Fourteenth USA/Europe Air Traffic Management Research and Development Seminar (ATM2021)

# Towards AI-based Air Traffic Control

Mudhakar Srivatsa, Raghu Ganti, Linsong Chu
IBM TJ Watson Research Center
{msrivats, rganti, lchu}@us.ibm.com

Martin Christiansson, Jens Nilsson, Sofia Rydell, Billy Josefsson
LFV - Sweden's leading provider of air traffic control and associated services for civil and military aviation
{martin.christiansson, jens.nilsson, sofia.rydell, billy.josefsson}@lfv.se

***Abstract*—Air traffic, despite the recent dip due to Covid, is expected to grow 30-40% year on year. With the potential inclusion of UAVs (Unmanned Aerial Vehicles) into controlled airspace over the next decade, it is anticipated that the congestion levels in airspace will increase 10 fold. This paper presents an AI-based approach to air traffic control, with the aim of alleviating the load and improving the efficiency of human agents (air traffic controllers). One of the primary goals of air traffic control is to safely navigate an aircraft through controlled airspace using real-time control actions - such as changes to speed, heading (direction of travel) and altitude of an aircraft. The safety critical nature of this environment calls for precise explanations (why take an action) and counterfactual (why not take an action) explanations, real-time responsiveness, the ability to present succinct actions to a human agent, while simultaneously optimizing for air traffic delays, fuel burn rates, and weather conditions.**

**This paper presents algorithms and a system architecture for anticipating separation losses (conflicts in airspace) and a lattice-based search space exploration AI planner to recommend actions to avoid such conflicts. The key contributions of the paper include: (i) fast detection (prediction) of conflicts in a controlled airspace, and (ii) fast lattice space exploration based AI solver to produce a set of feasible resolutions for the detected conflicts. Additionally, this paper discusses how to weight the different resolutions and how future work on optimisation techniques could improve the efficiency of the algorithm and address various known limitations of the current approach from both technical and human-agent perspective. The evaluations are conducted against an air traffic simulator, Narsim, showing the ability to avoid separation losses, while minimizing the number of actions even at 3 x normal capacity.**

***Keywords*—AI planner; Trajectory analysis; Conflict detection; lattice-based search space exploration; automation; air traffic management**

## I. Introduction

Expected growth in air transportation, and increasing demand in airspace capacity has lead to a need for more efficient air traffic management (ATM). Research in ATM has proposed different solutions to this issue including Airspace Management and Advanced Flexible use of Airspace [6], Trajectory Based Operations [7], [8], Enhanced tactical conflict detection and resolution [9], [10]. Many of these prior efforts attempt to increase level of automation while still preserving the role of the human agent as the utmost responsible element in the decision-making process.

In safety critical environments, it is often not sufficient to rely solely on automation oriented solutions. Therefore the infrastructure of such systems must be cognizant of performance of human agents. Cognitive limitations, e.g., multitasking, stress, memory, prediction, understanding, etc. must be accounted for.

Recently, ATM has entered a new era, where novel Artificial Intelligence (AI) and Machine Learning (ML) techniques are being explored. A substantial economical effort is spent in ATM research to integrate and investigate these new tools and how well they can inter-operate with the human agents in a safety critical environment. One of the underpinning challenges when applying AI/ML oriented techniques in this environment is explainability, i.e., how well the machine decision can be interpreted and understood by the human agent. The key challenge is therefore to utilize rich AI/ML models (e.g., deep learning) while ensuring a high level of explainability that is required for mission critical applications like air traffic control.

Considering AI/ML automation for an en-route sector, the solution must be able to identify possible conflicts, i.e., predict aircraft future behaviour, as well as finding solutions to resolve these conflicts. The key underlying approach developed and explained in this paper - termed Advanced Autoplanner (AAP) AI model - is to first identify possible conflicts for all aircraft within a specified en-route sector. Second, to find resolution for these conflicts by: (i) identifying safe actions that avoid future conflicts; and (ii) ranking these safe actions based on their optimality. It is evident that phase (i) has higher explainability requirements. Accordingly, the model uses lattice-based search space exploration as an AI solver that is amenable to both explanations (why take this action) and counter-factual explanations (why not take this action). The feasible course of action set, derived by the model is subject to optimality conditions and prioritization based on pre-defined rules that closely mimic how an air traffic controller works in today's operation. Discretization of the problem enables us to apply white-box AI models with better explainability for finding safe actions and black box AI/ML models including deep learning to learn ranking of these safe actions and suggest appropriate actions to the human agents. In this project we have focused on the first step, i.e. the white-box AI that finds safe actions to resolve the conflicts. In the future, the second step with black-box AI/ML models could be added for optimised performance.

The paper describes a detailed evaluation of this approach using an air traffic control simulator (NarSim [13]). The scenarios for evaluation were produced in close collaboration with air traffic controllers with decades of experience. The evaluations cover 10 scenarios under normal air traffic load and 10 scenarios under higher air traffic load, where each scenario covers about two hours of air traffic. Evaluations are conducted along various dimensions including the ability to avoid separation loss, reducing track lengths (and thus fuel burn rate), reducing the total number of actions (while prioritizing lower cost actions such as ground speed change and heading change over altitude change), and reducing the number of actions per unit time (six per minute) so as to not overload human agents. A video demonstration, summarizing the findings is available here [14].

## II. System and Architecture

Supporting the ATM environment requires a highly available (HA), fault tolerant, and real-time responsive system. Even though this paper is using only simulations, we designed our system to operate with responsiveness similar to real-world scenarios. A key design point is that the system should perform in dense and complex traffic situations.

The high level architecture is illustrated in Figure 1, which is comprised of a real-time streaming environment (Autoplanner-Streams), which is a scalable real-time platform on IBM Cloud. This platform was chosen because of its fault tolerance, high availability, and real-time features. We use databases such as Db2 and Cloudant for providing resilience in the event of failures. The Streams platform connects to Narsim instances, where messages of the current state of the airspace and the aircrafts flow from Narsim to this platform and recommendations to change altitude, speed, and heading are provided back to the simulator from the platform. The communication is over a TCP network socket. This is a fairly standard HA, fault tolerant, and real-time architecture. In the Figure 1, the boxes in pink are incomplete and they might be implemented in future iterations (discussed in Section VII).

A key element to observe from this figure is the different types of data received from Narsim; the relevant ones for the AAP are: (a) Radar track, which captures the current location, altitude, heading, ground speed, ascent/descent rate of a single aircraft; (b) Flightplan, which is a periodic broadcast that is the planned path an aircraft is supposed to be on, and (c) No-fly zones, which are polygonal regions with altitude ranges that an aircraft is not allowed to fly through without specific clearance from air traffic control and (d) STCA (Short Term Conflict Alert).

