Integrated State of the Art Report

Deliverable ID:	D2.1
Dissemination Level:	Public
Project Acronym:	MAHALO
Grant:	892970
Call:	H2020-SESAR-2019-2
Topic:	SESAR-ER4-01-2019
Consortium Coordinator:	DBL
Edition Date:	30 September 2020
Edition:	00.02.00
Template Edition:	00.02.02

Authoring & Approval

Authors of the document

Name/Beneficiary	Position/Title	Date
Stefano Bonelli/DBL	Project Coordinator	
Matteo Cocchioni/DBL	Project contributor	

Reviewers internal to the project

Name/Beneficiary	Position/Title	Date
Brian Hilburn/CHPR	WP2 Leader	
Carl Westin/LiU	WP6 Leader	
Francesca Margiotta/DBL	Quality Manager	
Erik-Jan van Kampen/TUD	WP5 Leader	
Clark Borst/TUD	WP4 Leader	
Supathida Boonsong/LFV	Project contributor	
Magnus Bång/LiU	WP3 Leader	

Approved for submission to the SJU By — Representatives of beneficiaries involved in the project

Name/Beneficiary	Position/Title	Date
Stefano Bonelli/DBL	Project Coordinator	30/09/2020

Rejected By - Representatives of beneficiaries involved in the project

Name/Beneficiary	Position/Title	Date

Document History

Edition	Date	Status	Author	Justification
00.00.01	01/08/20	Draft	Brian Hilburn	Proposed table of content

00.00.02	15/08/20	Draft	Carl Westin, Erik-Jan van Kampen, Clark Borst, Supathida Boonsong, Magnus Bång, Tiago Monteiro Nunes	All sections drafted
00.00.03	31/08/20	Internal review	Brian Hilburn	Internal peer review #1
00.00.04	04/09/20	Quality check	Francesca Margiotta	First quality check
00.00.05	11/09/20	Post 1st review draft	Carl Westin, Erik-Jan van Kampen, Clark Borst, Supathida Boonsong, Magnus Bång, Tiago Monteiro Nunes	Addressed minor changes from internal peer review #1 and first quality check
00.00.06	15/09/20	Internal review	Carl Westin, Erik-Jan van Kampen, Clark Borst, Supathida Boonsong, Magnus Bång, Tiago Monteiro Nunes	Internal revision
00.00.07	18/09/20	Internal review	Magnus Bång, Clark Borst, Supathida Boonsong	Document reviewed and harmonised.
00.00.8	23/09/20	Internal review	Carl Westin	Internal peer review #2
00.00.9	24/09/20	Quality check	Francesca Margiotta	Second quality check
00.00.10	25/09/20	Post 2nd review draft	Brian Hilburn	Addressed minor changes from internal peer review #2 and first quality check.
00.00.11	28/09/20	Consolidated version	Carl Westin, Erik-Jan van Kampen, Clark Borst, Supathida Boonsong, Magnus	Document reviewed and harmonised. Ready for PC approval

			Bång, Tiago Monteiro Nunes	
00.01.00	30/09/20	Final release	Stefano Bonelli	Deliverable approved for submission to the SJU

MAHALO

Modern ATM via Human / Machine Learning Optimisation

This project has received funding from the SESAR Joint Undertaking under the European Union's Horizon 2020 research and innovation programme under grant agreement No 892970.

Abstract

This document is the deliverable 2.1 that presents an integrated report on the output of the human performance and ML reviews, highlighting the latest theoretical and empirical work into each. The document integrates the outputs of two Tasks T2.1 and T2.2, respectively *State of the Art review, human performance* and *State of the Art review, ML*.

Several topics are analysed, from a general overview of the main activities performed by a modern ATCO and the best Conflict Detection and Resolution strategies, to the importance of integrating Machine Learning in future normal operations and the pivotal role of MAHALO Project in exploring the real possibilities in realising and applying such a technology. Furthermore, other important questions about human performance are reviewed, in terms of conformance and transparency.

The lesson learnt from this review that can be applied to the MAHALO project over the coming months is also summarised.

Table of Contents

1. Introduction

1.1 Context

Looking beyond the current COVID pandemic, long term forecasts predict continued growth in global air traffic. To meet this demand, the air traffic control (ATC)[1] system of the future will require greater application of more sophisticated automation, both in terms of system authority and autonomy, and in terms of control tasks. Generally speaking, ATC involves such tasks as: monitoring air traffic, detecting conflicts, making data entries, devising solutions, implementing solutions (by interacting with aircraft or other actors), revising plans and strategies, etc.

Until now, automation has generally only been applied to the simpler, routine tasks among these [Claudatos et al, 2018], such as processing flight data, checking flight plans, etc. For roughly 70 years, the so-called *Fitt's List* has served as a guide to function allocation between human and machine, based on the relative strengths and weaknesses of each. For example, humans excel at tasks involving improvisation and inductive thought, whereas machines are superior at tasks requiring speed, power, and computational strength. In recent years, the Fitt's List has fallen out of favor, as technological advances have begun blurring the line between the types of tasks that can be done by humans and machines.

One of the main challenges faced by the ATC community in the coming years is how to best develop and deploy new forms of automation incorporating Artificial Intelligence (AI) [SESAR, 2019; EAAIHLG, 2020; EASA, 2020]. Recent advances in AI, in particular Machine Learning (ML), present both a considerable opportunity and an enormous challenge to meeting these demands. ML offers the possibility that future ATC systems will be able to sense, learn, and act autonomously, and will be increasingly able to take over much of the cognitive work involved in separating and expediting air traffic [Kistan et al, 2018]. ML methods are developing at an explosive rate, and are being deployed in new domains seemingly daily. However, the potential capabilities of ML must be weighed against some clear challenges.

These challenges stem in part from the fact that ATC problems are often complex and ill-defined. ATC represents a large "solution space" in that air traffic control accommodates various successful

[1] This report focuses on ATC, but will also refer to *Air Traffic Management* (ATM), which refer to the broader set of air- and ground-based functions (including ATC services, airspace management, and flow management).

strategies within the basic system constraints. As long as aircraft are kept separated, controllers can differ in their overall strategies and individual solution choices.

Automation is something of a double-edged sword. It has often been said that automation excels in tasks that are 'dirty, dumb, or dangerous.' From manufacturing robotics, to hazardous material handling, to rail transport [Balfe et al, 2014], automation has provided clear benefits in terms of performance, safety, and efficiency. However, often-cited potential human performance costs include: transient workload peaks; situation awareness problems; return-to-manual control difficulties; and others [Manzey et al, 2012]. Increasingly, however, as the capabilities of automation approach those of the human, and machines can increasingly take over many of the 'thinking' parts of jobs like ATC, we are forced to consider how issues of trust, acceptance, and reliance will be impacted [(c.f. section 5); Lee & See, 1992; Kistan et al, 2018; Mekdeki, 2015; EAAIHLG, 2020].

1.2 Project scope and objectives

MAHALO focuses on a particular application of advanced automation that seems likely to change the nature of the controller's job in coming years: namely, conflict advisory (or decision support) automation, which provides the controller **real-time assistance with conflict detection and resolution.**

Given that the future ATC system will likely rely on AI approaches for designing such advanced automation [e.g. EAAIHLG, 2020], the question emerges of how we should be developing systems like this. The basic research questions behind MAHALO can be stated as follows:

In the emerging age of machine learning, should we be developing automation that <u>matches</u> the performance and strategies of the human, or should we be developing automation that is <u>understandable</u> to the human? Do we need both? Are there trade-offs and interactions between the two, in terms of air traffic controller trust, acceptance, or system performance?

Notice that these research questions touch on two important concepts in human-machine system design. These concepts will be discussed in more detail in chapter 5:

- **Conformance**—the MUFASA project introduced the term strategic conformance to describe the apparent match between human and machine strategies underlying conflict resolutions [MUFASA, 2013]. The project demonstrated that conformal automation could benefit acceptance and agreement, and therefore help foster trust in, and use of, automation;

- **Transparency**—has been defined as "automation's ability to afford understanding and predictions about its behavior" [Westin et al, 2016]. One of the biggest potential drawbacks of ML methods is that their output is often 'opaque,' unintuitive, and difficult to understand. In a sense, ML can sometimes seem just a black box that learns how to associate input and output. The human operator can then have a difficult time inferring and understanding the underlying process2 that the black box used.

The concepts of transparency and conformance might **trade off**. For example, it is possible that transparency would be less essential if automation could foster acceptance and trust (and eventual automation use) by appearing to solve problems in a way that conforms to humans' general strategies, or even to an individual's specific strategy. In that case, conformance could reduce the need for transparency. But what is the overall system performance benefit of highly conformal automation, that presumably just replicates the controller's performance (whether it be 'right' or 'wrong.')? Eventually, we want automation to extend the capabilities of the human. Figure 1.1 shows how the concepts of automation transparency and conformance can vary independently. This can lead, at the extremes, to one of four outcomes. MAHALO intends to explore, via human-in-the-loop simulations, the optimal balance between transparency and conformance of an AI conflict detection and resolution (CD&R) support system.

		TRANSPARENCY	
		Low	High
CONFORMANCE	Low	• Different solution than individual • Solution not explained	• Different solution than individual • Solution is explained
	High	• Same solution as individual • Solution not explained	• Same solution as individual • Solution is explained

Fig. 1. 1. Automation conformance and transparency can vary independently.

In addressing the research questions around automation transparency and conformance, MAHALO is leveraging its collective team expertise in ML modelling, ATC operations and concepts, and display development, to conduct the activities described in the following paragraphs.

Develop a hybrid ML capability for conflict detection and resolution

[2] As discussed in chapter 5, transparency is a multifaceted construct that can refer to the understandability of different aspects, including the output product and the underlying process, of automation.

MAHALO set out to create an integrated system of two ML models, to separately address the processes of conflict detection and conflict resolution. As discussed in later chapters, this State-of-the-Art Report (SOAR) aimed in part to help confirm the initial feasibility of this concept, and to identify candidate ML methods to achieve this.

Develop a control model of ATC, and experimental User Interface

As described later in chapter 2, ATC can be described a dynamic control task in which an agent detects system disturbances, and applies compensatory action so as to maintain system behavior. By describing ATC in terms of a generic feedback loop, the tasks can be assigned to either human or machine agent. This control model framework will also help us identify candidate control inputs (e.g. heading, speed) and monitored variables (e.g., distance between aircraft, groundspeed). Finally, this control model framework will help refine the experimental User Interface that will sit atop MAHALO's hybrid ML system.

Evaluate the impacts of automation transparency and conformance

MAHALO extends the experimental methods of the earlier MUFASA project [9], which simulated advisory automation, and explored conformance, via unrecognizable replays of previous manual performance. MAHALO, again, aims to build a stable hybrid ML system for resolution advisories. This ML system will then form the testbed for experimental human-in-the-loop simulations exploring the interactive effects of automation transparency and conformance on human / machine system performance. A question is how to operationalize the conformance and transparency of a ML system.

Develop an automation design framework

Based on the output of this review, as well as the system / UI development, and human-in-the-loop trials, MAHALO aims to produce a framework document to help guide the introduction of ML into ATC systems. This will include conclusions on ML approaches to CD&R, the roles of automation transparency and conformance, as well as the impact of contextual factors such as traffic load, complexity, and off-nominal conditions.

1.3 Purpose and Scope of the State-of-the-Art-Report

1.3.1 Task 2.1: Human performance review

MAHALO task 2.1 reviewed recent theoretical and empirical contributions in the relevant areas of human performance, including:

- Explainable AI (XAI), in particular ML interpretability;
- Transparency;

- Automation strategic conformance;
- User Interface principles for fostering trust, reliance, and transparency.

Outputs from such previous SESAR Exploratory Research as MUFASA, C-SHARE, STRESS, and NINA were included, as well as ongoing ER and IR activities (e.g. TERRA and relevant RPAS CD&R research). An important element of task 2.1 was review and cataloguing of controller CD&R strategies. A preliminary list of these CD&R strategies (work is ongoing) is presented in section 3.3 of this report.

1.3.2 Task 2.2: ML methods review

In parallel, task 2.2 reviewed recent research into AI and ML methods, including:

- Recent architectures and models for ML;
- Data requirements for ML;
- Object detection and localization in ML modelling;
- Integrating ML methods; and
- Candidate ML architectures applicable to the ATC CD and CR processes.

Although the MAHALO team brings broad hands-on experience with ML methods, and their application to CD&R applications, ML is an incredibly fast-moving field. Developments into new methods and theories are made nearly every day, it seems. We were therefore especially concerned about verifying the latest literature (recent five years) into ML methods to ensure we were taking onboard the latest thinking and techniques in the field.

This effort consisted of parallel reviews into human performance (task 2.1) and ML (task 2.2), combined into a single integrated report.

1.3.3 Assumptions and research questions

The task 2.1 and 2.2 reviews were integrated into one State-of-the-Art-Report (SOAR), which integrates both the human performance- and ML reviews. The team went into the SOAR with some basic assumptions about the ConOps and methods we would use. However, we did not want to pre-judge any of our ultimate methodological decisions. Especially in the field of ML, it was important to ensure that we were taking onboard the latest knowledge from the R&D and operational communities.

ASSUMPTIONS

Some basic assumptions at the outset of the SOAR included the following:

- The controller works together with an AI system in CD&R;
- We would focus on the executive controller in single person operations;
- We would focus on the en-route domain;
- We would use TU Delft's Sector X platform for simulations;
- We would focus on tactical control;
- We would assume Controller-pilot communication via data-link;
- We assume a future environment consisting of 4DTM, where the majority of separation conflicts have been solved strategically;
- We would explore the feasibility of flight-centric ATC;
- We would consider sector size and traffic levels to be considerably larger and higher than current levels;
- We would integrate both screen (traffic) and eye tracking data as input measures;
- We would manipulate conformance and transparency as independent variables;
- We would split the CD and CR functions;
- We would evaluate the system against measures of acceptance, trust, understanding, workload, and performance; and
- We would use some sort of combined 'hybrid' ML architectures for each of these CD&R functions.

MAHALO targets a future ATM environment in line with the digital European sky vision 2040 (i.e. phase D in the digital transformation) in the European ATM Master plan [SESAR, 2019]. Important characteristic of this vision includes dynamic airspace organization, flight-centric ATC, and free-routes. Central to this vision is the increase in automation capabilities in supporting the human. Here MAHALO targets a high automation capability (level 4), where the controller receives automation support in all human information processes stages (sense and perception, analysis, decision-making, and action implementation).

The vision also comprises expectations that AI support systems will substantially alleviate controllers' workload, controllers will delegate tasks to AI systems, AI systems will propose the best options to humans (flow, sequences, safety net etc), and AI will solve complex trajectory situations using machine-to-machine communication with aircraft.

More specifically, MAHALO assumes controller and automation to collaborate in CD&R, with the automation conducting conflict detection and either executing solutions or proposing solutions, to the controller, to be accepted. A simplified version of the ATM *Levels of Automation* (LOA) framework proposed in the ATM Master Plan is shown in Figure 1.4.

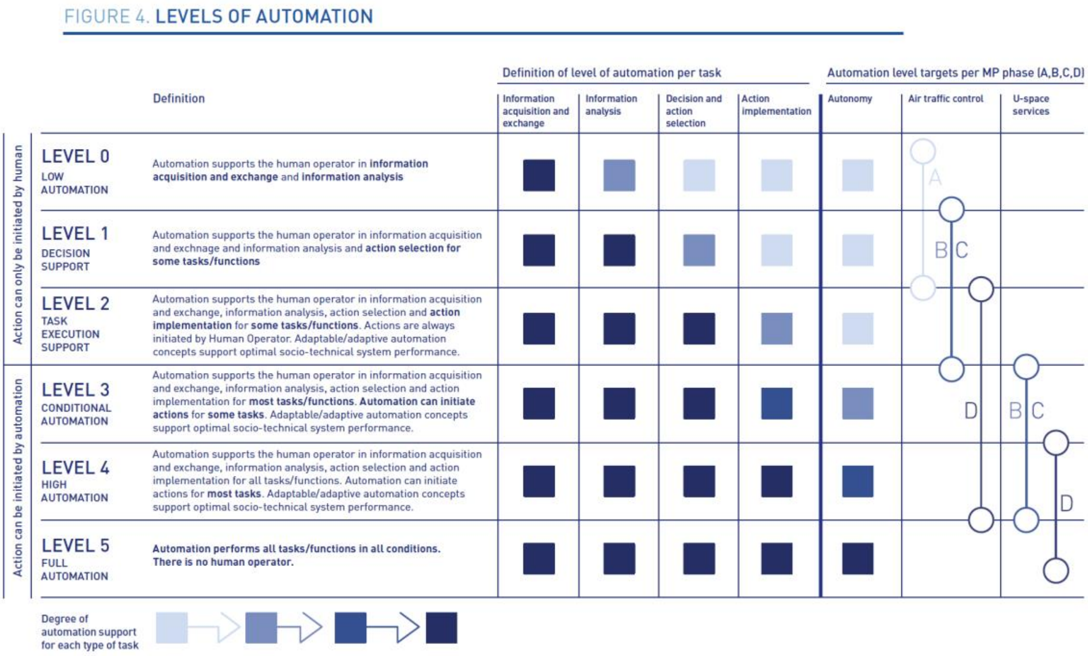


Fig 1.4 ATM Levels of Automation (LOA) framework (after SESAR, 2019).

