assistant through human-autonomy teaming design, in: 2023 IEEE/AIAA 42nd Digital Avionics Systems Conference (DASC), IEEE, 2023, pp. 1–9.

[10] G. de Rooij, A. B. Tisza, C. Borst, Flight-based control allocation: Towards human–autonomy teaming in air traffic control, Aerospace 11 (11) (2024) 919.

[11] Y. Shang-wen, H. Ming-hua, Estimation of air traffic longitudinal conflict probability based on the reaction time of controllers, Safety Science 48 (7) (2010) 926–930.

[12] K. M. Cardosi, Time required for transmission of time-critical air traffic control messages in an en route environment, The International Journal of Aviation Psychology 3 (4) (1993) 303–313.

[13] Y. Wang, F. Vormer, M. Hu, V. Duong, Empirical analysis of air traffic controller dynamics, Transportation Research Part C: Emerging Technologies 33 (2013) 203–213.

[14] EUROCONTROL, Performance review report an assessment of air traffic management in europe during the calendar year 2018, Tech. rep., EUROCONTROL, Brussels, Belgium (2018).