The Autoplanner AI box in Figure 1, is a key processing block that determines if and what action needs to be suggested to the Air traffic controller. The key inputs include: (a) Flightplan including the waypoints for an aircraft's planned 3d path, (b) Radar Track including the current observation on aircraft's position (latitude, longitude and altitude), ground speed and heading, (c) No-Fly zones that are represented as polygons and altitude ranges (though our current implementation does not explicitly handle them), (d) environmental conditions including wind speed, bad weather zones, etc. (the current implementation does not handle them). In addition, for validation purposes, we use STCA (Short Term Conflict Alert) from Narsim so that the conflicts that Narsim detects can be compared with the ones that the AI model detects.

The zoomed in version of the AI model along with the pre-processing and post-processing steps are shown in Figure 2. As the reader can see, it consists of three main components. The data parser, which is the first component, is a pre-cursor step, which enriches the Radar track with information from the flightplan as well as aircraft performance data. We note that the aircraft performance data is static and is loaded at initialization time.

The second stage of the AI model is to determine future encounters given the updated state. This is a key design choice for the system, where unless an observation from the Radar track is input to the system, it does not determine any conflicts nor suggest any actions, i.e., the system is event driven. Given this design choice, it is only needed to evaluate if the incoming observation is resulting in a state change, hence it suffices to determine if the aircraft corresponding to this observation can potentially result in an encounter in the future. It will be explained in more depth in the later sections regarding how this is determined.

The final stage of the AI model is to provide action recommendations if an encounter is detected. We use an approach where the AI model's forecaster is simulated with a search strategy to determine next best actions. These actions when performed within a certain time period will eliminate any potential encounters. We determine these actions using the lattice traversal, which is detailed in the next section.

The system detects and derives resolutions for conflicts, these actions are then ranked based on *ranking criteria*. Ranking criteria take into account the amount of change and can also provide differentiated weighting based on the type of change (e.g., speed change is preferred to altitude change). These ranked actions are then communicated back to Narsim in a sequential manner, where if Narsim rejects the proposed action, the next action in the ranked list is communicated.

Finally, the AI model has configuration parameters that control how far into the future the encounter detection forecasting can be done, the search radius, and the altitude search radius. We also provide the option of running multiple Autoplanners in parallel for providing various options. Given that the underlying platform scales horizontally, adding multiple instances of AAP does not hurt system performance and its real-time responsiveness. These are validated through exhaustive experimentation.

In the next two sections, we will describe the core algorithmic innovation of this paper. We will show how to detect loss of separation followed by searching for the successful actions using a lattice search space technique.

## III. Detecting Separation Loss

This section describes the approach that we have adopted for detecting separation loss between any two aircraft. Since