RESEARCH QUESTIONS

The integrated SOAR was intended to feed directly into a specification of the MAHALO ConOps, traffic scenarios, UI design, experimental / simulation design, data requirements, ML architecture, and metric selection.

Specifically, the SOAR aimed to address **eight high level research questions**, that fall generally into one of these areas:

ML issues:

1. Broadly speaking, how should we model the CD&R process?
2. Which candidate ML approaches seem most promising for modelling CD&R?
3. What main challenges exist in the use of ML methods, and how can we counter these?

Assessing human and system impacts:

4. Do transparency and conformance constructs still seem critical, and how feasible are they to experimentally explore?

5. How can transparency be operationalised in simulation trials?

6. How can conformance be operationlised in simulation trials?

7. How do we assess other aspects of human performance, including trust, acceptance, and workload?

Interface issues:

8. How should we address display issues to convey transparency and conformance?

1.4 Report Structure

The report is divided into six chapters.

The first chapter contains a background and introduction to the MAHALO project and this report.

Chapter 2 introduces **ATC and the core task of CD&R from** a generic control perspective.

Chapter 3 covers our literature review of **CD&R approaches**. Following a closer look at previous CD&R reviews and proposed CD&R taxonomies for ATM and UTM (i.e. relating to Task 2.1), a detailed literature review of ML approaches to CD&R was conducted.

In Chapter 4 we provide an introduction to **ML**. Both general approaches and specific approaches to ATM are discussed. This chapter also discusses ML interpretability - the ML approach to transparency. Chapter 4 covers Task 2.2 as outlined above, except for candidate ML architectures for CD&R. These will be covered in the D2.2 Operational Concept Report.

Chapter 5 takes a closer look at the **human performance aspects** relevant to MAHALO (Task 2.1). In particular, this chapter provides an information processing perspective on ATC functions, and examines MAHALO's two core human performance constructs: automation conformance and automation transparency.

The report closes with chapter 6, which sums up the lessons learnt from this review, that can be applied to the MAHALO project over the coming months.

2. ATC: control task definition

This chapter focuses on describing the ATC task from a generic control perspective, irrespective of the type of agent (i.e., human or computer) who is performing the control task.

2.1 ATC as a feedback control system

Air Traffic Control (ATC) is a subset of Air Traffic Management (ATM) and is located at the sharp end of ATM operations, namely tactical control when aircraft are airborne. Here, the primary purpose of ATC is to **safely** and **efficiently** organize and expedite the flow of air traffic from origin to destination, and provide information and other support for pilots.

ATC can be regarded as a dynamic control task where human and/or computerized agents need to close a feedback control loop, as illustrated in Figure 2.1. In such a control architecture, the agent observes the outputs of a system and evaluates those against certain goals and targets (e.g., Key Performance Indicators expressed in terms of safety and efficiency measures). Whenever system outputs deviate from the goals, it is the agent's responsibility to provide inputs to the system such that the goals will be met. In dynamic control tasks, system outputs will continuously be affected by disturbances, which will require the agent to continuously monitor and steer the system so as to mitigate the impact of disturbances on the system's outputs. In order for any system to be controlled successfully, important prerequisites are 'controllability' and 'observability'. This simply means that all critical system states must be able to be observed by the agent and sufficient degrees of freedom in terms of control inputs must be available.

In ATC, the 'system', 'disturbances' and the specific 'goals' (e.g., specific KPI thresholds) highly depend on the airspace under consideration. This requires a brief overview of the airspace organization.

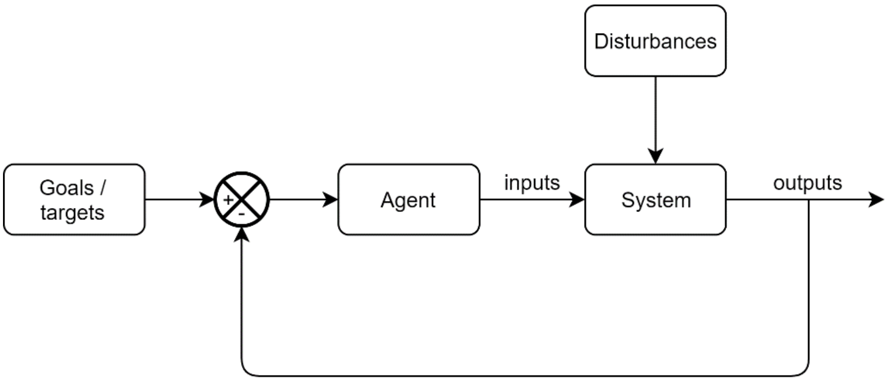


Figure 2.1: Generic feedback control loop.

2.2 System scope: airspace organization

Within a Flight Information Region (FIR), the airspace is subdivided by altitude regions into several "sectors", as shown in Figure 2.2. Each part of the airspace is the responsibility of a different team of controllers, each having different tasks. From top to bottom, the Upper Control Area (UTA) and Control Area (CTA) are controlled by Area Control (ACC). Closer to an airport, the Terminal Maneuvering Area (TMA) is handled by Approach / Departure Control (APP / DEP). In a cylindrical area around the airport, the Control Zone (CTR), Tower Control (TWR) is responsible for issuing take-off and landing clearances and monitoring runway and taxi movements.

In the UTA, ACC generally deals with en-route air traffic featuring mostly high-altitude overflights. In the CTA, ACC may also need to handle inbound and outbound traffic to and from airports. For example, in the Netherlands the CTA is handled by the Dutch Air Traffic Control (LVNL), who need to take care of lower altitude overflights, clear aircraft to higher altitudes and inbound and outbound traffic to and from Schiphol. In the Netherlands, the UTA is controlled by Maastricht Upper Area Control (MUAC) handling mostly en-route traffic flying at cruising altitude.

Each team of controllers may have different KPIs and/or KPI thresholds to take into account. For example, in the UTA and CTA, aircraft separation thresholds are commonly larger than in the TMA. In the TMA, landing intervals need to be considered, something which is outside the scope of ACC. In MAHALO, the scope is limited to ACC.

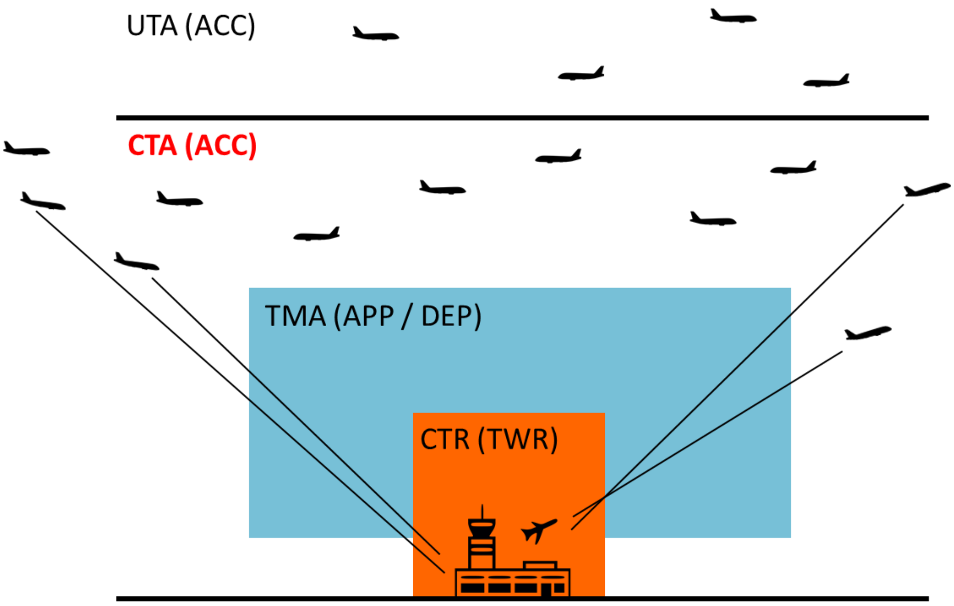


Figure 2.2: Generic airspace organization.

2.3 Goals and outputs of Area Control

In the UTA and CTA, a human or computerized agent in control of the system have several common goals to meet:

- Safety:
    - Maintain vertical and horizontal separation between aircraft, by respecting separation minima (1000 ft vertical, 5 NM horizontal)
- Effectiveness:
    - Ensure aircraft reach their navigational targets (i.e., designated sector exit waypoints)
    - Transfer aircraft to the adjacent sector(s)
- Efficiency:
    - Minimize time delays upon reaching navigational targets
    - Minimize additional flown track miles of aircraft
    - Adhere to pilot and airline company wishes and preferences

When a human agent is in full control, goals related to human performances will be included. For example, “workload management” and “maintaining situation awareness (SA)” are important goals to take into account for air traffic controllers (Westin et al., 2016), who tend to control the system in such a way that it results is more predictable traffic patterns that are easier to monitor and regulate. By doing so, efficiency targets are commonly loosened at the benefit of lowering workload and increasing situation awareness.

When a computer is in full control, workload and situation awareness (SA)constructs are irrelevant and thus higher system efficiency can be achieved. Note that this is also the point where human and computerized agents will differ (i.e., ‘optimize’ vs. ‘satisfice’), which has shown to contribute to acceptance problems when humans need to *collaborate* with computerized agents (Westin et al., 2016).

2.4 Disturbances

Although airspace use and route-allocation will be structured and optimized before flights are airborne, to achieve optimal system performance in terms of safety, efficiency and productivity, it is the unforeseen separation provisions, sequencing, weather and changing airspace constraints which inevitably require (small, tactical) changes in the pre-planned trajectories, see Figure 2.3.

ATC is considered as an ‘open’ system, which is subjected to uncertainties that cannot always be predicted beforehand. Unexpected sector entry delays (e.g., due to delayed departures or control actions by agents in the neighboring sector) may result in predicted separation losses (i.e., conflicts) that need to be resolved. Avoiding unexpected weather cells will require aircraft to be re-routed, resulting in additional flown track miles and rerouting delays if aircraft speeds remain unchanged. The probabilistic nature of the ATC environment therefore makes it a dynamic control task where competing goals may change over time. This requires agents to *adapt* to new and unexpected circumstances and come up with *creative* solutions to meet system goals.

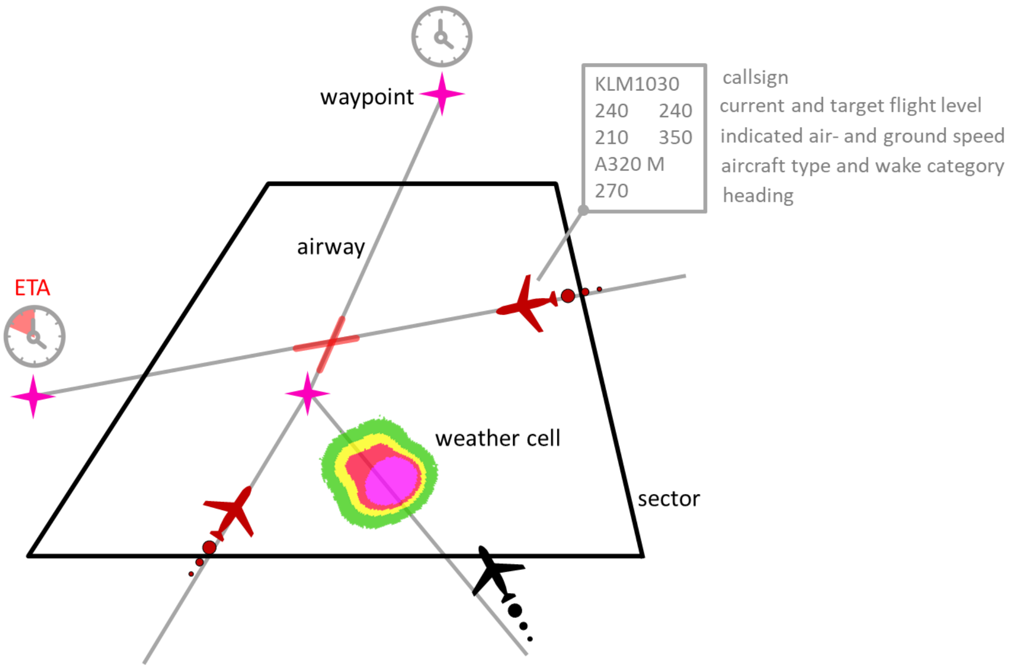


Figure 2.3: Common disturbances found in area control.

2.5 Control inputs

When facing system disturbances, the agent needs to be able to provide inputs to the system elements that will redirect the system's outputs. In ATC, an agent can only influence the system by changing the states of each individual aircraft. In its most succinct form, the agent basically has three control inputs (i.e., clearances) to give to aircraft: altitude, heading and/or speed (see Figure 2.4).

Depending on the sector organization, different clearance types may be prioritized due to physical constraints in terms of aircraft flight envelopes and performance capabilities. For example, in the UTA where aircraft are flying at cruising altitudes, altitude clearances are prioritized over speed clearances due to narrow speed envelopes and slow acceleration and deceleration profiles. Additionally, speed changes are not encouraged as they will require aircraft to deviate from their ideal cruising speeds, which will negatively impact fuel consumption and therefore airline cost models.

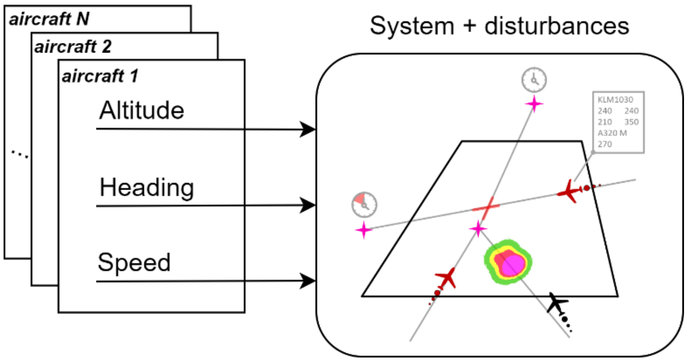


Figure 2.4: System inputs, which are constrained by aircraft performance capabilities.

2.6 Closing the loop: observability, controllability and stability

To be able to successfully close the control loop, human and computerized agents will need to be able to observe the state of the system (i.e., observability) and be able to have sufficient degrees of freedom to influence the system (i.e., controllability). In closing the ATC control loop, human and automated agents therefore need to be able to *observe*:

- the movement of air traffic
- the states of individual aircraft
- the type of aircraft (e.g., manufacturer)
- the controlled sector geometry
- vertical and horizontal distances between aircraft
- location of navigational targets
- procedural compliance
- atmospheric conditions (wind fields, weather cells, etc.)

Observing the above-mentioned system state is facilitated by several sensors that acquire information:

- Primary (PR) and secondary surveillance radars (SSR), acquiring:
  - aircraft ID (SSR Mode A)
  - callsign (SSR Mode S)

    - type and wake turbulence category (SSR Mode S)
    - Horizontal position (PR)
    - Flight track (PR)
    - altitude (SSR Mode C)
    - groundspeed (PR)
    - indicated airspeed (SSR Mode S)
    - Mach number (SSR Mode S)
- Automatic Dependent Surveillance System-Broadcast (ADS-B), providing all of the above-mentioned states including some additional information, such as:
    - Pilot's Mode Control Panel (MCP) settings (e.g., autopilot targets)
    - Flight Management System (FMS) outputs, such as "next waypoint" of the planned trajectory and Estimated Arrival Times (ETA)
- Meteorological services, providing:
    - Wind fields (wind speed and direction) per 2D location and altitude layer
    - Weather forecasts

In terms of controllability, human and automated agents can only control the states of individual aircraft by issuing altitude, heading and/or speed clearances over time. One important aspect that determines the stability of closed-loop control is *time delay*. In ATC, it is common that system states are updated every radar update. Updates may range between five to ten seconds, depending on the specific radar installation. For surveillance data published over ADS-B, updates may be available every second.

To successfully control a system that involves time delays between consecutive state observations, inter-related constructs such as *preview* and *feedforward* are important. Preview is defined as the provision of prior information about future reference changes. This includes, amongst others, knowledge about aircraft intent (e.g., flight plan data) and predicting where aircraft will end up after a certain amount of time (e.g., state-based extrapolation). Feedforward relates to accounting for disturbances before they have time to affect the system. For example, provisionally rerouting aircraft trajectories before a weather cell has manifested. Finally, feedback is defined as corrective actions based on the actual observed system response.

In ATM operations, feedforward is commonly part of flow and airspace management where aircraft routes are planned years, months, weeks and/or days before actual flight and thus before disturbances have manifested. Preview is mostly part of tactical ATC, for example in the core task of CD&R that aims at preventing loss-of-separation events before they have taken place.