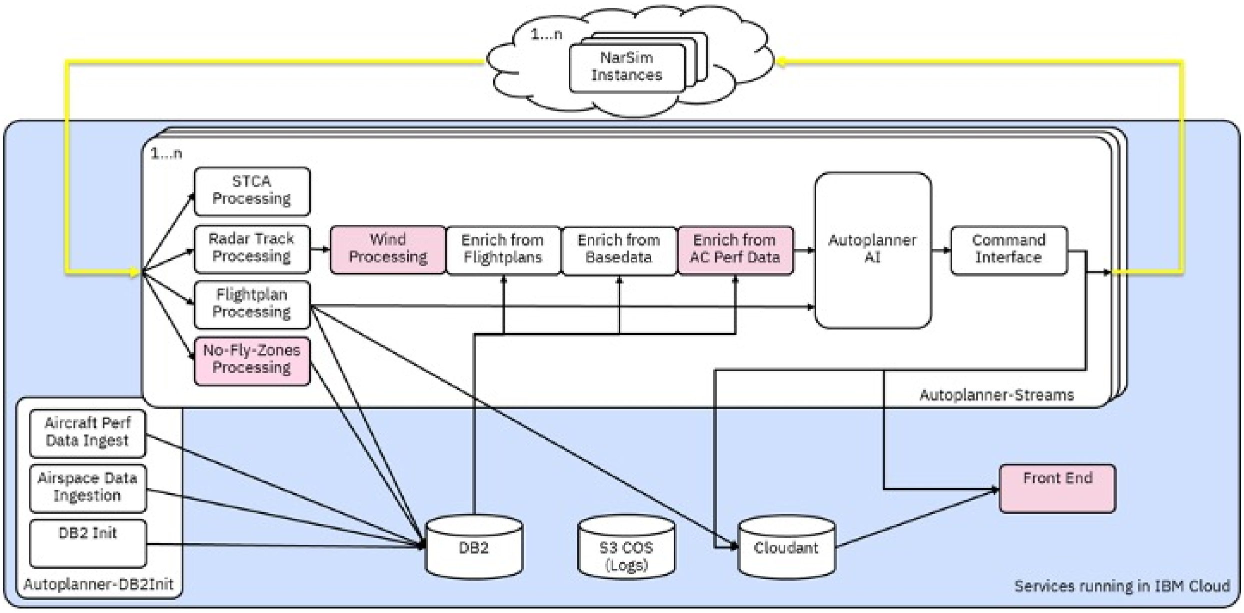


Figure 1: System architecture in Cloud

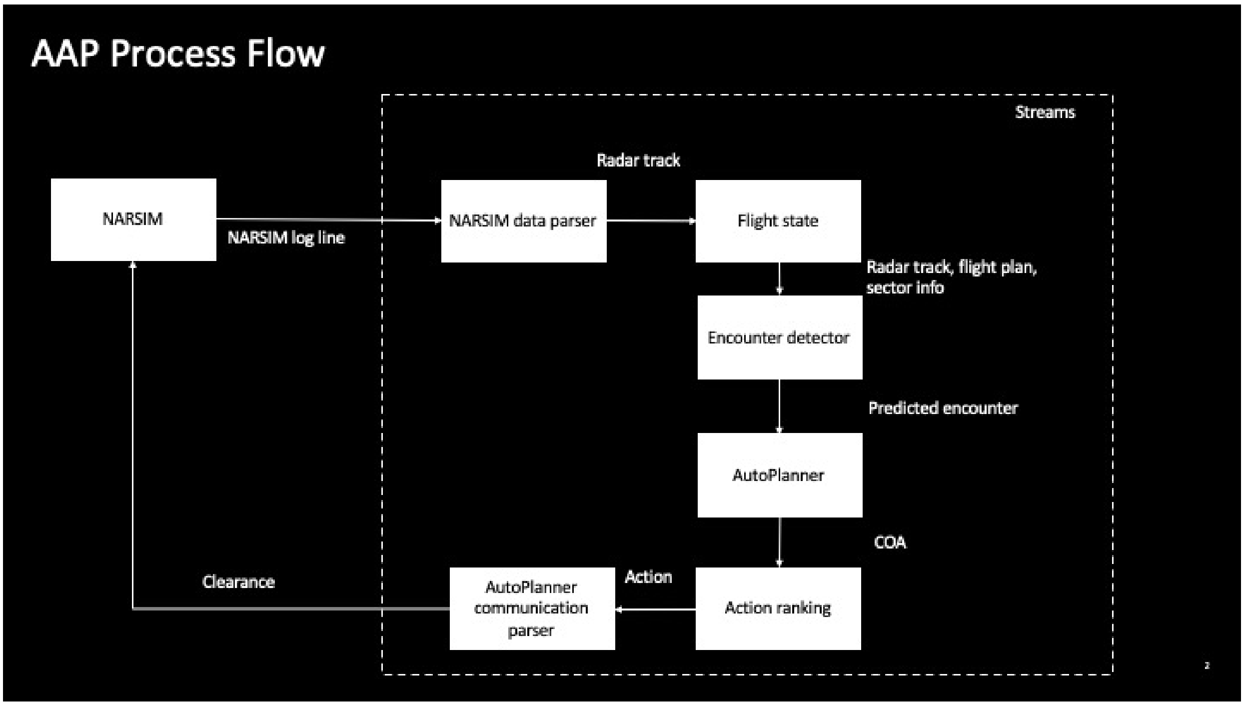


Figure 2: A deeper view of Autoplanner AI

we take an event driven approach, we evaluate separation loss on every Radar observation that comes into the system.

Given the current state of an aircraft (with the flight plan) we determine a 3d track of the aircraft over a time horizon of length $T$ (typically 20 minutes). The 3d track of an aircraft is represented as piece-wise continuous 3d line segments. The points of discontinuity occur at end of an ascent/descent or at a way point wherein the aircraft needs to orient itself towards the next waypoint.

The state of an aircraft is updated every second. Separation loss is defined as $ll_{sep}$ on the latitude/longitude dimension (5 nautical miles) and as $alt_{sep}$ on the altitude dimension (1000 feet). As the state of an aircraft is updated periodically, the goal is to efficiently detect separation loss, including time to separation loss and the closest separation between two aircraft. This is accomplished as follows:

- Construct the future 3d track of an aircraft based on its current state and flight path. On the latitude/longitude dimensions, the movement of the aircraft is modeled as elliptic arcs to account for the curvature of the Earth. All distance computations on the latitude/longitude are performed on the Ellipsoidal Earth. On the altitude dimension, the movement of the aircraft is modeled using a constant ascent/descent rate with a target altitude per the flight plan.

- Project the future 3d tracks onto the latitude/longitude 2d coordinate system and store them in an in-memory spatial index. Given an updated aircraft state, an updated 2d track is computed and a *within distance* search is performed on this index to identify other 2d tracks that are within $ll_{sep}$ on the latitude/longitude dimensions. The fast in-memory spatial index allows performing millions of queries per second. This test is guaranteed to have zero false negatives, i.e., any two aircrafts that will suffer a separation loss are guaranteed to be identified. However, there may be false positives, i.e., not all pairs of aircrafts identified will suffer from a future separation loss. This is because this test disregards the instantaneous position of the aircraft on both latitude/longitude and altitude dimensions. These false positives are eliminated in the subsequent steps.
- For every aircraft whose 2d tracks are within $ll_{sep}$ from each other, the closest altitude separation between the aircraft is computed. Aircraft with altitude separation below $alt_{sep}$ are selected. This essentially identifies two aircraft whose tracks are below $ll_{sep}$ distance on the latitude/longitude dimensions and below $alt_{sep}$ distance on the altitude dimension. Like the previous step, this step has zero false negatives, but it still has some false positives because it disregards the instantaneous position of an aircraft on its 3d track.This step eliminates false positives due to the altitude changes.
- For every aircraft whose 2d tracks are within $ll_{sep}$ and whose altitude separation is below $alt_{sep}$, the closest instantaneous separation is computed between them using fixed point iterations. The fixed point iterations are guaranteed to converge, since projectile motion along two great spherical/elliptic arcs have a unique separation minima. This identifies all aircraft pairs that will have separation loss, the time to separation loss, and closest separation between the two aircraft. The time to separation loss and the closest separation are used to prioritize resolution actions on the aircraft. At the end of this step we have zero false positive and false negative rate given the current state of the aircraft, under an elliptic arc model of the aircraft trajectory. Since the state of the aircraft may change in the future (or external factors such as wind may influence the state), these steps are applied on receiving every Radar track event from every aircraft.

## IV. State and Action Space

This section describes the approach that we take to explore an action space which is a class of optimization techniques that does a "clever" exploration of the possible solution space. The state of an aircraft is represented by a six tuple:

$$S = [lat, lon, alt, gs, az, ads]$$

where $lat$ denotes the latitude, $lon$ denotes the longitude, $alt$ denotes the altitude, $gs$ denotes the ground speed, $az$ denotes the azimuth (heading) with respect to true North and $ads$ denotes the ascent/descent speed.

In addition to the instantaneous state of an aircraft, it also has a flight plan $FP$ represented by a list of waypoints (coordinates in latitude and longitude) and target altitudes (when the aircraft is either ascending/descending). Hence, FP is defined as

$$FP = [wp_1, \cdots, wp_n], target\ alt$$

where $wp_i$ denotes the waypoints enroute. Note that $target\ alt$ is optional and only applies when $ads \neq 0$, i.e., when the aircraft is ascending/descending.

In this model, there are three dimensions of control for an aircraft: ground speed, azimuth (heading) and altitude. Action $A$ is represented as:

$$A = [\delta gs, \delta az, \delta alt]$$

where $\delta gs$, $\delta az$, $\delta alt$ can be positive, negative or zero (no change).

On applying an action $A$ to an aircraft in state $S$ the state of the aircraft changes as follows:

$$S' = [lat, lon, alt, gs + \delta gs, az + \delta az, ads']$$
$$FP' = [wp_1, \cdots, wp_n], alt + \delta alt$$

where $ads'$ = default ascent rate of 1000 feet/minute when $\delta alt > 0$, or default descent rate of 1000 feet/minute when $\delta alt < 0$, and zero otherwise. $ads$ is a configurable parameter that can be changed based on aircraft characteristics. The target altitude for the aircraft is $alt$+$\delta alt$. The list of waypoints associated with the flight plan remains unchanged, although the list may contain infeasible waypoints due to an action (e.g., when $\delta az \neq 0$). Note that this model makes a simplifying assumption that the ground speed and azimuth actions are instantaneous. However, the relatively slower altitude change action is not assumed to be instantaneous.

The action space is discretized based on a minimum quanta of 0.01 mach for $\delta gs$, 5° for $\delta az$ and 1000 feet for $\delta alt$. These are configurable parameters that can be changed by a subject matter expert with the following tradeoffs. Using fine-grained quantas increase the search space, potentially increasing the compute power required to determine feasible actions; however, using fine-grained quantas help identify actions that are smaller in magnitude and thus reduce the deviation from the current flight plan. The action space allows $N_{gs}$, $N_{az}$ and $N_{alt}$ quanta of change in each of the three dimensions, respectively. For example, when $N_{gs} = 10$ the number of quanta of change to the ground speed dimension is $2 * N_{gs} + 1 = 21$, ranging from $\{-10, -9, \cdots, 0, \cdots, +9, +10\}$ which correspond to $\{-0.1$ mach, $-0.09$ mach, $\cdots$, 0 mach, $\cdots$, $+0.09$ mach, $+0.1$ mach$\}$ of change to the aircraft's ground speed. The action space can be tuned by altering the definition of one quanta in any of the three dimensions and by altering the number of quanta of change permissible in each of these dimensions.

The action space is modeled as a *lattice*. Figure 3 shows a lattice with $N_{gs} = 5$, $N_{az} = 3$, $N_{alt} = 3$, i.e., 5 quanta of action on ground speed and 3 quanta of action on both the azimuth and altitude dimensions. There is an additional dimension with values $\{+1, -1\}$ not shown in this figure. Hence, each action $A$ = $[\delta gs, \delta az, \delta alt]$ corresponds to utmost $2^3 = 8$ actions, namely:

1) $[+\delta gs, +\delta az, +\delta alt]$
2) $[+\delta gs, +\delta az, -\delta alt]$
3) $[+\delta gs, -\delta az, +\delta alt]$
4) $[-\delta gs, +\delta az, +\delta alt]$
5) $[+\delta gs, -\delta az, -\delta alt]$
6) $[-\delta gs, +\delta az, -\delta alt]$
7) $[-\delta gs, -\delta az, +\delta alt]$
8) $[-\delta gs, -\delta az, -\delta alt]$

Note that the number of actions could be smaller than 8 since one of the dimensions is zero. For example, when $\delta gs \neq 0$, $\delta az \neq 0$ and $\delta alt = 0$, there are only four actions since $+\delta alt$ = $-\delta alt = 0$. For the sake of simplicity, we will leave the $\{+1, -1\}$ dimension implicit in the subsequent descriptions. Also, in subsequent discussions we will denote an action using integers (e.g., $[2, 1, 1]$), denoting the number of quantas of change in each of the three dimensions $gs$, $az$ and $alt$.

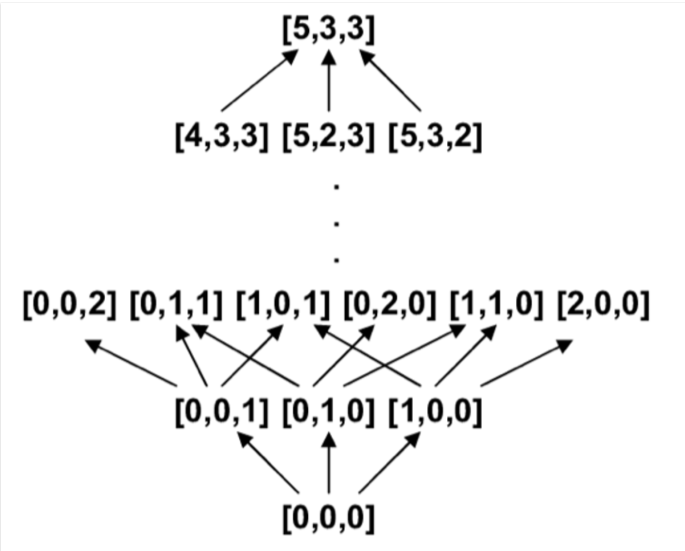


Figure 3: Action Space Modeled as a Lattice

A lattice consists of a partially ordered set in which any two elements have an unique *supremum* and an unique *infimum*. Given any two actions $A_1$ = $[\delta gs_1, \delta az_1, \delta alt_1]$ and $A_2$ = $[\delta gs_2, \delta az_2, \delta alt_2]$, action $A_1$ is said to dominate action $A_2$ if and only if ($\wedge$ denotes the logical AND operator):

$$A_1 \geq A_2 \iff (\delta gs_1 \geq \delta gs_2) \wedge \\ (\delta az_1 \geq \delta az_2) \wedge \\ (\delta alt_1 \geq \delta alt_2)$$

Unlike completely ordered sets it is possible that neither $A_1 \geq A_2$ nor $A_2 \geq A_1$ hold, in which case $A_1$ and $A_2$ are said to be incomparable to each other (denoted as $A_1 \parallel A_2$). For example $[2, 0, 0] \geq [1, 0, 0]$ and $[2, 0, 0] \parallel [1, 1, 0]$.

Given any two actions $A_1$ and $A_2$ its unique supremum (lowest upper bound), i.e, the smallest action $A$ such that ($A \geq A_1$) $\wedge$ ($A \geq A_2$) and there exists no $A'$ such that $A' < A$ ($max(a, b)$ = $a$ if ($a \geq b$); otherwise $b$):

$$sup(A_1, A_2) = [max(\delta gs_1, \delta gs_2), \\ max(\delta az_1, \delta az_2), \\ max(\delta alt_1, \delta alt_2)]$$

and its unique infimum (highest lower bound), i.e., the largest action $A$ such that ($A \leq A_1$) $\wedge$ ($A \leq A_2$) and there exists no $A'$ such that $A' > A$ ($min(a, b)$ = $a$ if ($a \leq b$); otherwise $b$:

$$inf(A_1, A_2) = [min(\delta gs_1, \delta gs_2), \\ min(\delta az_1, \delta az_2), \\ min(\delta alt_1, \delta alt_2)]$$

For example, $sup([2, 0, 0], [1, 1, 0]) = [2, 1, 0]$ and $inf([2, 0, 0], [1, 1, 0]) = [1, 0, 0]$. Note that even though $[3, 2, 1]$ is greater than both $[2, 0, 0]$ and $[1, 1, 0]$, it is not the supremum since it is not the *lowest* upper bound. Similarly, even though $[0, 0, 0]$ is lower than both $[2, 0, 0]$ and $[1, 1, 0]$, it is not the infimum since it is not the *highest* lower bound.