2.7 Implications for MAHALO

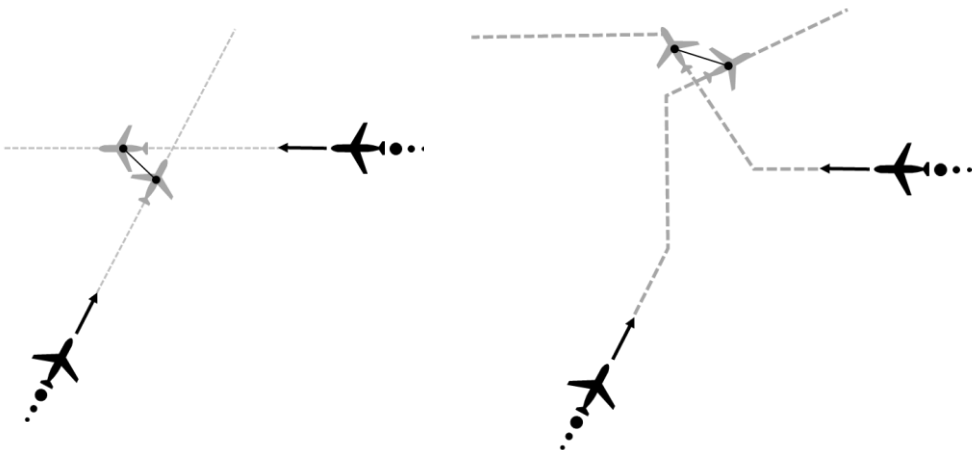
This chapter described the ATC task as a generic control task. This is an important foundation, as it helps describe how human and machine might share a common control architecture and system goals, and allow ML design to focus on the different strategies each uses to close the control loop. The foll0wing chapter 3 will review our literature review into human (section 3.1) and machine (section 3.2) approaches to CD&R, and how they differ.

3. CD&R Methods

The purpose of this chapter is to review and **compare the CD&R methods used by humans and machines, to obtain an overview of past and current approaches**. This aims to help MAHALO build a ML system capable of working with a human in carrying out CD&R. Of particular interest in our current review were AI and ML approaches to CD&R. Therefore, a state-of-the-art literature review was conducted with the objective to identify recent approaches for solving the CD&R problem with the help of AI and ML. Notice that in some cases, theoretical and empirical work has been done into generic CD&R approaches, divorced from the agent. That is, some approaches apply to either human or machine. The output from this review will be used to obtain lessons for the MAHALO project and guide the development of a ML CD&R system

CD&R is one specific, yet core, task in ATC. It serves the main purpose of meeting the system's safety goals in terms of maintaining separation between aircraft well before a loss of separation has occurred. Note that when separation has been lost, pilots are responsible for avoiding mid-air collisions by following Traffic Collision Avoidance System (TCAS) instructions.

Conflict detection (CD) involves predicting a loss of separation occurrence (either horizontal or vertical) ahead of time, making it a control task relying on preview. It can rely on state-based extrapolations and/or taking into account aircraft intent (i.e., planned trajectories), see Figure 2.5. Aircraft are considered to be in conflict when the predicted Closest Point of Approach (CPA) is less than the radius of the protected zone (5 nm horizontally) and less than 1,000 ft in altitude. The urgency of the conflict is determined by the time at which the CPA will be reached.



(a) State-based extrapolation (b) Intent-based extrapolation

Figure 2.5: Conflict detection

Conflict resolution (CR) involves the action that is undertaken to solve a conflict. In tactical control, this is done by issuing one of more control inputs (altitude, heading and speed) to one or multiple aircraft involved in the conflict. Here, humans generally tend to give one command to one aircraft at the time, whereas computers could issue several commands simultaneously to multiple aircraft.

Section 3.1 takes a closer look at evidence on how human air traffic controllers achieve CD&R. previously proposed CD&R taxonomies in the literature. Section 3.2 details the method for the review of AI/ML approaches to CD&R. Section 4.3 then presents the results of this review activity by means of a CD&R taxonomy and overall discoveries.

3.1 Human CD&R approaches

There is a rich body of past research, including a few recent reviews, into the strategies and specific actions controllers apply in CD&R. Some of the main sources retrieved include:

- Fothergill & Neal, 2008;
- Hao, 2018;
- Kirwan & Flynn, 2002;
- Kuchar & Yang, 2000;
- Pelegrin & d'Ambrosio, 2020;
- Rantanen & Nunes, 2005;
- Rantanen & Wickens, 2012;
- Ribiero et al, 2020;
- Seamster et al, 1993;
- Van Dam, 2009; and
- Yang, 2019.

Literature suggests that CD and CR generally involve

- pairwise comparisons between aircraft
- time scale of detection (i.e., how much preview in terms of look-ahead time is adopted?)
- uncertainty in current and predicted aircraft states
- time of resolution (i.e., when to take action (proactive vs reactive))
- type of actions (altitude, heading and/or speed)
- direction of action (make aircraft fly up/down, left/right, faster/slower)
- resulting traffic pattern and impact on efficiency goals (e.g., additional track miles, delays, etc.)

Table 3.1 captures our provisional list of controller CD&R strategies. This list is a living document, and will be expanded over the coming months, in step with Work Package 3 activities.

Category	Principle	Description
High-level principles	Safety	Make it safe first, before going further [with additional considerations etc.]
		Need to have a fail-safe plan B
		Anticipate that things can deteriorate (uncertainty)
		Handle the emergency first – everyone else can wait
	Workload management	Keep it [the resolution] simple
		Minimise the number of aircraft to move
		Solve easy conflicts first
		Look for the one key action that resolves the problem
		Prefer resolutions which require less co-ordination
		Leave the over-fliers alone
		Identify conflicts pairwise
		Reduce complexity
	Efficiency	Penalise the one that needs something (leave alone the ones in steady state)
		Inconvenience least people
		Minimise the penalty for aircraft
		Change in line with aircraft intentions
		Give initial (level) change early on and then fine-tune later
Conflict Detection	Attention	Focus visual attention on crossing points and sector borders
		Maintain a consistent scanning pattern
		Adopt a look-ahead time between 5 and 10 minutes
		Scan traffic in small sector area, then expand
		Scan sector in regions
		Search for closest aircraft pairs, or dense traffic areas
		Adopt circular scan pattern (clockwise, counter clockwise, or spiral)
	Strategy	1. Compare aircraft altitudes
		2. If same altitude, compare flight directions (e.g., head-on vs. crossing conflicts)
		3. If same altitude and direction, compare speeds (e.g., overtaking conflicts)
	Contextual factors	Determine urgency and priority by estimating relative speeds and distances
		In low workload conditions, wait and see before taking action
		In high workload conditions, act immediately after detecting a conflict
Conflict Resolution	Attention	First solve conflicts pairwise and later check for consequences on other traffic
		Select resolution that requires least amount of monitoring and coordination
		Select resolution that requires least amount of sector disruption
	Strategy	1. Prefer level changes; Try to keep aircraft at the same levels
		2. vector aircraft; lock aircraft on headings when using vectors
		3. Use speed solutions last (at cruise, speed enveope is only 10-20 kts)
	Contextual factors	Turn slower aircraft behind (in order to minimise extra distance flown)
		Put aircraft behind, rather than through the middle
		Solve the head-on conflicts first
		Turn faster aircraft direct to route, so it leaves sector before slower one on same route

Table 3.1 Some controller CD&R strategies.

Over the years, several CD&R methods have been developed. Thereto, multiple reviews have been conducted and CD&R taxonomies suggested. These have, however, not specifically focused on AI approaches for solving the CD&R problem. A seminal work in this context is that of Kuchar and Yang [2000], who created a taxonomy of CD&R that distinguished six basic CD&R functions:

1. state dimensions considered (horizontal, vertical, or both);
2. state propagation method (nominal, probabilistic, or worst-case);
3. conflict detection threshold;
4. conflict resolution method (prescribed, optimised, force field, or manual;
5. maneuvering dimensions (speed, heading, vertical, or combinations); and
6. number of aircraft managed (pairwise or global).

Among the 68 CD&R methods and systems reviewed, Kuchar and Yang did not find a superior method. They identified several weaknesses with the methods reviewed, such as neglecting uncertainty, ability to handle system degradations or dilutes, and computational limits. They also noted issues of pilot and controller acceptance of CD&R systems.

Ribero et al. [2020] recently extended previous taxonomies via an extensive review of CD&R methods in both manned and unmanned aviation. Over one hundred CD&R methods were reviewed. A taxonomy was proposed that classifies CD&R algorithms in ten categories divided across conflict detection and conflict resolution.

For conflict detection methods, three categories were identified:

1. the type of surveillance (centralised, distributed, or independent),
2. trajectory propagation (state-based or intent-based), and
3. predictability assumption (nominal, probabilistic, or worst-case),

For conflict resolution methods, seven categories were identified. The fifth category, the level of control, distinguished between centralised and distributed methods. The following six categories were organised as sub-categories to the level of control category:

1. the conflict resolution method (centralised: exact of heuristic; distributed: prescribed, reactive, or explicitly negotiated),
2. approach to multi-actor (>2) conflicts (centralised: sequential or concurrent; distributed: pairwise sequential, pairwise summed, or joint solution),

3. the timescale on which avoidance planning takes place (centralised: strategic of tactical; distributed: escape/collision avoidance),

4. manoeuvre employed for resolution (centralised: heading or speed; distributed: vertical or flight plan),

5. obstacle types (centralised: static or dynamic; distributed: all), and

6. optimisation objective (centralised: flight path or flight time; distributed: fuel/energy consumption).

Of particular interest are the different conflict resolution methods applied. Exact and heuristic algorithms are used by a centralised CD&R system that strives to find the best global solution. Prescribed, reactive, and explicitly negotiated are categories of algorithms used in a disturbed CD&R system where solutions depend on the aircraft involved in the conflict. Prescribed algorithms derive resolutions from pre-defined rules. Reactive algorithms trigger solutions based on the intruder's position and the conflict geometry. Ribeiro states that a common principle is to find the 'shortest way out'. The Explicitly negotiated is the only category of algorithms that determine a solution based on a negotiation between the involved aircraft (an example is TCAS). Among all CD&R methods reviewed, Ribeiro found that about a third of the method applied conflict resolution methods that does not fit into above categories (e.g, by organising traffic flows).

From Ribeiro et al [2020], the following insights are provided: the majority of methods reviewed were found to address tactical CD&R, distributed control, using a nominal predictability (i.e. does not consider uncertainty). The majority of methods were also limited to heading manoeuvres. Most approaches considered dynamic objects (e.g., other aircraft). For predicting the trajectory of aircraft, most approaches made use of state-based linear interpolation. Notably is that only three approaches considered all three tactical conflicts resolution manoeuvre available: heading, vertical, and speed.

The paper did not distinguish between AI approaches to CD&R and conventional algorithms.

In contrast to the previous reviews, Pelegrín and d'Ambrosio [2020] reviewed CD&R methods originating in the Operations Research community and proposed a taxonomy for classifying CD&R methods from a mathematical programming perspective. The framework comprises nine categories: dimensions (2D or 3D); motion (single line or multiple lines connected by waypoints); instantaneous (if maneuvers are initiated directly when the time horizon starts or not); separation constraints (see below); resolution maneuvers (speed, heading, several flight levels); trajectory recovery (yes or no); in what way the formulation is discretized (time, space, maneuvers); type of mathematical programming model (Mixed-Integer Quadratic Programming MIQP; Mixed-Integer Nonlinear Programming MINLP; Mixed-Integer Programming MIP; MIQP + Mixed-Integer Quadratic Constrained Programming MIQCP; Bilevel Programming BP); and objective (deviation; fuel, time; largest conflict-free set; minimize number of conflicts; time; minimize number of conflicts, fairness, time).

The authors classified CD&R methods according to how their mathematical formulation (i.e. equation) of the separation condition that defines a conflict. The 21 mathematical programming formulations for CD&R reviewed explored in total seven different separation equations. These can, generally be divided into one of three categories: geometrical conditions that identifies and solves conflicts using the trigonometric relationships between aircraft; analytical conditions that applies analytical calculus, and crossing point conditions that solves the CD&R problem by referring to intersection of trajectories as the origin in calculations.

Pelegrín and d'Ambrosio [2020] found that the majority of approaches rely on MIP or MINLP for solving the CD&R problem. Only one out of 21 methods reviewed approach the CD&R problem in 3 dimensions, the rest assume 2 dimensions (heading and speed). Most methods were found to assume that aircraft motion is uniform (i.e. linear with no waypoints) and that resolutions are implemented instantaneously when conflicts are detected. Moreover, the majority of methods had the objective to minimize route deviation when calculating solutions.

There are many parameters that the above taxonomies of CD&R methods do not consider. One example is the instance in time at which a solution is implemented. In general, most methods assume that implementing a solution instantaneously (i.e. when a conflict is detected) is most beneficial (e.g. it minimizes the heading deviation required while the separation distance requirement stays the same). When the solution is implemented then depends on the methods look-ahead time for conflict detection. Note that this is not necessarily how a human controller would solve the conflict. For once, it is unlikely that a human controller applies the identical strict look-ahead time in conflict detection. More importantly, the timing of an intervention can depend on other factors, such as the proximity of other aircraft that are not currently in conflict, but restricts the solution space if a resolution is to be implemented directly. By waiting a few minutes, leaving the detected conflict situation unresolved, the solution space may open up once the aircraft in proximity has moved away. The most suitable timing for resolving a conflict may also depend on factors beyond those associated with conflicts, such as the possibility of direct routings according to the aircraft's flight plan which may be restricted now but shortly available, the proximity of adverse weather which promotes waiting with solving the conflict. Another excluded parameter regards the choice of aircraft subjected to the resolution maneuver, or whether both aircraft initiate de-conflicting maneuvers such as in TCAS. Some models may choose aircraft to intervene with randomly, while others do so deterministically according to e.g., rules of the air.

Ribeiro et al [2020] evaluated the performance of four different conflict resolution algorithms on measures of safety (number of conflicts, separation losses, and conflict duration), stability (number of secondary conflicts), and efficiency (aircraft extra distance and time flown). The evaluation consisted of fast-time traffic simulations in the open-source Air Traffic Simulator BlueSky [Hoekstra2016]. The four algorithms were:

- Modified Voltage Potential. Classified as distributed, reactive, state based, nominal, Pairwise summed, dynamic obstacles, and considers resolutions in heading, speed, vertical. Applies shortest way out. Based on [Hoekstra, 2002],

- Solution space. Classified as distributed, reactive, intent based, joint solution, nominal, only considers heading and speed, dynamic obstacles, applies shortest way out. Based on [Van Dam2009],

- Explicit coordination. Classified as explicitly negotiated, distributed, state based, only heading, probabilistic, pairwise sequential, aircraft have deconflicting policies that determine the solution chosen. Based on [Yang, 2019], and

- Sequential cost. Classified as centric, heuristic, intent, probabilistic, sequential strategic, flight plan, chooses trajectory with lowest (path) cost. Based on [Hao, 2018].

To allow for comparison, the four methods were adapted so that all aircraft acted on a tactical timescale (3-5 minutes) and could perform the same maneuvers (heading and/or speed). For conflict detection, all algorithms searched for conflicts every second using a look ahead time of five minutes. Trajectory propagation was defined as linear (i.e. state-based). For manned and unmanned traffic, separation minima were defined as 5NM and 50m, respectively. In addition to the conflict resolution method, three different traffic densities were considered that reflect the gradual increase in traffic volumes from current day until 2035.

Results showed that the Modified Voltage Potential and Solution space diagram methods performed better than the Explicitly coordinated and Sequential cost methods. Performance of all methods in general worsened as an increase of traffic density. Overall, the former methods resulted in less conflicts, less secondary conflicts (i.e. when solving one conflict causes another conflict), reduced the duration aircraft were in conflict, and reduced additional flight distance and flight time attributed to the conflict resolution applied. Noteworthy is that the Modified Voltage Potential method had the lowest number of separation losses, whereas the Solution Space Diagram caused a notably higher number of separation losses. This was attributed to an inability of the Solution Space Diagram method in finding a solution in some situations, and therefore not initiating a resolution maneuver. This limitation may be related to how initial versions of the Solution Space Diagram determined the ‘no-go’ zones of conflicts without applying a time-window that limited the algorithms look-ahead time.[CW1] A reason for why the Modified Voltage Potential method performed best in terms lowest number of separation losses is that it allows for creating temporary secondary conflicts. While it is generally considered a risk to cause a secondary conflict situation, it may be the only way to solve some conflicts in high traffic density scenarios.

Taken together, the comparison made by Ribeiro et al [2020] suggests an advantage for tactical conflict resolution, distributed methods with reactive algorithms that solve conflicts locally by using the “shortest-way-out” strategy (in contrast to searching for an optimal global solution). Given an ATC environment where control is distributed, such approaches appear most suitable. In contrast, today's strategic and tactical CD&R approach in ATC can be considered centralized in that the controller exercising control is responsible for separation assurance. This, however, does not rule out the use of decentralized CD&R methods for aiding the controller in CD&R. This may, in fact, represent a suitable division of responsibility between the automation and the human, where the human is responsible for finding the global best solutions, and the machine locally best solutions.

3.2 ML CD&R approaches

Of relevance to MAHALO is how ML approaches to CD&R have been used in previous ATC related research. We therefore conducted, as part of this SOAR, a focused review into the literature specifically addressing ML applications in ATC. The purpose of the literature review was to 1) overview and determine the current state of the art in AI approaches in general, and ML approaches specifically, for solving the CD&R problem, and 2) provide a taxonomy of the AI/ML approaches used. This study extends previous work on ATC and UTM CD&R taxonomies proposed by Kuchar and Yang [2000], Jenie et al [2017], Ribeiro et al [2020], and Pelegín and d'Ambrosio [2020]. The goal is to use the framework as a guide to the MAHALO project for determining the most suitable method(s) and approach for developing a ML CD&R system.