In figure 3 a directed arrow $\rightarrow$ between from action $A_1$ to $A_2$ indicates that $A_1 < A_2$. The transitive property of $<$ relationship holds ($\alpha \implies \beta$ denotes the logical implication, i.e., if $\alpha$ is $true$ then $\beta$ is $true$:

$$(A_1 < A_2) \wedge (A_2 < A_3) \implies A_1 < A_3$$

Also for any two actions $A_1$ and $A_2$, such that $A_2$ is not reachable on the lattice from $A_1$ and $A_1$ is not reachable on the lattice from $A_2$ then $A_1 \parallel A_2$.

A lattice is also conveniently represented in *levels*. Action $A$ = $[\delta gs, \delta az, \delta alt]$ with total quanta of change of equal to:

$$l(A) = |\delta gs| + |\delta az| + |\delta alt|$$

As shown in the figure level 0 has one action $[0, 0, 0]$ and level 1 has three actions $[1, 0, 0]$, $[0, 1, 0]$ and $[0, 0, 1]$ and so on. The total number of possible actions is in the lattice is

$$num(A) = N_{gs} * N_{az} * N_{alt} * 2^3$$

and the number of levels is

$$max(l(A)) = N_{gs} + N_{az} + N_{alt} + 1.$$

A lattice state space naturally allows us to prioritize the search for safe actions and capture preferences in the air traffic control domain.

- First, the lattice may be explored in a breadth-first bottom-up manner. Such an exploration would proceed in the following order: $[0, 0, 0]$, $[0, 0, 1]$, $[0, 1, 0]$, $[1, 0, 0]$, $[0, 0, 2]$, $\cdots$, where the lattice is explored in increasing order of level and each level is explored left to right. This ensures that actions with lower quanta of change is explored first, naturally capturing the preference to actions with lower quanta of change.
- Second a lattice structure allows for efficient pruning of the action space. For instance, if an action $A_i$ is deemed *safe* then no action $A_j$ such that $A_i < A_j$ needs to be explored. For instance, if action $[1, 0, 0]$ is safe, then actions $[2, 0, 0]$, $[1, 1, 0]$, $[1, 1, 0]$ in level 2 can be pruned.

Indeed the entire sub-lattice whose infimum is $[1,0,0]$ can be pruned from the original lattice (as shown in the Figure 4.)

- Third, if an action $A_i$ is deemed *safe* then no action $A_j$ such that $l(A_j) > l(A_i) + threshold$. For instance, if action $[0,0,1]$ is safe and $threshold = 2$ then any action $A$ whose level is less than or equal to $l([0,0,1]) + threshold = 1 + 2 = 3$ will be explored. Hence, an action $[0,2,0]$ may be explored, while actions at level 4 or above such as $[0,4,0]$ and $[0,2,2]$ will be pruned.

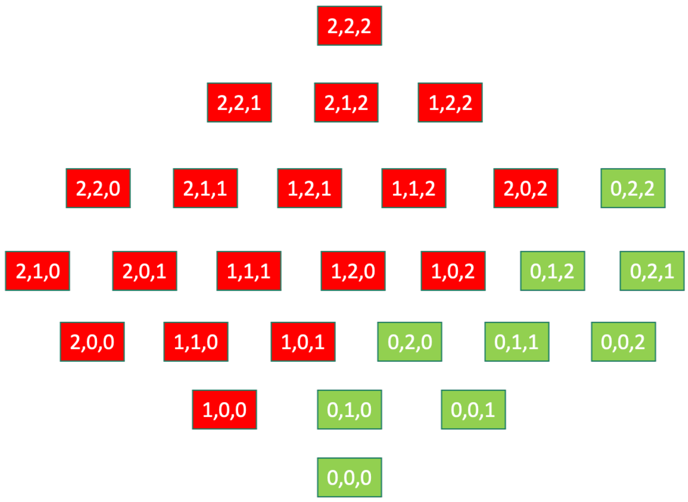


Figure 4: Lattice Pruning: Red nodes are pruned from the search space if action $[1,0,0]$ is successful

The lattice-based state space exploration's pruning strategy is succinctly expressed as follows ($\exists\alpha$ denotes the existence of $\alpha$ and $\vee$ denotes the logical OR operator):

$$skip(A_j) = \exists A_i,\ safe(A_i) \wedge \\ [(A_i < A_j)\ \vee\ (l(A_i) + threshold < l(A_j)]$$

A lattice-based search space exploration approach naturally lends itself to both explanations (why take an action) and counterfactual explanations (why not take an action). In particular, the minimum separation between the aircraft in conflict can be identified when an action is recommended. Similarly, when an action is not recommended, the resulting conflicts are identified; for example, an attempt to resolve a conflict between aircraft $\alpha$ and $\beta$ may result in a new conflict between $\alpha$ and $\gamma$. The cascading effect of such actions are explicitly addressed in this solution.

Further, it may be infeasible to take actions on aircraft that are outside the air traffic controller's sector. In such cases, if one of the aircraft is currently inside the sector, then actions are explored only for the aircraft inside the sector. If neither of the aircraft are inside the sector then actions are explored the moment the first aircraft enters the sector or the situation is resolved otherwise (e.g., by actions taken on these aircraft in other sectors). Finally if no action is deemed safe, then the action that maximizes the minimum separation between the aircrafts is chosen.

## V. Evaluation

This section presents the evaluation of the AAP using Narsim. A variety of scenarios have been constructed in close collaboration with air traffic controllers with decades of experience. The results below cover 20 runs. Runs 1-10 were performed with current sector load in the concerned sector. Runs 11-20 were performed with an increased sector load. The definition of sector load is the maximum number of flights at one time in a sector. Each run was approximately 2.5 hours long. In both these scenarios of loads, the AAP found solutions to address the separation losses in a quick manner with low costs in most cases. However, given the importance of safety, we will explore the rare cases when AAP could not find a solution.

### A. Separation Losses

Our analysis indicates that the separation losses can be linked either with a set climb rate being used when predicting future conflicts or a congestion problem in the AAP solution. The problem with climb rate can be solved by using a varying and more accurate rate of climb rate that consider the aircraft performance. The congestion problem can be solved by:

- parallelizing search over the lattice space (evaluate multiple feasible actions in parallel),
- reduce the search space by investigating single dimensional actions first (avoid actions that require changes to two or more dimensions), and
- suppress state radar track events from being sent to the AI planner to ease back pressure during congestion.

The separation losses that occurred can be divided into the below scenario types:

- One aircraft enters the sector while climbing, the other aircraft is inside the sector (Run 5, Run 16 and Run 17): these separation losses occur when one aircraft has recently entered the sector and is climbing towards an altitude above the one that the other aircraft is cruising at. During the climb lateral separation is lost while there is less than 1000 feet altitude separation between the two aircrafts. These conflicts occur because the model currently uses a default climb rate (1000 feet per minute ascent/descent) when predicting conflicts. According to the model's predictions the flight that is entering the sector will climb steeper than what it actually does and therefore it predicts that it will not encounter the other aircraft. This type of situation can be solved by using a more accurate rate of climb in trajectory predictions.
- One aircraft enters the sector on the same flight level as another aircraft inside the sector (Run 12): the separation loss occurs between one aircraft that has been in the sector for a while and one that has entered the sector shortly before on the same flight level. One of the aircraft is given a climb clearance a short while after they are both in the sector, this is however not sufficient and separation is lost, this is due to the same reason as above regarding set climb rates being used for trajectory predictions. However, as one of the aircraft had been in the sector for a while, the model should have issued an

| Indicator/Run | Run 1 | Run 2 | Run 3 | Run 4 | Run 5 | Run 6 | Run 7 | Run 8 | Run 9 | Run 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| Sector load | 18 | 18 | 19 | 19 | 19 | 19 | 21 | 18 | 17 | 20 |
| # predicted conflicts | 28 | 23 | 11 | 12 | 20 | 22 | 15 | 10 | 13 | 6 |
| # instructions | 31 | 32 | 21 | 21 | 21 | 25 | 18 | 18 | 22 | 9 |
| Separation losses ($<$ 5 NM) | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| # clearances / minute $>$ 6? | No | No | No | No | No | No | No | No | No | No |
| Track miles AAP active / AAP inactive | 1.0 | 1.0 | 1.0 | 1.0 | 0.997 | 0.999 | 0.999 | 1.0 | 1.0 | 1.0 |
| # speed instructions | 16 | 18 | 10 | 12 | 12 | 13 | 11 | 10 | 9 | 5 |
| # heading instructions | 11 | 11 | 8 | 4 | 7 | 9 | 5 | 6 | 9 | 3 |
| # flight level instructions | 4 | 3 | 3 | 5 | 2 | 3 | 2 | 2 | 4 | 1 |
| % actions over 3000 feet or 0.06 Mach or 30 degrees | 39% | 41% | 24% | 67% | 33% | 36% | 56% | 44% | 50% | 56% |
| % clearances with multiple instruction types | 82% | 32% | 82% | 75% | 67% | 92% | 64% | 80% | 82% | 80% |

TABLE I. Runs 1-10 - 2.5 hour runs based on current Sector Load

| Indicator/Run | Run 11 | Run 12 | Run 13 | Run 14 | Run 15 | Run 16 | Run 17 | Run 18 | Run 19 | Run 20 |
|---|---|---|---|---|---|---|---|---|---|---|
| Sector load | 25 | 27 | 26 | 25 | 26 | 27 | 29 | 25 | 27 | 26 |
| # predicted conflicts | 29 | 15 | 38 | 39 | 38 | 43 | 32 | 28 | 25 | 22 |
| # instructions | 35 | 30 | 40 | 29 | 51 | 39 | 38 | 51 | 28 | 34 |
| Separation losses ($<$ 5 NM) | 0 | 1 | 0 | 0 | 1 | 2 | 1 | 1 | 0 | 0 |
| # clearances / minute $>$ 6? | No | No | No | No | No | No | No | No | No | No |
| Track miles AAP active / AAP inactive | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| # speed instructions | 19 | 15 | 18 | 16 | 24 | 21 | 21 | 28 | 14 | 17 |
| # heading instructions | 8 | 11 | 18 | 10 | 18 | 12 | 13 | 19 | 8 | 13 |
| # flight level instructions | 8 | 4 | 4 | 3 | 9 | 6 | 4 | 4 | 6 | 4 |
| % actions over 3000 feet or 0.06 Mach or 30 degrees | 43% | 33% | 30% | 34% | 37% | 38% | 45% | 37% | 39% | 41% |
| % clearances with multiple instruction types | 84% | 65% | 70% | 75% | 81% | 81% | 52% | 61% | 87% | 82% |

TABLE II. Runs 11-20 - 2.5 hour runs based on increased Sector Load

instruction to this aircraft earlier; it did not due to the congestion problem mentioned above.

- Two aircraft enter the sector at the same time on the same flight level (Run 3, 16, 18 and 15): In run 3, 16 and 18 one of the aircraft receives a climb and/or speed instruction immediately after entering the sector. However, this is not enough to maintain separation. This is again due to the model using a set rate for climb in trajectory predictions. Further, in run 15 no instructions are issued to solve the conflict between these two aircraft, this is due to the congestion problem described above.

### *B. Number of clearances*

It has been important to measure the number of clearances issued as an assumption has been made that no data link is available and all clearances need to be communicated by voice without overloading the radio channel. Based on prior experiences from air traffic controllers a threshold of six clearances per minute was set for all voice communications. In all runs this conditions was met, indicating that it would be possible to issue the instructions by voice in the scenarios tested.

### *C. Track miles*

We compare runs on the same scenario with and without the AAP. In particular, the number of track miles flown when the AAP AI model is issuing instructions is compared with a scenario wherein no instructions were issued. The results show that when the AAP is controlling the flights there is approximately the same number of track miles flown as when there are no instructions issued. This shows that the AAP model does not add a large number of track miles in order to achieve separation. In a few instances we observe that the number of track miles are lower when using the AAP: this is because when the AAP has issued a heading instruction, it can give a direct route clearance to a point further along the flight plan after the conflict is solved. In future versions of the AAP it may be investigated to use shortcuts (i.e., skip waypoints) with the goal of reducing track miles even when no conflicts have been detected. Currently the AAP only explores the possibility of skipping waypoints when attempting to bring an aircraft back to the original flight plan after the aircraft has been issued a heading clearance.

### *D. Type and magnitude of instructions*

There were roughly 40% instructions associated with quite a large delta compared to the current trajectory, at least 3000 feet altitude change, speed change of Mach0.06 and/or 30 degree change in heading. This is due both to the fact that currently the model is not allowed to give clearances to flights that are outside of the sector and for flights that are in the sector it uses a 10 minute time horizon for searching for conflicts. Because of this, instructions are sometimes given with short notice and therefore sometimes results in quite a large delta being required to maintain separation. We believe that this can be solved by the model searching for conflicts earlier than what it does today.

### *E. Illustrative Scenarios*

We demonstrate three scenarios from several experiments that were conducted using the Narsim simulator where the decisions of the AAP are elaborated.