The literature review was conducted in July-September, 2020 and comprised three steps: First, relevant articles were searched using a set of keywords. The keywords were: *conflict detection*; *conflict resolution*; *machine learning*; and *artificial intelligence*. In addition, sub-terms to ML were used: *reinforcement learning*; *supervised learning*; and *neural net/works*. The following databases were searched: SCOPUS; IEEE Explore; Web of Science; and Google Scholar. Step 2 consisted of reviewing article abstracts, and selecting relevant papers for further analysis. In step 3, papers were evenly distributed among five researchers for an in-depth review. The articles were reviewed in a structured way using excel and summarised accordingly:

- What was the paper about?
- What were the main findings?
- How is it relevant to MAHALO?

The following inclusion criteria were denied: articles should discuss solutions for conflict detection and/or resolution in context of ATC and UTM using ML or other AI approaches. The search was not limited to a time period.

In total 45 articles, written between 1998 and 2020, were selected based on the abstract review in step 2. The number of papers considered for the in-depth review was reduced to 28 after the decision was made to only focus on articles written after 2015. The main reasons for this were twofold: firstly the field of ML has seen a lot of advances and innovations that have rendered some of the previous papers and methods suggested in them outdated; Secondly the advances in the amount of computing power and the more optimized tools for using said computing power have led to certain applications and methods being viable where previously they would be considered too computationally expensive to be of any practical use. Eight articles were categorized as either literature reviews or concept papers that did not present a specific ML solution to CD&R. These were excluded from the review. Two articles were found to be close to identical and only one was retained for in-detail review [Tran et al. 2020]. The final list of reviewed articles totaled 19.

3.3 Literature review results

We classified ML CD&R methods following a nine-category framework. The objective was to create a framework that allowed for overviewing and comparing all the papers and their respective approaches. Table 3.2 details the 19 articles reviewed in relation to the AI CD&R methods framework. Table 3.3 details the abbreviations used for categories and classifiers used in the framework.

The majority of research applying ML methods for CD&R have focused on the conflict resolution problem. Almost all approaches have relied on RL. Only a few have addressed conflict detection using ML methods. None of the research reviewed considered the transparency of the systems created. In terms of conformance, three papers have attempted to create conformal resolution advisories. Van Rooijen et al. [2020] (van Rooijen, 2020 in table) and Regtuit et al. [2018] (Regtuit, 2020 in table) proposed individual sensitive systems building on SL and RL approaches, respectively, that strive to suggest personalized resolution advisories. Tran et al. [2020] (Tran, 2020 in table) had the RL system learn from ATCo's conflict resolutions during training, approaching conformance on a group level.

Table 3.2. CD&R methods reviewed. Listed by publication year, starting from oldest.

Reference	Problem	Plan	Dom	Dim	App*	MA	Feat	ReMan	ReObj
[Kim et al., 2016]	CD&R	T	ATM	3D	ML,SL,(NN, SVM)	N	Fp, RT, Rc	H,A	N/A
[Calvo-Fernández et al., 2017]	CD	S	ATM	3D	UL (kmC)	N	FP,RT	S/A	MR/M/P/Conf
[Sathyan et al., 2017]	CR	T	ATM	2D	GFS(GA, FIS)	N	Tr,Tb	H	MD
[Xu-Rui et al., 2018]	CD	T	ATM	3D	ML,SL, (SVM, EL)	N	C_x, C_v	N/A	N/A
[Regtuit et al., 2018]	CD&R	T	ATM	3D	ML,UL, (KmC,GKC), RL(QL)	N	S	H	Conf
[Pham et al., 2019a]	CR	T	ATM	2D	ML,RL (DDPG)	N	Tr,FP	H	MR,MP
[Brittain and Wei, 2019a]	CR	T	ATM, UTM	2D	ML,RL (DRL, PPO)	Y	T	H	M,MD
[Liang et al., 2019]	CR	T	ATM	2D	ML,RL (MDP,SA)	N	S, Tr, T_T,RA	S/T/T_T	M,MD
[Wang et al., 2019a]	CD	T	ATM	2D	ML,SL (MLR,SVM, FFNN,GBM, RF), UL(KNN)	N	RT.Tr, T_b	N/A	N/A
[Wang et al., 2019b]	CD&R	T	ATM	2D	ML,DRL (DQN,MDP)	N	Tr	H	MD
[Wen et al., 2019]	CR	S	ATM	2D	ML, RL (MDP,DDPG)	N	Tr	H	MD
[van Rooijen et al., 2019]	CR	T	ATM	2D	ML,SL (CNN)	N	PD,Rc	H,S	Conf
[Pham et al., 2019b]	CR	S,T	ATM	2D	ML, RL (DQl, DDPG)	N	Tt, RT	H	MR
[Brittain and Wei, 2019b]	CD&R	T	ATM, UTM	2D	ML,RL (DRL,PPO, LTSM, D2MAV)	Y	Tr	H	M,MD
[Ghosh et al., 2020]	CR	S	ATM	2D	ML,RL (MDP,KBSF, EL,PPO, QL)	Y	FP	S	MC,MD
[Wang et al., 2020]	CD	T	ATM	3D	ML,SL (FFNN, GBM,RF), UL (KNN)	N	Tr, T_b	N/A	N/A

Continued on next page

Table 3.2 – *Continued from previous page*

Reference	Problem	Plan	Dom	Dim	App*	MA	Feat	ReMan	ReObj
[Mollinga and Hoof, 2020]	CD&R	T	ATM	3D	ML, DRL (MDP,GCN, GAT)	N	Tr,RT	H,S,A	MD
[Tran et al., 2020]	CD&R	T	ATM	2D	ML,RL (DDPG)	Y	S,C_S,C_v	H	Conf
[Brittain et al., 2020]	CD&R	T	ATM, UTM	2D	ML,RL (DRL,PPO, LTSM, D2MAV)	Y	Tr	H	MD

Table 3.3. Categories and classifiers of the CD&R methods taxonomy.

Category	Abbreviation	Classifiers	Acronym
Problem addressed	Probl	Conflict detection Conflict resolution Both	CD CR CD&R
Avoidance planning	Plan	Strategic (20 min or more) Tactical (3-20 min) Collision avoidance (escape)	S T CA
Domain	Dom	ATM UTM	ATM UTM
Dimension	Dim	Horizontal Vertical Horizontal and vertical (3D)	2D V 3D
Main AI approach	App	Machine learning Supervised learning Reinforcement learning Unsupervised learning Genetic Fuzzy Systems	ML SL RL UL GFS
Multi-agent	MA	Yes No	Y N

Feature engineering	Feat	Trajectory Time (binary) Pixel data Flight plan Secondary radar traces Conflict states (angle, CPA, time to CPA, exit point) Resolution command Coordinates space Coordinates velocity	Tr T_b PD FP RT S Rc C_s C_v
Resolution maneuver	ReMan	Speed Heading Altitude Time Turning time on sequence leg Runway allocation	S H A T T_T RA
Resolution objective	ReObj	Minimize route deviation Minimize congestion Minimize delay Maneuverability penalty Consensus Merging Conformal	MR MC MD MP C M Conf
Conformance	Conf	Individual (human) Group (human) None	I G N
Transparency	Trans	Yes No	Y N

Three papers explored ML methods for only CD. Two of these were from the same group of researchers [Wang et al. 2019b; Wang et al 2020]. They compared in total six different ML approaches for CD. Five methods are considered SL approaches: Feed-Forward Neural Networks (FFNN), Multiple Linear Regression (MLR), Support Vector Machine (SVM), Gradient Boosting Machines (GBM), and random forests (RF). The K-Nearest Neighbour (KNN) method is a UL approach. In the first study where all six methods were compared, the MLR and SVM methods were found to perform the worst [Wang et al. 2019b]. These were excluded in the second study. The methods performing the best were the GBM and FFNN.

The majority of previous approaches have restricted CD&R to a 2-dimensional representation of the environment (aircraft fixed to one altitude) and limited resolutions maneuvers to heading changes. Only five articles considered CD&R in relation to a 3-dimensional environment, representative to the real world. Only one paper considered all three resolution types (heading, speed, altitude) for solving conflicts [Mollinga et al. 2020]. In the study by Liang et al. [2019] where CR in the TMA was explored, additional high-level resolution maneuvers were considered as part of a merging task. In addition to speed and time changes (i.e. arrange an expected time overhead a fix), two more resolution maneuvers could be used merging decision of aircraft along the same route for approach: the turning time on a sequence leg, and runway allocation.

Several papers were written by the same or similar group of authors, indicating an overlap between these. Four research clusters were found to attribute to a majority (11) of the 19 papers reviewed in detail. Two papers from Wang et al. [2019b; 2020] explore ML methods exclusively for CD. Researchers at the Technical University of Delft (TUD) have in two studies explored conformal ML approaches for CD&R. Brittain and Wei form one research cluster originating in the USA, with three papers published [Brittain et al. 2019, 2020a, 2020b]. Their ML method is among the more advanced, relying on deep RL (DRL), Proximal Policy Optimization (PPO), and a Deep Distributed Multi-Agent Variable framework with attention networks. However, approach is limiting as it only envisions a 2D environment where the only resolution option is heading changes. The fourth research cluster originates from Nanyang Technological University in Singapore and the Air Traffic Management Research Institute (ATMRI). This group of researchers has explored an advanced RL approach making use of a Deep Deterministic Policy Gradient (DDPG) algorithm.

Following initial review, the review team discussed the overall quality and relevance of the work presented in each article. It was found that most papers reviewed had not modelled the domain environment realistically and in line with MAHALO assumptions, due to such factors as: the usage of a simplified state space (2D environment, for example), the limitation to only use a subset of all the possible conflict resolution methods, the fact that the focus of the article was not aligned with that of MAHALO (in one case the article focused on deconfliction on a purely strategic level), among others.

This discussion led to some conclusions being drawn and to a rating system being made in order to sort the papers by relevance and applicability to MAHALO. A two-step rating system was used. The first step consisted of culling non-relevant literature. The criteria for passing the first step was that a paper should present an ML approach for conflict detection, conflict resolution, or both. Following this step,20 papers were retained (also in Table 3.2).

3.4 Implications for MAHALO

Whereas CD is relatively binary (it is a conflict or not)[3], CR has many more possible actions and an almost an infinite number of possible outcomes (especially on larger times scales). This is where humans and computers will differ most: humans tend to adopt a more limited look-ahead time (e.g., 10 minutes) and therefore may have a limited view on the many possible outcomes of actions. Computers could potentially evaluate almost an infinite set of actions and outcomes, similar to how the chess-playing computer Deep Blue could evaluate 200 million positions per second [Campbell, 1999].

Differently from chess, operational conditions in ATC are often less predictable. Anticipating multiple steps ahead into the future requires the adoption of longer look-ahead times, featuring increased uncertainty regarding aircraft (navigational) states and environmental disturbances (e.g., changing wind conditions).

Some of the conclusions impacting MAHALO from the literature review were:

- The need for a conflict **detection** and a conflict **resolution** algorithm.
- The fact that using the conflict **detection** and **resolution** algorithms in conjunction with each other appears to be the more efficient technique.
- The most desirable ML method for the conflict **detection** is likely to be a **Supervised Learning** algorithm.
- The most desirable ML method for the conflict **resolution** is likely to be a **Supervised Learning** algorithm in conjunction with a **Reinforcement Learning** one so as to be able to learn from expert knowledge but also be able to find more optimal solutions based on conflicting objectives (conformance to operator, additional miles flown from original flight plan, etc).

One of the main concerns with using ML agents to help ATCos is the likelihood that the ATCo will accept the solution given by the ML agent. In this context it is important to consider those algorithms that attempt to be conformal to the specific ATCo they are assisting as well as those algorithms that seek to become more transparent and thus provide an explanation for the suggested solution.

It is also important to mention that ATM, in particular, and aviation, in general, are so called "safety critical domains". this means that safety comes first and foremost before any other goal the system might have such as reducing fuel consumption or travel time. All this means that having a system that is both reliable and safe is a primary concern when it comes to considering what methods and approaches can be used within the MAHALO context. This disqualifies several systems and

[3] Loss-of-separation has a legal / binary definition (e.g. 5nm and 1000 feet vertically), but conflict threshold, and conflict detection methods, can vary.

methods that do not have a good enough performance. For some applications, for example a voice recognition software on a smart television, a 75% success rate might be acceptable but for ATM this is absolutely not the case.

Overall, literature review supported our initial hope that ML methods can provide attractive solutions to solve both conflict detection and conflict resolution problems, as displayed by the articles described. It is, thus, accurate to say that ML is one possible promising approach to be used in future ATC systems.

4. Machine Learning

This chapter will briefly introduce the domain of **Machine Learning (ML) and discuss its applications both in general, and in relation to ATM**. For MAHALO, ML model explainability and transparency is an essential part of the research.

Section 4.1 sketches a short history of ML developments.

In section 4.2, a classification of current ML approaches is provided.

Sections 4.3 and 4.5 describe some relevant applications of ML, both in general and in the field of ATM.

In section 4.4 the challenges of ML are discussed.

Section 4.5 contains an analysis of the transparency challenges and opportunities associated with ML.

This chapter will end with a summary of implications (section 4.7) for the MAHALO project.

4.1 History

Machine learning (ML) is a fairly recent outgrowth and subset of the Artificial Intelligence (AI) field. Unlike traditional AI, ML does not rely on pre-programmed 'if-then" approaches. Rather, ML generally involves relating input to output via a self-organising process. Much of the early history of ML comes from the field of statistics, for example Bayes' Theorem that allows inferences based on prior knowledge (data).

A key property in ML is that the models can automatically improve themselves based on the availability of new data. This is analogous to human learning, where connections in the brain are continuously being adapted to create memories and to make decisions based on inputs from the environment.

In the 1940's McCulloch and Pitts [McCulloch, 1943] used this analogy of the human brain to describe the workings of neurons using logical calculus, thereby creating a model of the biological neuron, which later developed into artificial neural networks (ANNs). Although early ANNs were mainly used to study the behaviour of biological networks, they became more powerful as generic function approximators in the 1980s [Rummelhart, 1986] with the development of backpropagation methods, which allow for automatic updating of the neural network weights such that a model can be trained to approximate a given input-output relation in a dataset. Since then, ANNs have been the main type of models used in ML, although many other model structures (e.g. decision trees [Kotsiantis, 2018], and genetic algorithms [Lee, 2013]).

4.2 ML Methods and Approaches

ML generally distinguishes three high level approaches: Supervised Learning (SL), Unsupervised Learning (USL) and Reinforcement Learning (RL). Figure 4.1 presents a high-level overview of the most currently popular ML methods, grouped under each of these three classes.

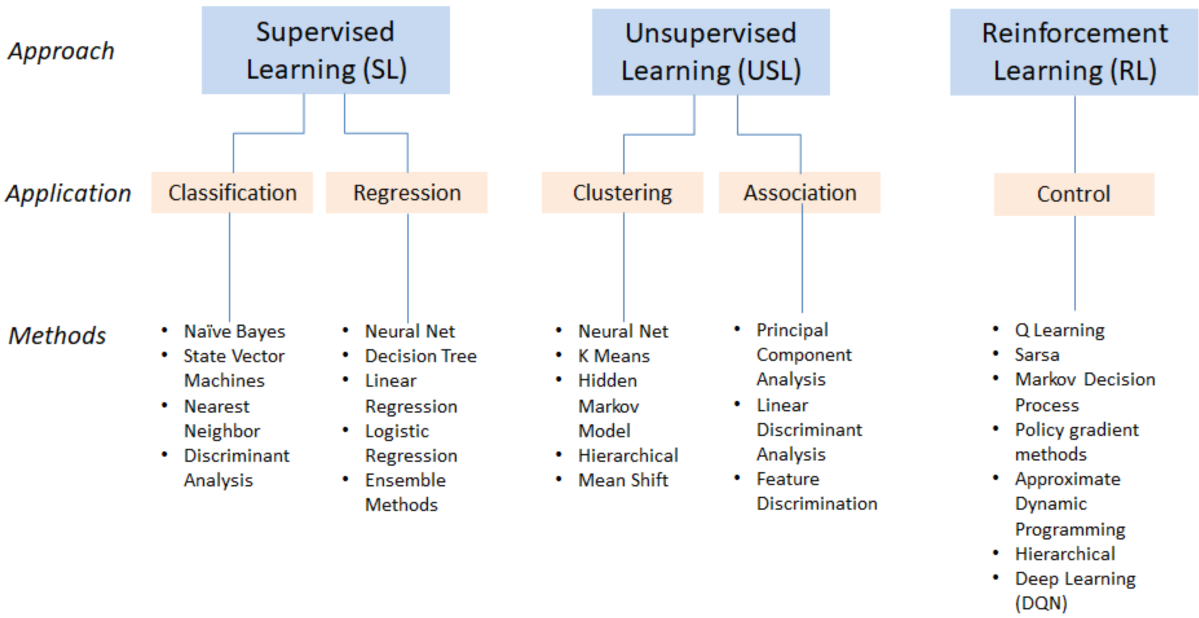


Fig 4.1 Some popular ML methods, grouped by general approach and application.