*1) Scenario 1: Simple Speed Change:* The first scenario illustrated in Figure 7 shows two aircrafts LFV050 and LFV042 heading into the sector. The AAP determines a loss of separation nine minutes into the future and as soon as LFV042 enters the sector, it issues a slight increase of speed to Mach 0.78 (by 4 knots). This results in a conflict resolution.

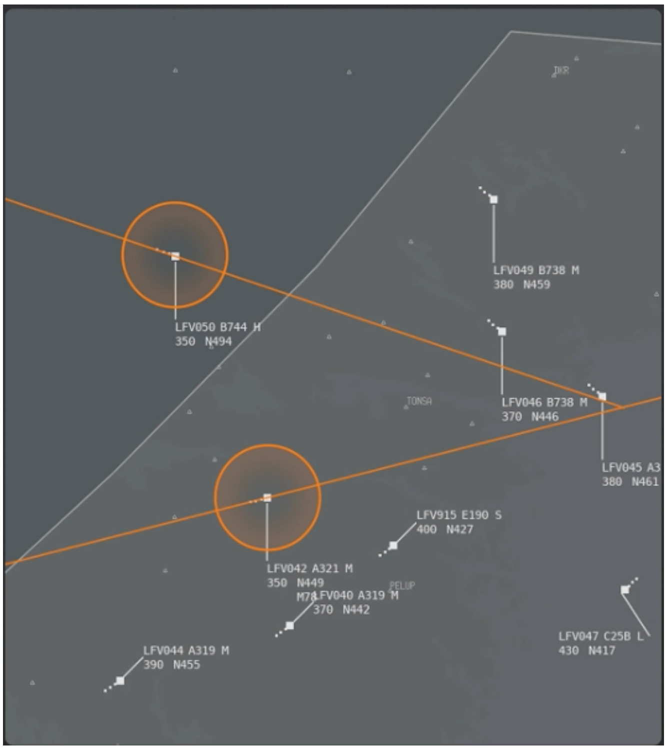


Figure 5: Scenario 1: Speed change

*2) Scenario 2: Multiple changes at once:* The second scenario involves a potential loss of separation between three aircrafts, LFV915 (entered from South West), LFV900 (entered from North), and LFV911 (entered from East). The orientation of the scenario images are North is to the top of the page. The AAP determines loss of separation 10 minutes in advance and provides a clearance of increased speed and turn five degrees to the right to LFV900 to avoid a conflict with LFV915. This ensures that the separation between these two aircrafts goes from 6.1 nautical miles to 13 nautical miles. At the same time, LFV911 is issued a climb to 41000 feet to avoid a conflict with LFV915. Once the conflicts are resolved, LFV900 is issued a clearance to its next waypoint (in the original route). This demonstrates a successful multiple actions to multiple aircrafts to avoid a conflict involving three aircrafts as well as the AAP's ability to return the aircraft to its original route.

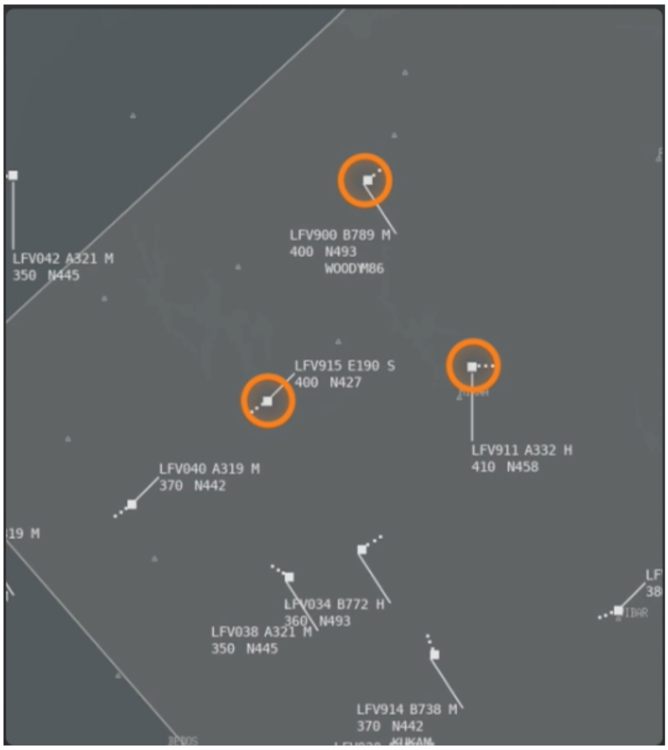


Figure 6: Scenario 2: Multiple changes at once

*3) Scenario 3: Multiple clearances:* Finally, the third scenario showcases what happens when the instructions are not followed by the pilots. In this case, LFV957 has a potential future loss of separation with LFV005. LFV957 is given a clearance by the AAP to turn by five degrees. However, the pilot only turns two degrees, which increases the predicted minimum separation from 2.9 nautical miles to 4.4 nautical miles. This is still below the minimum allowed separation of 6.1 nautical miles. The AAP identifies the situation and issues a further clearance of turn five degrees to the right, again the pilot only turns by two degrees. Even though this increases the predicted separation to 5.8 nautical miles, it is not enough to achieve the minimum allowed separation. The AAP finally turns LFV957 by a few degrees to achieve the required buffer. Note that the AAP acts on LFV957 only as LFV005 was outside the sector at the time of taking actions on LFV957.

In the illustrations, we only show the moments when separation losses are resolved and not the entire sequence of actions. The reader can view the entire video on YouTube for a detailed explanation of the above scenarios.

## VI. Related work

Artificial Intelligence and Machine Learning within Air Traffic Management is a highly active research domain. Many papers have been published in the last years on how to optimize the air transportation management using novel AI/ML techniques. In [10], the authors describe how reinforcement learning is used to resolve two dimensional conflicts in a way that encaptures the preferences of the air traffic controller. There are several other approaches similar to [10] that use


Figure 7: Scenario 3: Multiple clearances

reinforcement learning [2]–[5]. However, our approach is based on trust in the AI models with high explainability [15] as to why a particular action was suggested and how it will result in the resolution of conflicts. This is a fundamental requirement that is not addressed by other approaches.

The European research project SESAR [1] is funding AI/ML related projects in order to prepare and gradually transition to a more advanced and efficient aviation. EASA - the European Union Aviation Safety Agency has initiated the so called Artificial Intelligence Roadmap [6] to ensure a safe approach to introduce novel AI/ML techniques into the aviation sector, and to tackle the challenges that lies within the future work in this domain.

"The Artificial Intelligence Roadmap establishes the Agency's initial vision on the safety and ethical dimensions of development of AI in the aviation domain". This is very much the basis for our trusted and explainable AI models.