In SL a model is trained on labeled data, meaning that a set of input and corresponding desired output is presented to the model. The model then has to adjust its parameters such that a good fit is made between the model output and the desired output. SL is often used for classification tasks, for example to interpret handwriting or to identify in a camera picture which blob of pixels represents a bicycle or a pedestrian. Typically, there needs to be a large set of pictures in different environmental settings, light and weather conditions etc, to be able to train such a model with a high level of accuracy for the intended use case. [Asadi 2009] presents an example of the current approaches used for SL. SL training is based on a cycle of forward propagation of the input followed by a backpropagation of the error (desired vs actual output of the network) where the weights of each neuron are adjusted.

USL refers to machine learning approaches for investigating data sets to find patterns and structure such as classes and relationships. In USL there is no set of labeled or desired data present in the training stage. Instead, the model has to adjust its parameters in such a way that the model outputs satisfy some specific properties, for example that the output is clustered into a possibly predetermined number of groups. USL is often used to discover patterns in data, since it does not require the data to be labelled *a priori*. [Sathya 2013] presents a comparison between the approach taken by USL and SL.

RL, unlike SL and USL, is mainly used for agent based tasks, where the agent is interacting with the environment and thereby changing the state of the environment that the agent can observe. A RL model can be described as a trial and error process that comprises the following fundamental parameters: an environment that is assessed at a certain point in time. The RL model interprets and describes the environment in terms of states [Nguyen 2019]. The RL agent can act on these states by performing actions. The action to be performed (i.e. the decision made) depends on the inferred state and is governed by the RL agents policies. The possible actions that can be taken can be described in terms of an action space, which often is defined as discrete. A policy is considered deterministic if the behavior is predictable. If the behavior is unpredictable, the policy is stochastic. An 'optimal' deterministic policy can be developed from a stochastic policy by means of policy improvement, where the agent is allowed to update its policy until a better policy cannot be found. When an action has been performed, the agent evaluates the outcome based on a reward function: the behavior is reinforced if the outcome is desirable according to this function, and changed if the outcome is undesirable.

In RL there is also no set of desired outputs that is given to the model. Instead a reward is provided to the agent based on the current state that the agent can observe. The model then has to modify its parameters such that the model outputs change the state of the system in a way that maximizes the expected sum of future rewards. An RL agent's decision process and policy is typically modeled using a Markov Decision Process (MDP), where the goal is to find an optimal policy that determines the actions taken for different states. MDPs are forming the theoretical foundation for many RL algorithms. An MDP model of a decision process is defined in terms of a list of elements to be considered (i.e. a *tuple*): typically a 4-tuple consisting of element S for describing the state (I.e. state space), element A for the available actions (I.e. action space), element P for the probability that an action will lead to a certain state, and element R for the reward function. In an MDP the future state is only a function of the current state and current action, meaning that previous states do not directly affect the future state.

If a system is not Markov, i.e. when the future state depends on other things than the current state and action, the RL agent is not guaranteed to converge to the optimal policy. A comprehensive introduction to RL and some of its principal methods can be found in the book by Sutton and Barto [Sutton, 2017]. Additionally, there are several extensions to the traditional RL approach that are capable of handling processes that do not meet the conditions of an MDP. As an example, in [Liang et al, 2019] a Semi-MDP is used. These extensions are particularly useful when considering Multi-Agent scenarios where the MDP conditions are no longer met, for example, in [Chu et al, 2020]. Recent developments in RL include the usage of Deep Neural Networks, leading to the creation of the subset known as Deep RL. A comprehensive survey on the state of Deep RL can be found in [Arulkumaran, 2017]. [Hongmin et al, 2020] gives a more recent overview of current RL methods, see figure 3.2.

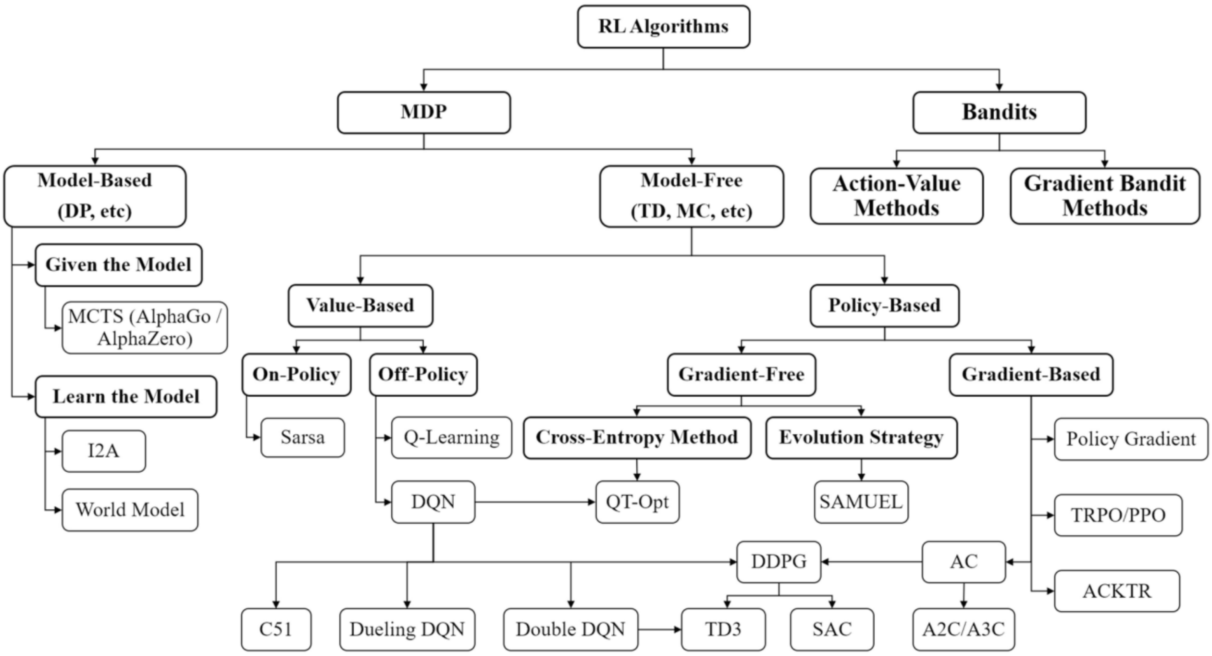


Fig 4.2 Overview of RL methods from [Hongmin et al, 2020]

4.3 Domain applications of ML

This section will list some of the major applications of ML. Applications in the field of ATC/ATM will be treated separately in section 4.6. It is out of the scope of this chapter to give a complete review of all ML applications, but the main fields will be treated and a few specific examples of applications will be shown.

Machine learning is being applied in almost every field of science. A few examples of ML applications are:

- Natural language processing
- Computer vision

- Financial market analysis
- Handwriting recognition
- Adaptive control systems (e.g. robotics, automotive, process industry)
- Optimization
- Health monitoring and maintenance predictions
- Board games (GO [SIlver et al, 2016], Atari [Mnih et al, 2013])

Figure 4.2 shows an example of how ML is used to classify objects in the camera view of a car. The model is capable of distinguishing cars, other road users, road signs, traffic lights, road markings. Based on this information, an automatic controller can steer the car and avoid obstacles.

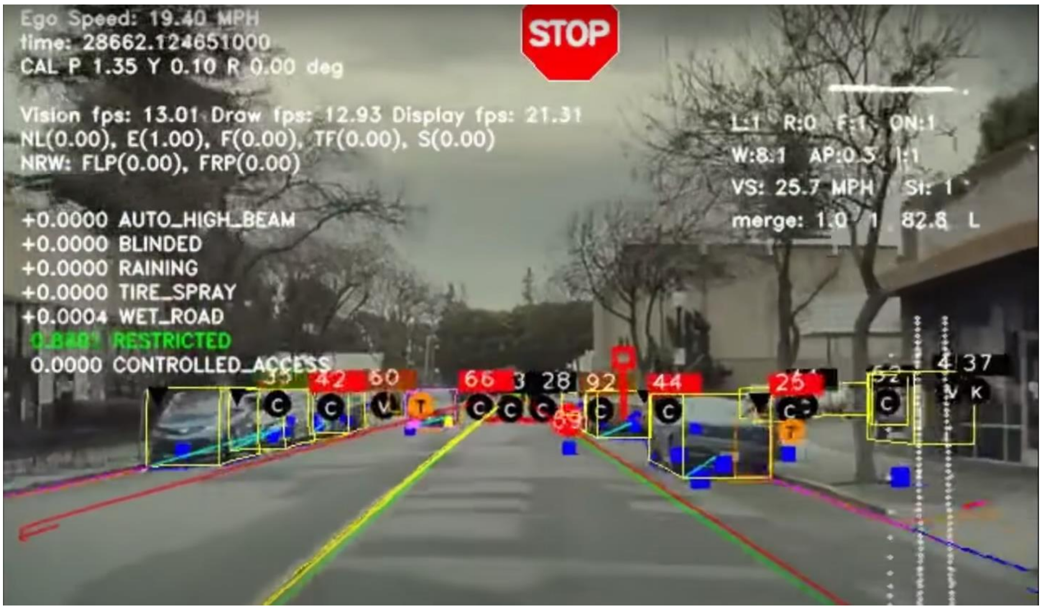


Fig 4.2. ML for automotive computer vision (https://www.tesla.com/autopilotAI (Sep, 2020))

In 2019, Springer published a book on Lithium-Ion Batteries that was written by a machine [Author, 2019]. It contains a summary of the recent literature of that field and has auto-generated summaries and links to other articles. Perhaps the most advanced system for text generation is OpenAI's current GPT-3 model, which produces humanlike prose so convincing that it has been called 'too dangerous to release."

ML is becoming ubiquitous in day-to-day life: personal assistants on virtually every smartphone (Amazon's Alexa and Apple's Siri, among others), ML algorithms that calculate how much we will pay for a cab ride given the supply and demand of the current area (used by Uber, for example), email filters that “learn” how to better filter malware and spam and recommendations on products based on previously bought or searched for goods on most online retailers. Photo editing software uses ML to offer features such as content-aware fill, where parts of an image can be filled based on previously learned patterns or classifications.

It can be concluded that AI and ML in particular is becoming normal and accepted in everyday life, even if it is not always understood. In Section 3.6, a detailed literature study on application of ML in ATM/ATC will be provided.

4.4 Challenges of ML

Although ML has seen many successful applications, there are still many challenges in ML that hold back its widespread application. This section will discuss some of these challenges and how they are going to be addressed in the MAHALO project.

The challenges will be split into three main categories:

1. Data quantity and quality
2. Hyperparameter tuning
3. Explainability

4.4.1 Data quantity and quality

ML methods are known for their large data requirements. The efficiency of identifying patterns in data is low compared to what humans can do. However, because ML can operate fast, there are applications where ML outperforms humans (e.g. Alpha Go).

The quality of the data is important, since ML models are in general poor in extrapolating to unseen data. This means that the model that is trained can be a good fit for the data it was trained on, while having bad performance for a different realisation of the input set from the same generating process. This is also linked to the process of overfitting, where so many parameters are used in the ML model, that the model learns to fit to such a level of precision that noise on the input data is also modeled. This leads to poor generalisation, since a new batch of input data with different noise, will give incorrect outputs. Humans have great skill in generalising inputs and behaviour, since we can often make good decisions in situations that we have never seen before, but which are similar in some sense to what we have experienced before. This skill is what is lacking in ML algorithms, where similarity is attempted to be captured by creating features in the input space. Even when handcrafted features are presented to the ML model, the performance is not close to what humans can do. The ultimate goal is to have the ML model learn by itself what relevant features are and what elements of the input are important when it comes to determining similarity to previous experiences. Until these skills are getting close to the human level, ML models will need much more data to converge to a good output.

Training data should be rich enough, and a reasonably approximate (unbiased) representation of the data for which the ML model will be used in operation. For MAHALO this means that enough data for training the ML models is required. This will likely be a mix of human generated data, from expert air traffic controllers, and computer-generated data, for example from existing expert systems.

Related to data quality and quantity is the way how data is presented to the ML algorithm. Information from a data source can be represented in different ways before it is fed to the ML algorithm. It can even be pre-processed in order to extract the relevant information, or to discard information that is deemed irrelevant. This step of processing the information from the data source before it is fed to the algorithm is known as Feature Engineering.

4.4.2 Hyperparameter tuning

A major challenge of ML is the dependency of the outcome of the learning process on the settings of the algorithm that is being used, something that is nicely caricatured by the cartoon of figure 4.3.

Fig. 4.3. Machine Learning tuning https://xkcd.com/1838/

ML models often consist of many parameters or coefficients, such as the weights in a neural network, but besides this there are settings that determine how the training is performed. These settings are called *hyperparameters*, which distinguish them from the parameters of the model itself. For example, when looking at Reinforcement Learning these are some of the hyperparameter s[Sutton 2017]:

- **Learning rate**: determines the step size in the learning process
- **Discount factor**: determines relative weight of immediate rewards versus long term rewards
- **Exploration rate**: trades of exploration and exploitation of knowledge
- **Reward function**: tries to capture the goal for the agent
- **Model topology**, e.g. how many layers to use in a neural network
- Parameter initialisation

The high dimensionality in hyperparameters that needs to be tuned, together with the lack of transparency of ML models makes designing and tuning ML models challenging. Tuning of hyperparameters in ML often requires expert knowledge or brute force methods, although some generic approaches have been developed that claim they can optimize hyperparameters for any ML model [Wu et al 2019]. An additional difficulty is that hyperparameters do not generalise well to different applications for the same algorithm, meaning that an ML algorithm that is designed to perform an ATC task can have very different optimal hyperparameters then the same algorithm applied to control the motion of a UAV.

There is expert knowledge available within the consortium on hyperparameter tuning of RL and SL algorithms, but also automated optimisation tools can be employed.

4.4.3 Explainability

This section provides a computer science background on transparency, especially from a ML perspective. For a Human Factors (HF) perspective on transparency, see section 5.3.

As previously discussed, some ML models lack transparency, which prevents the growth of shared awareness and shared intent between machines and humans [Bhaskara et al 2020]. A deep neural network with multiple hidden layers and possible thousands of connections is a good example. This model can represent the policy of a ML agent that, presented with a certain input, will generate a suggested action to take. However, even if the output of the action seems right to the human observer, the human cannot understand the process of how this output is determined by just looking at the neural network weights. There is no clear causality between a change in inputs and a change in output of the model, because the relationship depends on thousands of parameters. It is also generally not clear how the output of the model will change, with constant input, when one of the model parameters is modified.

There are many approaches within the AI community that strive to explain aspects of how the system works or how a particular output was derived. In the AI community, the term transparency is used in context of recommender systems research [Ricci, 2015], while other sub-communities use similar terms. The most notable are ML interpretability [Murdoch, 2019], Explainable AI (XAI) [Gunning, 2017], and intelligibility of context-aware systems [Bellotti, 2001].

Explainability of ML is part of a broader concept of explainability of Artificial Intelligence, XAI. Within ML some effort has been made on making the models more explainable. For example, in RL, there are ways to decompose rewards that the agent receives, such that after training one can get an idea of which parts of the reward function were most used to converge to the final policy [Juozapaitis et al 2019]. For deep neural networks, which can be used in either (un)supervised learning or reinforcement learning, there are possibilities to visualize which parts of the input (features) are most important in generating an output [Seifert et al., 2017].

In RL, the output of the ML model is an action that the agent takes. Therefore, the mapping from input to output can be seen as a control policy or control law. This policy is strongly dependent on the reward that is assigned to states and actions. The lack of transparency in the ML model can lead to surprising and unexpected behaviour of the agent. For example, when a robot is taught to find the shortest path to a goal state, then a common way to give rewards is to penalise every step taken, in order to minimise travel time, to give a reward for reaching the goal, and to give additional penalties for hitting any obstacles on the way. If the reward for reaching the goal is not set high enough, then the agent will learn to hit the closest obstacle it can find, thereby ending the episode so it does not have to collect penalties for every step it would take to reach the goal. This is completely opposite to the behaviour that we want to see, but the agent is not malfunctioning because it is still maximising its expected sum of future rewards. There are many examples of unexpected behaviour like this in RL research. Therefore, careful selection of the reward function is an important part of the ML algorithm design.

In the context of AI and ML systems, transparency becomes intrinsically difficult to achieve due to the amount of data processed and complexity of the systems (e.g., multiple deep layers, number of rules) that greatly exceed human abilities to timely make sense of the data. Echoing transparency, ML interpretability has been defined as the “ability to explain or to present in understandable terms to a human” [Guidotti, 2018]. In the field of interpretable ML, Guidotti et al. [2018] found that graphical decision tree representations and textual decision rules (i.e., ‘if-then’ algorithms) are among the most commonly used methods for explaining both the ML model (i.e., global explanation) and its specific output (i.e., local explanation). The output of linear models is often explained by highlighting key input parameters and their relative importance. Similarly, explanations of deep neural networks (DNNs) used for image recognition often make use of either saliency mask, which visualizes key areas/features in the input image, or activation maximization, which determines key neurons in certain layers activated by the input image. Example methods in image classification using convolutional neural networks (CNNs) are the Pixel-Wise Decomposition (PWD), which uses heatmaps to visualize individual pixels of the input image that determine the output [Bach, 2015], and the Visual Back Prop (VBP) method, which uses masks to visualize the set of pixels in the input image that determine the output [Bojarski, 2016].

Other recent developments include, for example, Local Interpretable Model-Agnostic Explanations (LIME) [Ribeiro, 2016], Contextual Explanation Networks [AlShedivat, 2017], and Contextual Decomposition [Murdoch, 2018]. However, some have been shown to be unstable (i.e., LIME) by providing different explanations for similar inputs that prevent their use in high stake domains [Alvarez-Melis, 2018]. Contextual Explanation Networks is an interesting approach to XAI that combines ML methods and probabilistic models [Al-Shedivat, 2017]. Contextual Explanation Networks processes a subset of input features and generates parameters for a sparse linear model which can be assessed by domain experts. Subsequently, the generated model is applied to another subset of inputs and produces a prediction [Al-Shedivat, 2017]. According to the developers, this approach is robust and a candidate for high stake domains. Contextual Decomposition [Murdoch, 2018] provides explanations by decomposing the output. Although primarily used for natural language processing, this approach should be able to provide importance scores for individual features and feature interactions also for other LSTM-based models and domains such as aviation. Moreover, reward decomposition approaches have been used to explain decisions, in particular action selection, of RL agents [Juozapatis, 2019].

Murdoch et al (2018) proposed a framework for determining which interpretable ML method to choose when trying to understand how a ML model has addressed a particular problem. The PDR framework comprises three criteria to consider: the ML model's Predictive accuracy, Descriptive accuracy, and Relevance. The two accuracies relate to the different errors that can occur in a ML model: either when interpretable ML methods are applied to understand aspects of the ML model in the model stage when data is processed and relationships learned (Predictive accuracy), or when interpretable methods are applied to analyse and understand the output of the ML model (descriptive accuracy). Relevance determines how well the interpretable method affords its intended audience (e.g., users) insight to the problem the ML models is attempting to solve.

Murdoch et al also reviewed a large number of interpretable ML methods and categorised them as being either model-based or post-hoc based. Both strive to increase the descriptive accuracy.

Model-based interpretability methods are further divided into five categories:

- Reduced sparsity approaches: the assumption is that it will be easier to understand the ML model if the number of parameters, or features, considered by the interpretable ML method are reduced. The objective is often to determine key features driving the prediction/outcome. An example method is sparse coding.
- Simulatability approaches: the assumption is that understanding of the ML model increases if the user can 'simulate' (i.e. reason) the ML model's decision-making process. Example methods are decision trees and rule lists.

- Modular approaches: the assumption is that understanding can benefit from considering parts of the ML model independently. Example methods include generalised additive models, attention, and modular network architectures.

- Domain-based feature engineering approaches: the assumption is that understanding increases if the features used by the ML model are created (or selected) using domain expertise.

- Model-based feature engineering: the assumption is that understanding can benefit from having a ML method automatically determining the features from the data and provide descriptions of their structure. Several unsupervised learning methods are given as examples, such as clustering, matrix factorisation, and dictionary learning. Other methods such as principal component analysis, independent components analysis, and canonical correlation analysis are exempted, which reduce the number of dimensions of the original data.

Post-hoc interpretable methods are further divided into two sub-approaches with six categories:

- Dataset level approaches consisting of:

    - Interaction and feature importances: the assumption is that understanding can be increased by showing the importance of individual features, or the importance of specific interactions between features, for a specific output. Examples of feature scoring methods have been used with neural networks, random forests, and generic classifiers.

    - Statistical processing of features: the assumption is that understanding can benefit by converting raw values of features to statistical confidence measures and determine if a feature is statistically significant in relation to the prediction/output.

    - Visualisations: The assumption is that understanding benefits from visualising what a ML model has learned. Example methods include using regression plots for linear models. For image data processed by s, example methods include applying visualising filters, revealing responses of individual neurons, and grouping different neurons. For LSTMs, tools are proposed for tracking the ML models decision process.

    - Analysing trends and outliers in predictions: the assumption is that a ML model can be better understood by revealing trends and outliers in the data.

- Prediction level approaches comprise methods used to increase the understanding of how a ML model makes specific predictions. Approaches consists of:

    - Feature importance scores: the assumption is that by scoring, or otherwise indicate, how important different features are for a specific prediction/output. Examples include tabular visualisation of how different features are weighted for a particular output, or using heat maps to highlight important aspects in an image.

- Other than the above, such as methods for revealing the importance of interactions between features for a prediction/output.

Despite the amount of research on interpretability, there is a shortage of empirical research exploring the effects of transparency on acceptance and trust [Wang 2016]. Research on transparency in the AI domains has been criticized for focusing primarily on how to build explanations while neglecting the underlying psychology and human interpretation of them [Abdul et al. 2018; Murdoch et al. 2019].

4.5 ML applications in ATC

Recent years have seen an increased interest in AI solutions for the aviation and ATC community. For example, ML and big data techniques have been explored to improve the accuracy in trajectory prediction [Koyuncu, 2017; Vouros, 2018], to identify novel route patterns and predict airlines' route selection [Marcos, 2017], and for speech recognition of ATCo-pilot communication [Helmke, 2018]. In the FLY AI Report, the European Aviation/ATM AI High Level Group (EAAI HLG) recently concluded that AI solutions for realising future ATM operations (i.e. the digital European sky vision) must be trustworthy and human centric [EAAIHLG, 2020]. Among key areas requiring more research on AI solutions for human-machine collaboration and in safety-critical operations. The report also provides an overview of completed and ongoing ATM AI/ML research.

In current operations, controllers work with advanced automation support tools and safety nets, such as While the FLY AI report identifies AI solutions as reasonable for further developing such systems, there are significant challenges that require further research [EAAIHLG, 2020]. Notable among these are issues related to the:

- Controller's understanding of these systems and their reasoning. Further research is required on how to make such systems transparent and their behaviour explainable.
- Methods for achieving a controller-automation partnership.

A requirement is to involve operational staff early on in the design process of such systems.

To address capacity bottlenecks during en-route operations, European ATM is targeting an operational environment with the core functions of trajectory-based operations, flight-centric services, dynamic airspace usage, free-route airspace. ATC automation support system incorporating AI, and in particular ML, is considered a key enabler for future advanced separation management systems where controllers and automation collaborate to detect and solve conflicts on strategic and tactical time windows [EAAIHLG, 2020; SESAR, 2019]. There has been much research into the possibility of using several different ML approaches to assist the controller in CD&R, through conflict resolution suggestions, and into the use of ML as the basis for fully automated ATC. For the purposes of the MAHALO project we are mainly concerned with the applications that relate to having ML as an agent collaborating with the executive controller in tactical CD&R. Tactical CD&R here refers to the time window between collision avoidance measures (i.e. the time between loss-of-separation and CPA) and strategic CD&R (>20 minutes before CPA).

4.6 Implications for MAHALO

R & D into ML is moving incredibly quickly these days. We were therefore pleased to confirm, through our review, that **our initial aim of applying ML to separate CD and CR functions seems workable.** Both of these lend themselves to automation, and we have a short list of candidates to evaluate further. Development of workable models will certainly require testing and tweaking. However, we came away from the review confident that this is achievable.

It is also clear that **challenges remain in using ML.** These have primarily to do with data requirements, and in procedures for parameter tuning. These issues seem manageable.

Finally, explainability (transparency) is an ongoing challenge with ML models in general, and with some specific approaches in particular. **This issue of explainability was a known ML issue, and lies at the heart of the MAHALO project.**

5. Human Performance in ATC

Chapter 5 summarises our review into human performance aspects of ATC, that are relevant to MAHALO. This chapter focuses primarily on automation conformance and transparency, two constructs that are key to MAHALO.

This chapter will also briefly cover related concepts of trust, acceptance, and user reliance on automation.

Finally, it will identify implications for the MAHALO project.

5.1 An information processing model of ATC

The ATM Master Plan shown in figure 1.4 (SESAR, 2019) uses the Levels of Automation (LOA) framework of Parasuraman, Sheridan & Wickens [2000], which distinguished LOAs (low- to full automation) from broad task classes (acquisition, analysis, decision, and action). This perspective fits well with our evolving view of the ATC *information processing cycle*, in which controllers detect problems, formulate plans, implement those plans, and evaluate outcomes, in an ongoing cycle. As shown in figure 5.1, this cycle has two important implications for MAHALO.

First, notice that the initial acquisition-analysis phase corresponds to the conflict detection function, and the subsequent decide-act phase corresponds to the conflict resolution cycle. Second, this perspective allows us to identify the four subfunctions of CD and CR. Together, this can help us to define compatible human- and machine subfunctions.

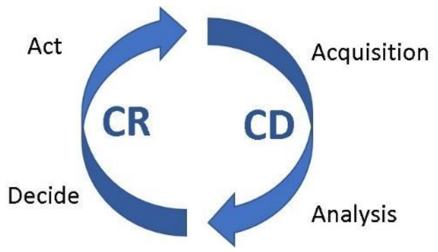


Fig 5.1 The information processing cycle of ATC.

The following sections 5.2 and 5.3 will now review human performance issues in ATC, through the lens of two constructs that are key to MAHALO: automation conformance, and automation transparency.

5.2 Automation conformance

5.2.1 Theoretical foundations

Automation is typically developed to optimize the performance of a task or solution to a problem, beyond that of human capabilities and preferences. In contrast, humans tend to apply heuristics that satisfy rather than optimize performance [Simon, 1956; Gigerenzer, 2011]. Westin et al. [2016] argued that in contexts where humans and automation are expected to work together, the divergence in decision-making processes can negatively affect human acceptance and trust of the automation. Some researchers have therefore proposed automation that can adapt to an individual's needs and preferences [Westin, 2016; Liu, 2011; Szalma, 2009; Parasuraman, 2012].

The term *strategic conformance* was introduced to describe the *apparent* match between human and automation solutions [Westin, 2016]. This similarity is external, overt, and observable, and is the extent to which cause and effect can be observed. Conformance does not provide an explicit explanation for the output. Instead, conformal automation supports understanding by providing a solution that, in appearance, matches the strategy or solution preferred by the individual. To the individual, the solution makes sense and therefore no further explanation is required. Since conformance is an attribute of the automation, it requires that the automation know something about the individual's preferred strategy or solution is. Only with this knowledge is the automation able to provide an output that can be considered conformal. The concept of strategic conformance took inspiration from the concept of cognitive tools proposed in the European Commission sponsored *Role of the Human in the Evolution of ATM* (RHEA) project (RHEA, 1998) and later explored in the Conflict Resolution Assistant (CORA) project [Kirwan, 2002].

MUFASA proposed an information processing model of automation reliance, which placed strategic conformance within the broader context of how/whether an operator comes to trust, accept, and rely upon an automated decision aid (figure 5.2).

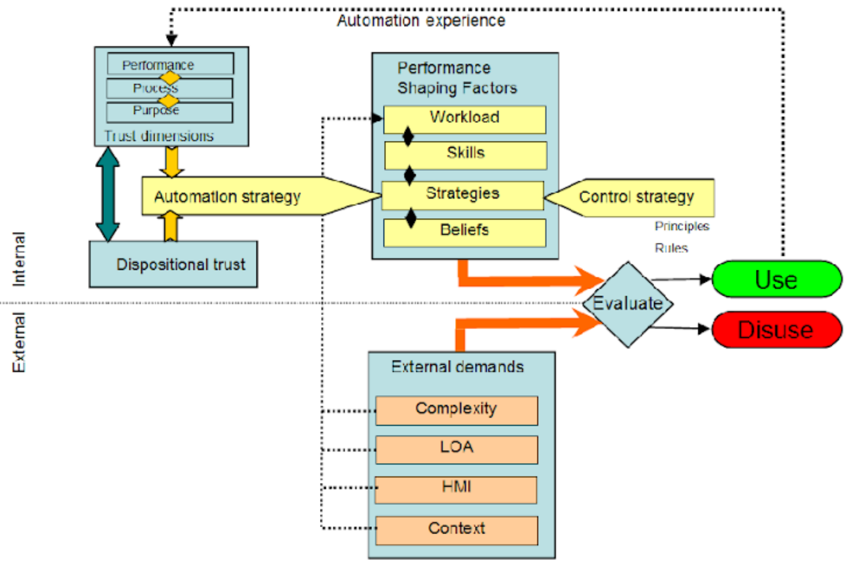


Figure 5.2 Controller reliance on automation (after MUFASA, 2013).

In this model, an operator decides whether to accept automation's advice ('should I use the system's advice?') based on a combination of external demands (e.g. time pressure) and internal demands. A large part of these internal demands is made up of trust in the system, based on both longer-term propensities to rely on automation, but also on an updated assessment of the observable and inferred functioning of the system. Essentially, trust and subsequent reliance increase as experience demonstrates that automation 'does the right thing' (or, rather, does it the way I would do it). Notice one unfortunate implication of this model, for reasons of designing future advisory automation: if the system is NOT used, there is no experience on which to build trust. One paradox of advanced automation could be: which comes first, trust or reliance? The MUFASA project proposed that the notion of strategic conformance could break this impasse, at least for novices. If automation mirrors what I would do, I will come to trust it.

Conformance can be considered on a scale ranging from fully conformal at one end, where the system adapts fully to the individual's strategies and preferences. At the other end, the automation neglects any adaptation to the human. In between these extremes, tradeoffs can be found. Before considering the individual human, the system could consider humans as a general group. Another term for such approaches to automation design is human-centered design.

5.1.2 Research on automation conformance

Personalized automation has been widely researched in the context of self-driving cars. In the automobile domain, individual driving styles have been explored to increase the human acceptance and comfort of self-driving cars [Kuderer, 2015]. This research presumes that human drivers are more likely to trust and accept an autonomous car if the car's driving style is similar to that of the driver. An AI domain that has long explored personalized automation is *recommender systems* [Ricci, 2015]. Personalized algorithms are commonly applied by actors active in electronic commerce (e-commerce), such as digital music, books, and streaming entertainment services. By collecting data about the individual's habits in using a particular service, a unique "user profile" can be created using personalization algorithms. For digital music services, algorithms can track listener activities including the songs and playlists listened to, liked, shared, skipped, and added to a library.

There has not been much research conducted on conformal automation in the context of ATC. The notion of personalized, conformal automation was first raised by the MUFASA project [2013], which simulated conformal automation via unrecognisable replays of controllers' previous performance. Results showed that when controllers were given conformal advisories (i.e., solutions matching their own), they accepted more advisories, agreed with them more, and responded to them faster.

In the study by Regtuit et al. [2018] a RL agent was developed that could replicate human-like CD&R strategies based on ATC 'best practices' in simple two-aircraft conflict situations. Although the study showed promising results, the personalization of solutions and controller acceptance were not considered. A follow-on study by Van Rooijen et al. [2020] aimed to achieve personalization by developing an individual prediction model of conflict resolutions based on pixel data from interface images of conflict situations captured. That study used a visual representation of velocity obstacles, in combination with a tailored Convolutional Neural Network (CNN), to predict controller solutions based on observed controller data collected in an ATC simulation. Results indicated that controller consistency and the selected (visual) feature(s) in conflict resolution play important roles in prediction accuracy. Of particular interest was that the personalised ML models performed better than the average group models.

Another design approach to conformal automation consists of adapting the automation's behavior according to the operator's cognitive states in real time. This can be achieved by means of the automation assessing the operator's physiological measures from monitoring, for example, heart rate, blink rate, fixations, respiration rate, and brain waves, to understand the operator's cognitive state (e.g. stress, workload, attention, fatigue etc) [Parasuraman 2012; Kistan 2018; Arico 2020]. Kistan et al [2018] proposed a cognitive human-machine interface (C-HMI) for ATM that adapts the information provided and functions available based on an inferred understanding of the individual's cognitive state. In the NINA project (neurometrics indicators for ATM), controller's brain activities (electroencephalogram), heart rate (electrocardiogram), and eyeblink rates (electrooculogram) were measured to assess their performance (cognitive resources) and evaluate their learning progress in en-route ATC simulations [Borghini, 2014]. Such approaches can be used to improve learning and training efficiency on a personalized level. In [Ohneiser, 2019] eye-tracking equipment was used to monitor the controller's visual behavior to determine if the operator was missing potentially important situations. When the operator's gaze was found to be elsewhere, the system drew attention to the situation at hand by highlighting information (visual cues) in the interface (e.g. a handover event) in three escalation levels.

5.3 Automation transparency

This chapter considers transparency as explored in the human factors and cognitive engineering domains. While these research communities and the AI communities view and define transparency similarly, the human factors and cognitive engineering domains have focused on the end-user's requirements for transparency. The algorithmic and mathematical methods for achieving transparency have not been at focus, rather the objective has been to determine what needs to be explained (e.g. algorithm, uncertainty, goals), how it should be presented (e.g. in text or graphics), and to explore the impact of transparency on acceptance, use, workload, situation awareness, trust, performance etc. Section 5.3.1 in this chapter explains the theoretical foundations underlying automation transparency. Section 5.3.2 presents two models proposed in the literature for designing automation transparency, and some empirical research findings. Section 5.3.3 wraps up this chapter with an introduction to the Ecological Interface Design (EID) approach.