In addition there are collaborations around the world between universities, research institutes and government entities setting the pace and the goals for the development of future techniques and technologies. One example is The Flexible Automation project, F AUTO, which is a cross domain research project which explores the use of flexible automation in today's and future control room environments to enhance human-automation collaboration. Involved partners are the Air Navigation Services of Sweden and Linköping University for the Air Traffic Control (ATC) domain, Uppsala University and The Swedish Transport Administration for the train control domain, and Linnaeus University and The Swedish Maritime Administration for the Vessel Traffic Service (VTS) domain.

## VII. Conclusions and Future Work

Given the task of managing aircraft separation in an en-route sector the following conclusions were made on the performance of the AAP:

The AAP is capable of autonomously issuing clearances to aircraft while flying through the sector and to do so without overloading the radio frequency and/or adding a large amount of track miles. In simulation runs, there have been some separation losses, our analysis indicates that this can be explained by a set rate of climb being used for trajectory predictions and problems with congestion.

In the future this could be avoided by adjusting the aircraft performance data used or through access to the actual aircraft data. Further, the congestion problems can be solved using a plethora of measures. The AI model may evaluate multiple actions in parallel or reduce the search space by investigating single dimensional actions first to lessen or even avoid congestion. The driving Streams application can optimize the critical radar track path to reduce backpressure faster and it may suppress radar track data sets from being sent to the AI model to immediately realign the application with current data after congestion resolves.

An important aspect of future work involves the exploration of the time dimension. The current model uses a fixed horizon of 10 minutes when searching for future conflicts. Within this time horizon, the model proposes clearances that are immediately acted upon. Future work should explore the ability of taking actions at a certain time in the future to enable further optimization. This would avoid creating near term conflicts while addressing future conflicts in a timely manner.

Other factors that should be considered in the future are inclusion of weather parameters such as convective winds and no-fly zones. To bring back the flights that have been acted on by the Autoplanner to its original flight plan should also be considered in future work. Another aspect is to ensure that actions can be taken when aircraft enter a sector with separation loss and when aircraft loose separation in the sector. It should also be considered to include issuing of direct routes and to adjust the input regarding ranking of actions and which actions that can be used in the same clearance.

Future work should also explore the possibility to incorporate active learning to improve ranking of actions. This can be done through feedback accrued over long periods of many flight hours. Deep learning approaches could be explored to learn hidden factors in decisions made by ATCOs and use their rich experience to improve the suggested actions. Finally, human-in-the-loop simulations should be performed, where controllers issue the instructions that the model proposes.

## Acknowledgements

This work has been performed by LFV (Sweden's leading provider of air traffic control and associated services for civil and military aviation) and IBM as part of a joint research project, Advanced Autoplanner, which was funded by the Swedish Transport Administration. We would like to extend special thanks to: Johan Timius and Vidar Skreden (LFV air traffic controllers) for providing detailed feedback on air traffic control scenarios, safety criterion, operating practices and feedback on the performance of the AAP, Norbert Schultz (IBM) and Daniel Johansson (LFV) for designing and setting up the IBM Streams infrastructure for AAP and Mikael Björk (LFV) and Fredrik Bennison (LFV) for setting up Narsim for AAP and Robin Hughes (LFV) and Michael Broekman (IBM Cloud) for identifying this opportunity and initial problem formulation. A more detailed evaluation can be found in [16].

## References


[1] Crespo, Antonio Marcio Ferreira and Weigang, Li and de Barros, Alexandre Gomes, Reinforcement learning agents to tactical air traffic flow management, International Journal of Aviation Management, Vol 1, No 3, pp 145–161, 2012.

[2] Agogino, Adrian and Tumer, Kagan, Regulating air traffic flow with coupled agents, In Proc. of 7th Intl. Joint Conference on Autonomous Agents and Multiagent systems, Vol 2, pp 535-542, 2008.

[3] Brittain, Marc and Wei, Peng, Autonomous Air Traffic Controller: A Deep Multi-Agent Reinforcement Learning Approach, arXiv preprint arXiv:1905.01303, 2019.

[4] Abdoos, Monireh and Mozayani, Nasser and Bazzan, Ana LC, Traffic light control in non-stationary environments based on multi agent Q-learning, In Proc of 14th International IEEE conference on intelligent transportation systems (ITSC), 2011, pp 1580-1585.

[5] Van der Pol, Elise and Oliehoek, Frans A, Coordinated deep reinforcement learners for traffic light control, In Proceedings of Learning, Inference and Control of Multi-Agent Systems (at NIPS 2016)

[6] Airspace Management and Advanced Flexible Use of Airspace. https://www.sesarju.eu/node/2160

[7] Trajectory-based Operations. https://www.sesarju.eu/sesar-solutions/trajectory-based-operations

[8] Enhanced Tactical Conflict Detection and Resolution Services and Conformance Monitoring Tools for Enroute. https://www.sesarju.eu/sesar-solutions/enhanced-tactical-conflict-detection-resolution-cdr-services-and-conformance

[9] MAHALO: Modern ATM via Human/Automation Learning Optimisation. http://mahaloproject.eu/

[10] Phu N. Tran, Duc-Thinh Pham, Sim Kuan Goh, Sameer Alam and Vu Duong. An Interactive Conflict Solver for Learning Air Traffic Conflict Resolutions. In Journal of Aerospace Information Systems, 2020. https://arc.aiaa.org/doi/10.2514/1.I010807

[11] High Performing Aviation for Europe. https://www.sesarju.eu/

[12] Human-centric Approach to AI in Aviation. https://www.easa.europa.eu/ai

[13] NLR Air Traffic Control Research Simulator Tower (NarSim). https://www.nlr.org/capabilities/nlr-air-traffic-control-research-simulator-narsim-tower/

[14] LFV and IBM investigate the use of Artificial Intelligence for Swedish air traffic control. https://lfv.se/en/news/news-2020/ai

[15] Vijay Arya and Rachel K. E. Bellamy and Pin-Yu Chen and Amit Dhurandhar and Michael Hind and Samuel C. Hoffman and Stephanie Houde and Q. Vera Liao and Ronny Luss and Aleksandra Mojsilović and Sami Mourad and Pablo Pedemonte and Ramya Raghavendra and John Richards and Prasanna Sattigeri and Karthikeyan Shanmugam and Moninder Singh and Kush R. Varshney and Dennis Wei and Yunfeng Zhang, One Explanation Does Not Fit All: A Toolkit and Taxonomy of AI Explainability Techniques, https://arxiv.org/abs/1909.03012, 2019

[16] Sofia Rydell, Advanced Autoplanner Final report, 2021-03-24, https://fudinfo.trafikverket.se/fudinfoexternwebb/pages/PublikationVisa.aspx?PublikationId=4600