5.3.1 Theoretical foundations

The notion of automation transparency has been researched broadly across many domains, including human-computer interaction, human factors, and numerous AI subdomains. Differences in the type of automation explored and requirements for understanding that automation has given rise to different terms across domains. What is shared between them is that all strive to make aspects of the inner workings of the automation 'black box' understandable to the human. Examples from the human factors and cognitive engineering domains include automation [Westin et al., 2016] and agent transparency [Chen et al., 2014], automation visibility [Dorniech et al., 2017], understandability [Sheridan, 1992], observability [Woods, 1996] and comprehensibility [Campbell et al., 2016]. In the AI community, the notion of transparency is mirrored in terms of ML interpretability [Murdoch et al., 2019], Explainable AI (XAI) [Gunning, 2017] and intelligibility of context-aware systems [Bellotti and Edwards, 2001]. In the AI sub-domain recommender system*s* research, the term transparency is used [Ricci et al., 2015].

Automation transparency has been defined as: "the ability of the automation to afford understanding and predictions about its behavior" [Westin et al., 2016]. In this regard, automation transparency is a property of the automation. The human reaction to this property can be assessed in terms of understanding and predicting the automation's behavior. Transparency is a multifaceted construct that ultimately is shaped by what is sought to be understood: what the human is trying to understand governs what needs to be explained.

In the human factors and cognitive engineering domains, two models of agent transparency have been proposed for autonomous systems; Lyon's [2013] model for human-robot teaming approach transparency from the requirement of establishing a shared intent and awareness between robot/autonomous systems and humans, and Chen et al.'s [Chen et al., 2014] situation awareness-based agent transparency (SAT) model for human-agent teaming. The SAT model specifies three levels of transparency, paralleling the SA levels of perception, comprehension, and predictions.

5.3.2 Research on automation transparency

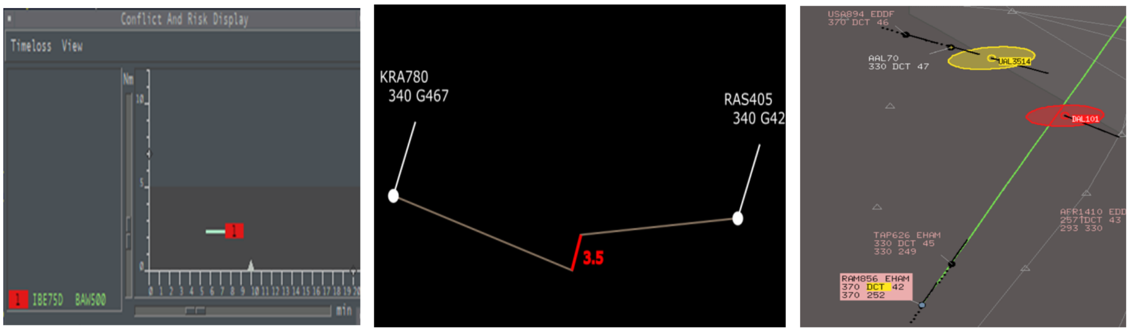
Although there is more research into transparency (and related concepts such as XAI) than into conformance, no previous research exploring ML approaches to CD&R have considered system transparency. More generally, however (i.e. away from CD&R), experiments have shown that automation transparency can benefit humans' understanding, trust, and acceptance of automation, while also improving performance (e.g. [EAAIHLG, 2020]). Consequently, transparency has been considered for many different reasons, such as to explain abnormal automation behavior [Kim, 2006], why the automation might err [Dzindolet, 2003], the behavior of intelligent agents and autonomous robots [Core et al., 2016] [Selkowitz et al., 2015], as an indication of the automation's reliability [Jamieson et al., 2008], and the automation's proximity to its performance envelope [Helldin et al., 2013]. In a literature review on agent transparency, Bhaskara et al. [Bhaskara et al., 2020], investigated the effects of transparency on performance, response time, subjective workload, situation awareness, trust, and usability. Five studies were reviewed, all of which explored military applications and most of which operationalized transparency on the SAT model.

Bhaskara et al. distinguished between four levels of transparency: low (supporting perception in terms of basic information or advice), medium (supporting comprehension in terms of adding information about the agent's reasoning), high (supporting prediction in terms of adding information on either expected outcome/consequence or added reasoning for a recommendation), and very high (supported prediction by adding more information in the high transparency level (e.g., both an outcome prediction and uncertainty information). While findings show some benefits of increased (levels of) agent transparency to acceptance, trust, situation awareness, workload, and response time, Bhaskara et al. noted that results so far are inconclusive and that there are simply not enough results to form stable conclusions [Bhaskara et al., 2020].

Researchers have explored transparency of intelligent agents (e.g., robots) by providing explanations based on the agent's decision-making processes. Frameworks such as the Belief-Desire-Intention (BDI) [Rao and Georgeff, 1995], Partially Observable Markov Decision Process (POMDP) [Wang et al. 2016], and Parallel-rooted, Ordered, Slip-stack Hierarchical (POSH) frameworks [Theodorou et al., 2017] have been used for modeling the decision-making processes and actions of agents. The framework used to model the mental process of the agent can also be used to provide explanations of the agent's behavior. Generalized, all three frameworks model the agent's behavior against three components: *goals* for which *actions* are accomplished based on an understanding of the current *state*. For increasing the user's understanding of the agent and its conduct, transparency of each component can be afforded. In experiments with an agent, built on the POMDP framework, explanations of the agent's inferred state (based on its sensors) and associated uncertainty were shown to benefit understanding, trust, mission success, and percentage correct decisions made, particularly when the agent's reliability was low.

5.3.2.1. Early ATC interface designs for transparency

An important aspect for the success of any transparency measure is the way in which the information is communicated. In ATM and many other transportation systems, visual human-machine interfaces are seen as effective ways to communicate aspects of the work domain to human agents. In ATC CD&R, the majority of visual interfaces that have been developed over the past decades (e.g., as part of PHARE and Tactical Controller Tools) put the emphasis on portraying problematic areas for the air traffic controller to solve. That is, the computer predicts conflicts using flight data and expected traffic patterns and indicates conflicts on either the electronic radar screen and/or a secondary interface, see figure 5.3. The role of the air traffic controller is then to observe the problem and subsequently formulate and implement actions to solve the problem.



Conflict and Risk Display (CARD) | Relative State Vector | Hyper Interactive Problem Solver (HIPS)

Fig. 5.3. Visual interfaces for CD&R. CARD display is a secondary display, showing time- and distance-to-CPA of on a 2D axis system, whereas the relative state vector and HIPS are integrated on the plan view display.

The visual representations mainly aim to direct the controller's attention to conflicts that need to be resolved. Besides the HIPS, the CARD and relative state vector assist mainly in conflict detection, not necessarily resolution. The HIPS display portrays loss of separation areas on the radar screen with "blobs", which not only indicate the presence of a conflict (when the trajectory intersects the blob), but also implicitly hint towards heading solutions that would make any updated trajectory pass the blob to either the left or right.

These visual tools have been tested successfully in ATC evaluation studies, and some of them (e.g., CARD) are even integrated in the standard tool set in ATC centers. Despite their success, controllers also indicated room for improvements. For example, the avoidance zones in HIPS did not always correctly represent the controller's perception of the nature and severity of the problems. Sometimes there could be inconsistencies between the conflict detection algorithms and the problem-solver tool, resulting in ambiguous information causing controller distrust.

5.3.3 Ecological Interface Design

Ecological Interface Design (EID) is a design framework that originated from the process control domain. The goal of the EID framework is to create interface representations that reveal the deep structure (e.g., physical relationships) of a control problem in meaningful ways for operators to "chunk" information, reducing the demands on memory and supporting productive thinking (e.g., through direct manipulation, metaphorical design, clever geometrical shapes, etc.). Different from user- and technology-centered design, EID is founded on the starting point that humans are creative problem solvers who can adapt to novel situations, or, work domain states.

In processes governed by the laws of physics, such as air transportation, creative solutions and actions are limited and bounded. For example, an aircraft cannot sustain flight when it is flying slower than the stall speed. The turn radius of an aircraft is constrained by the maximum allowable load factor to maintain structural integrity. Besides these 'internal' aircraft constraints, the maneuverability of aircraft is also affected by 'external' static and dynamic environmental (i.e., ecological) constraints such as other air traffic and weather. These physical boundaries define the line between safe and unsafe actions that result in accidents as shown in Figure 5.4. Besides the physical boundaries, there are also intentional boundaries that add safety margins on actions. For example, in aviation the horizontal separation standard between aircraft is 5 nautical miles, which is obviously larger than the physical dimensions of any aircraft. Violating the intentional boundary will then lead to an incident. Additionally, actions may be further restricted by automation that, for example, only support one (or a small range of) optimal action(s).

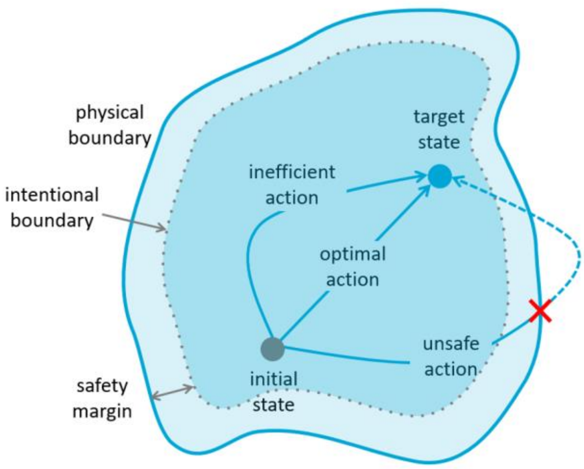
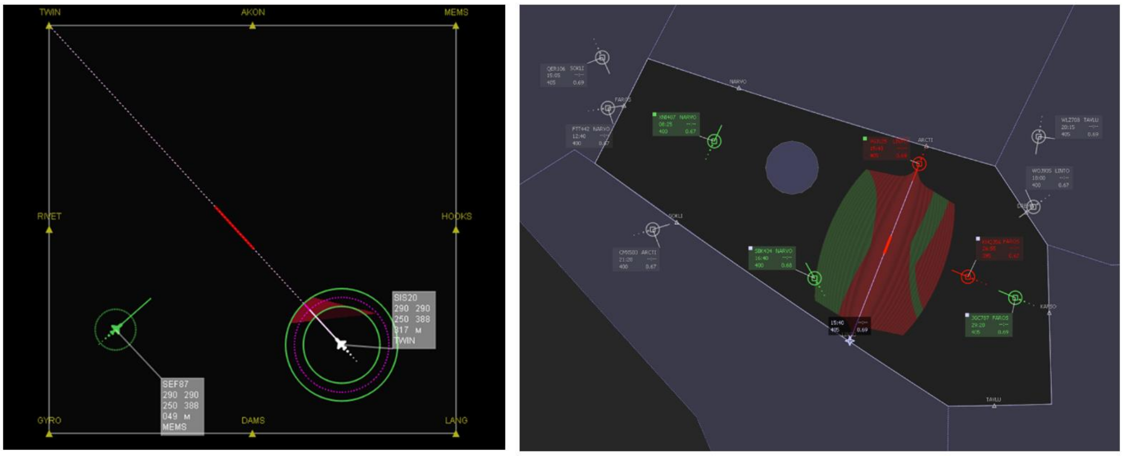


Fig 5.4. Ecological Interface Design as a constraint-based approach to interface design, which aims to support <u>any</u> action that lies within the space of possibilities bounded by physical laws and work domain principles.

In EID, the design emphasis is put on modeling and visualizing all constraints on actions onto a visual interface such that operators can literally perceive the 'space of (action) possibilities.' Doing so can help the operator decide when and by how much to deviate from standard operations and evaluate the consequences of crossing certain boundaries. The design challenge is then how to map the constraints onto a visual display in a way that supports productive thinking. Here, principles of user- and technology-centered design can be helpful in guiding both modeling (e.g., predicting the range of future vehicle states based on the equations of motion) and mapping (e.g., maintaining visual momentum, adhering to the Proximity Compatibility Principle and choosing the right reference frame for projection). Thus, EID is not meant to replace those design approaches, but rather complement them by changing the focal point of design, which is geared towards letting users "finish the design" in terms of helping them navigate through the space of possibilities in their own preferred way, whilst being aware of the boundaries on actions.

So far, EID has been successfully applied in many application domains, including aviation. In ATC, "solution space" concepts have been designed for supporting perturbation management in state-based and 4D trajectory-based operational environments, see figure 5.5. Similar to other ATC interfaces, such as the HIPS display, the solution space concept requires the human operator to take action. Thus, the computer integrates information related to flight performances, traffic patterns and environmental conditions (e.g., no-fly zones, weather cells, etc.) to calculate and portray both solution and problem areas, but does not suggest any specific action to undertake. That also means that users could still decide to implement suboptimal solutions that are safe (i.e., circumventing the problem areas within the boundary of the feasible envelope), but are also regarded suboptimal from an efficiency perspective.



(a) State-based solution space (b) Trajectory solution space

Fig 5.5. State-based solution space, showing all speed and heading options within the speed envelope of the selected aircraft that resolve the conflict (red triangle, based on 5NM protected

Founding Members

zone). The trajectory-based solution space shows safe (green) and unsafe (red) waypoint locations within the flight performance envelope that ensure the aircraft will reach the sector exit at its desired exit time in case a re-route is needed. Extensions of these visualizations in altitude have been developed as well.

5.4. Implications for MAHALO

Review confirmed that Conformance and Transparency remain two critical concepts to explore in ML based advisory automation. Conformance can be seen on a continuum from specific (personalised) to normative (group) tuning. This is an important lesson for MAHALO, since we intend to experimentally manipulate conformance. We also acknowledge the role of conformance in automation trust, acceptance, and reliance, and this view seems to have gained some popularity over the last five years. Finally, in experimentally manipulating level of conformance, MAHALO simulations can consider use of adaptive measures, such as physiological triggers (Arico, 2020; Kistan, 2018] for level-of-conformance manipulation.

Review also confirmed that the concept of transparency is growing in relevance and research focus, in the age of ML. It also is clearly seen as a multi-faceted concept, and is being modeled in different ways [Lyon, 2013; Chen, 2014]. Finally, we are encouraged in our preliminary view that the ecological interface design (EID) approach can enhance human-machine cooperation with a hybrid ML advisory system.

6. Conclusions

The goals of the MAHALO project are ambitious: to build a functioning hybrid ML CD&R system, and to evaluate the impact of this system via human-in-the-loop trials. The main aim of this SOAR was to review the latest theoretical and empirical evidence on ML methods, known impacts on human performance, and their intersection with ATM. In the process, concrete lessons have emerged.

6.1 Lessons learnt

First, **our candidate ML approaches and methods fit with the current state-of-the-art.** Given the pace of progress especially in ML research, this is encouraging.

Second, our original aim to create a hybrid ML approach seems, after review, **to fit well with a model that distinguishes conflict detection (CD) and conflict resolution (CR) functions.**

Third, **a combination of SL and RL approaches seems to fit the CD and CR functions** especially well.

6.2 Next steps

Additionally, review has begun (the process is ongoing) to identify the **'rules of the road' control strategies** that must be captured in the RL module for CR.

Finally, review has begun to narrow our focus on **specific methods** within the broader ML approaches (i.e., SL and RL).

During the review, some effort has been dedicated to understand what results coming from past EU-funded projects way be relevant for our project. In the MAHALO original technical proposal, we identified several projects that would play foundational parts in refining the theory and methods of this project. Among these were five SESAR projects, under either the WP-E or ER1 umbrella. These five were rated medium or high in terms of their impact on MAHALO. Table 6.1 below shows **how we intend to draw upon these five projects.**

Project	Specific contribution to MAHALO
MUFASA	• Operationalising conformance • Refining ConOps and test scenarios • Automation acceptance / trust framework • Data collection subjective methods • Experimental design • Advisory alert issues (tweaking timing, logic, etc)
C-SHARE	• Ecological UI design • Designs for transparency • Advisory / alerting logic • Experimental design
TERRA	• Candidate SL methods for CD • Synthetic traffic generation methods • Data requirements for SL models • Offline and online post-hoc analysis methods • RL for safety modelling (data requirements, analysis methods)
STRESS	• Off-nominal mode analysis • Human performance assessment methods
NINA	• Neuro-ergonomic indicators • Adaptive triggering mechanisms and design issues • ConOps and scenario design guidance • Experimental design guidance.

Table 6.1 Specific contributions of five SESAR projects to MAHALO.

6.3 Addressing the original research questions

Section 1.2.3.1 laid out the following eight broad questions, which the SOAR sought to address:

ML issues:

1. Broadly speaking, how should we model the CD&R process?

2. Which candidate ML approaches seem most promising for modelling CD&R?

3. What main challenges exist in the use of ML methods, and how can we counter these?

Human and system impacts:

4. Do transparency and conformance constructs still seem critical, and how feasible are they to experimentally explore?

5. How can transparency be operationalised in simulation trials?

6. How can conformance be operationalised in simulation trials?

7. How do we assess other aspects of human performance, including trust, acceptance, and workload?

Interface issues:

8. How should we address display issues to convey transparency and conformance?

6.3.1 ML issues (research questions 1-3)

Figuring out which type of ML algorithm should be used for the CD and CR tasks is a non-trivial task. On the basis of literature review, we confirmed that our preliminary view that the two functions can be separated, is feasible and preferable. The CD&R process will likely be modelled as two separate tasks, a classification task for CD, and an optimization task for CR, with CD output driving CR function input.

The specific approach (i.e. model architecture) will be chosen over the course of task 2.3 (Concept and scenario definition). From the literature review it became clear that SL has an edge over RL when it comes to the task of CD, this is due to SL's great strength at detecting patterns in data when trained from labelled data. SL has been used in CD tasks, but its performance has not always been very high (80%). For this reason, we will continue to explore candidate CD architectures over the coming months. CD candidates approaches include SL, algorithmic approach, and ganged (sequential) combinations of architectures. For the CR task, current thinking is that an RL-based model will be used, in which the agent will optimize based on both flight path deviation (or additional miles flow, fuel consumed, less is better approach) and conformance to ATCO demonstrations

For both CD and CR, several challenges were identified (as summarised in chapter 4.4). Chief among these is the requirement of ML to have access to large data sets. One solution to this issue is adapting the methods of vsn Rooijen et al [2020], in which human-in-the-loop data seeded creation of synthetic traffic, via the introduction of stochastic noise. This and other challenges will be discussed more fuller in deliverable 2.2, but thus far, literature review has encouraged us that our general ML approach seems feasible and potentially valuable.

6.3.2. Human – system impacts (research questions 4-7)

Review of the literature on conformance and transparency, led to the conclusion that they remain as critical as ever, in the emerging age of machine learning. The two concepts seem to play different roles in enhancing human-machine interaction. Conformance seems a means to foster initial acceptance and trust, and transparency seems a means to expand the human's understanding. As shown in figure 6.1 (based on figure 1.1), a novice might initially work with a conformal transparent system (in the lower right cell). Learning, in effect, can be operationalised as moving the operator from the bottom right to the top right cell—that is, moving the operator into an initially non-conformal way of working, that is hopefully more optimal. Teaching is, after all, making the unfamiliar familiar.

		TRANSPARENCY	
		Low	High
CONFORMANCE	Low	• Different solution than individual • Solution not explained	• Different solution than individual • Solution is explained
	High	• Same solution as individual • Solution not explained	• Same solution as individual • Solution is explained

Figure 6.1 Learning in a human-machine system.

In terms of operationalising conformance, we are encouraged to use the distinction between individual and group tuning (van Rooijen, 2020), and also to explore adaptive triggering of conformance level. Conformance is intended to be a naturally-occurring, non-manipulated variable. Earlier research using "Wizard of Oz" techniques simulated conformance by using replaying self- or other generated solutions. Part of the aim is to assess the performance of automation using a non-manipulative approach. An ATCO's conformance rating for a given ML solution defines the conformance level of that solution for that controller.

On a related point, MAHALO requires that the CD&R system be able to determine if a resolution, derived by the system, is conformal to the individual controller or not. If the solution is determined to be non-conformal, the system should provide an explanation to the individual controller as to why the system believes the solution to be non-conformal and why the proposed solution is considered better. This is considerably more complicated than to provide an explanation for why a particular solution is chosen, which does not require an assessment of the individual's preferences. The system must be able to:

- Derive a best solution,
- Determine the controller's preference s for CD&R,
- Compare the derived solution with the solution preferred by the controller,
- Determine if the derived solution is non-conformal (i.e. differs from the solution preferred by the controller),
- Explain to the controller why the system considers the derived solution to be 'better' than the controller's preferred solution.

Transparency will be operationalised as explainability along two dimensions:

- How the ML system arrived at a particular solution (i.e., process and relationship between input and output), and
- Why the ML system has derived that a particular conflict resolution matches the individual's preferences (i.e., why it is strategic conformal).

The concepts of transparency and conformance are partly at odds. Both automation transparency and conformance can foster human understanding and acceptance of automation advisories. The literature, however, indicates that the two constructs achieve this objective in different ways.

Strategic conformance fosters acceptance based on how similar the system's strategy or solution (i.e., output) is with the human's preferred solution. Since the conformance of a system is judged based on its apparent behavior, the human will have difficulties understanding in detail what the system does when solving a problem, how it solves a particular problem, and why a particular solution is chosen. If an ML system proposes a solution in line with the controller's strategy, the human may assume that the system will use a similar method as the human and apply a similar underlying strategy. In an ATC CD&R task, the strategic conformance of the ML system can relate to the conflicts detected by the system or the conflict resolutions proposed by the system.

In contrast, automation transparency strives to explain aspects of the automation's underlying reasoning and decision-making process, including for example how the automation perceives the environment and problem, the algorithm (i.e. method) used for addressing the problem, how a specific solution (i.e prediction) was derived based on the input, or the goals pursued for finding a solution. As such, automation transparency fosters acceptance by providing an explanation for what the system does when solving a problem (e.g., explain the ML model or automation algorithm), how the automation will solve the specific problem (e.g., explain the relationship between input and output), and why this particular solution is chosen (e.g., because the following factors are considered/weighed most important for determining the output: a, b, x, and z).

The benefits of transparency might only emerge when automation offers a solution different from that of the human (i.e., non-conformal). An important challenge to solve is then for automation to determine if a proposed solution differs from the human, or else an explanation may not be relevant. This is perhaps where conformance becomes most valuable. Previous literature has explored physiological measures of the operator's individual state [Parasuraman 2012; Kistan 2018; Arico 2020; [Borghini, 2014; Ohneiser, 2019] as an input to automation for adapting its behavior. Similarly, by deriving understanding of an operator's preferred strategy or solution type, automation could determine if the solution it finds most optimal differs from that of the individual, which would warrant an explanation. By knowing the individual's solution preferences, the system will be in a better position to afford transparency and explain its reasoning for recommending another (non-conformal) solution. The system will be able to provide an argument for why the proposed solution is better than that of the individual.

6.3.3. Interface issues (research question 8)

ATM tools have thus far focused more on supporting CD than CR. Systems like TCAS or short-term conflict alert can automate CD, but this far we have been reluctant to automate the CR process. Conflict resolution is left entirely in the hands of the controller. Although this may be desired from a controller's perspective, differences in controller's "satisficing" strategies could counteract the efficiency targets set for the future ATM system. However, increasing the level of automation involvement at the levels of making control decisions and implementing actions has not been embraced by the ATC community amongst controllers. It seems that when automation starts to mingle in taken actions, for example by suggesting single optimized solutions, controllers reject such computer-based advice as they cannot always understand why and how the computer arrived at that solution.

In MAHALO, it will thus be important to design visual representations that would allow controllers to not only perceive problem areas, but also the complete set of solutions with additional information on why a computer would prefer one particular solution over all other possible solutions within the space of possibilities. Designing visual representations guided by the Ecological Interface Design (EID) paradigm could potentially address this issue.

6.4 Final wrap-up

Again, the goals of MAHALO project are ambitious: to build high-performance hybrid ML CD&R automation, and to evaluate the human-machine system impacts of this automation via real time simulations. In the process, we have **set up the potential for competing goals:** If forced, would we sacrifice system performance to accomplish human-machine testing? Do we compromise testing to maximise system performance?

First, we aim to build a hybrid CD / CR system, that is fundamental to the MAHALO vision, and literature review revealed nothing to discourage that view.

Second, we aim to develop as high performance a system as possible. Review revealed that Sl might not be up to standards for this CD function. For this reason, we will continue to explore (in Task 2.3) specific architectures that can be sued for the CD function.

Third, we want to pursue the development of (perhaps among others) SL methods for this CD function. Part of the aim here would be to refined and extend previous work into six different SL methods for CD&R, which this SOAR revealed.

The next report in our series, D2.3, will present how successful the MAHALO team was in refining these methods.

Bibliography

Abdul, A., J. Vermeulen, D. Wang, B. Y. Lim, and M. Kankanhalli (2019). Trends and trajectories for explainable, accountable and intelligible systems: An HCI research agenda, CHI Conference on HF in Computing Systems, Montreal QC, Canada, Apr. 21-26, 2018.

Adella Bhaskara , Michael Skinner, and Shayne Loft. Agent Transparency: A Review of Current Theory and Evidence. IEEE TRANSACTIONS ON HUMAN-MACHINE SYSTEMS, VOL. 50, NO. 3, JUNE 2020.

Al-Shedivat, M., K. A. Dubey, E. P. Xing (2017)., Contextual explanation networks, arXiv e-prinst: 1705.10301.

Alvarez-Melis, D., & T. S. Jaakkola (2018). On the robustness of interpretability methods, 3rd ICML WHI, Stockholm, Sweden, Jul. 14, 2018

Aricò, P., Reynal, M., Di Flumeri, G., Borghini, G., Sciaraffa, N., Imbert, J.-P., . . . Babiloni, F. (2019). How Neurophysiological Measures Can be Used to Enhance the Evaluation of Remote Tower Solutions. Frontiers in Human Neuroscience, 13(303). doi:10.3389/fnhum.2019.00303

Arulkumaran, K., Deisenroth, M. P., Brundage, M., and Bharath, A. A., "Deep Reinforcement Learning: A Brief Survey," in *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 26-38, Nov. 2017, doi: 10.1109/MSP.2017.2743240.

Asadi, R., Mustapha, N., Sulaiman, N., & Shiri, N. (2009). New supervised multi layer feed forward neural network model to accelerate classification with high accuracy. European Journal of Scientific Research, 33(1), 163-178

Bach, S. A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek (2015). On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation, PLOS ONE, 10(7), pp. e0130140, 2015.

Balfe, N., Sharples, S., & Wilson, J. R. (2015). Impact of automation: measurement of performance, workload and behaviour in a complex control environment. Applied Ergonomics, 47, 52–64.

Bellotti, V. & K. Edwards (2001). Intelligibility and accountability: Human considerations in context-aware systems, Hum.-Comput. Interact., 16(2), 2001, pp. 193-212.

Beta Writer (2019). Lithium-Ion Batteries, A Machine-Generated Summary of Current Research. Beta Writer. Springer, Cham, 2019. DOI https://doi.org/10.1007/978-3-030-16800-1

Bhaskara, A., M. Skinner, and S. Loft (2020). Agent transparency: A review of current theory and evidence, IEEE Trans. Human–Mach. Syst., 50(3), 2020, pp. 215-224.

Bojarski, M., A. Choromanska, K. Choromanski, B. Firner, L. Jackel, U. Muller, and K. Zieba (2016). VisualBackProp: Efficient visualization of CNNs, arXiv e-prints: 1611.05418.

Borghini, G., Aricò, P., Graziani, I., Salinari, S., Babiloni, F., Imbert, J.-P., . . . Pozzi, S. (2014, 2014-11-25). Analysis of neurophysiological signals for the training and mental workload assessment of ATCos. Paper presented at the SID 2014, 4th SESAR Innovation Days, Madrid, Spain.

Brittain, M., & Wei, P. (2019). Autonomous Air Traffic Controller: A Deep Multi-Agent Reinforcement Learning Approach. arXiv e-prints: 1905.01303.

Brittain, M., & Wei, P. (2019). One to Any: Distributed Conflict Resolution with Deep Multi-Agent Reinforcement Learning and Long Short- Term Memory.

Brittain, M., Yang, X., & Wei, P. (2020). A Deep Multi-Agent Reinforcement Learning Approach to Autonomous Separation Assurance. arXiv eprints: 2003.08353.

Calvo-Fernández, E., Perez-Sanz, L., Cordero-García, J. M., & Arnaldo-Valdés, R. M. (2016). Conflict-Free Trajectory Planning Based on a Data-Driven Conflict-Resolution Model. Journal of Guidance, Control, and Dynamics, 40(3), 615-627. doi:10.2514/1.G000691

Campbell, J. L. Brown, G. J. S, C. M. Richard, M. G. Lichty, T. Sanquist, P. Bacon, R. Woods, H. Li, D. N. Williams, and J. F. Morgan (2016). Human factors design guidance for driver-vehicle interfaces, National Highway Traffic Safety Administration, Washington, DC DOT HS 812 360, Dec. 2016.

Campbell, M. (1999). Knowledge Discovery in Deep Blue. Communications of the ACM,42(11), 65-67,Nov1999.

Chen, J.Y.C., K. Procci, M. Boyce, J. Wright, A. Garcia, and M. Barnes (2014). Situation awareness–based agent transparency, ARL, Aberdeen Proving Grounds, MD ARL-TR-6905, Apr. 2014.

Chu, T., Wang, J., Codecà, L., and Li, Z. "Multi-Agent Deep Reinforcement Learning for Large-Scale Traffic Signal Control," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 3, pp. 1086-1095, March 2020, doi: 10.1109/TITS.2019.2901791.

Claudatos, L. (2018). The Effect of Automating Routine Tasks on Air Traffic Controller Conflict Detection Performance. Master's Thesis. 4933.DOI: https://doi.org/10.31979/etd.q3dt-95n9https://scholarworks.sjsu.edu/etd_theses/4933

Core,M.G., H. C. Lane, M. van Lent, D. Gomboc, S. Solomon, and M. Rosenberg (2016). Building explainable artificial intelligence systems, 18th IAAA Conference on Innovative Applications of Artificial Intelligence - Volume 2, Boston, MA, Jul. 2016, pp. 1766-1773.

D'Arcy, J. F., & Della Rocco, P. S. (2001). Air Traffic Control Specialist Decision Making and Strategic Planning-A Field Survey. http://oai.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=ADA389823

Dam, S.V.; Mulder, M.; Paassen, R. The Use of Intent Information in an Airborne Self-Separation Assistance Display Design. In AIAA Guidance, Navigation, and Control Conference; American Institute of Aeronautics and Astronautics: Chicago, IL, USA, 2009.

Dorneich, M.C., R. Dudley, E. Letsu-Dake, W. Rogers, S. D. Whitlow, M. C. Dillard, and E. Nelson (2017). Interaction of automation visibility and information quality in flight deck information automation, IEEE Trans. Human-Mach. Syst., vol. 47, 2017, pp. 915-926.

Dzindolet, M.T., S. A. Peterson, R. A. Pomranky, L. G. Pierce, and H. P. Beck (2003). The role of trust in automation reliance, Int. J. Hum. Comput. Stud., 58(6), 2003, pp. 697-718.

EAAIHLG (2020) The FLY AI Report. Demystifying and Accelerating AI in Aviation/ATM. European Aviation Artificial Intelligence High Level Group, Mar. 5. https://www.eurocontrol.int/publication/fly-ai-report

EASA (2020) Artificial Intelligence Roadmap. A Human-Centric Approach to AI in Aviation. European Union Aviation Safety Agency, Feb. 2020. https://www.easa.europa.eu/sites/default/files/dfu/EASA-AI-Roadmap-v1.0.pdf

Edwards, L. & M. Veale (2017). Slave to the algorithm? Why a 'right to an explanation' is probably not the remedy you are looking for, Duke L. & Tech. Rev., 18, 2017

F.Doshi-Velez, F. & B. Kim (2017). Towards a rigorous science of interpretable machine learning, arXiv e-prints: 1702.08608v2

Federal Aviation Administration (2019), FAA Aerospace Forecast, Fiscal Years 2020-2040, Retrieved from https://www.faa.gov/data_research/aviation/aerospace_forecasts/media/FY2020-40_FAA_Aerospace_Forecast.pdf

Fothergill, S., & Neal, A. (2008). The Effect of Workload on Conflict Decision Making Strategies in Air Traffic Control. Proceedings of the Human Factors and Ergonomics Society 52nd Annual Meeting, 52(1), 39–43. https://doi.org/10.1177/154193120805200110

Ghosh, S., Laguna, S., Lim, S. H., Wynter, L., & Poonawala, H. (2020). A Deep Ensemble Multi-Agent Reinforcement Learning Approach for Air Traffic Control. arXiv e-prints: 2004.01387.

Gigerenzer, G., Hertwig, R. & T. Pachur (2011). Heuristics: The Foundations of Adaptive Behavior, New York, NY: Oxford University Press.

Guidotti, R., A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi (2018). A survey of methods for explaining black box models, ACM Comput. Surv., 51(5), Article 93.

Gunning, D. (2017). Explainable Artificial Intelligence (XAI), Defense Advanced Research Projects Agency (DARPA), 2017. Retrieved from https://www.darpa.mil/attachments/XAIProgramUpdate.pdf [Online].

Hao, S.; Cheng, S.; Zhang, Y. A multi-aircraft conflict detection and resolution method for 4-dimensional trajectory-based operation. Chin. J. Aeronaut. 2018, 31, 1579–1593.

Helldin, T., G. Falkman, M. Riveiro, and S. Davidsson (2013). Presenting system uncertainty in automotive UIs for supporting trust calibration in autonomous driving, 5th International Conference on Automotive User Interfaces and Interactive Vehicular Applications, Eindhoven, The Netherlands, Oct. 2013.

Helmke, H. (2018).Machine learning of speech recognition models for controller assistance (MALORCA): Final project results report. D5-3, 2018, Retrieved from: https://www.sesarju.eu/projects/malorca [Online].

Hilburn, B., C. Westin, and C. Borst. (2014). Will controllers accept a machine that thinks like they think? The role of strategic conformance in decision aiding automation, Air Traffic Control Q., 22(2), 2014, pp. 115-136.

Hoekstra, J.; Ellerbroek, J. BlueSky ATC Simulator Project: An Open Data and Open Source Approach. In Proceedings of the Conference: International Conference for Research on Air Transportation, Philadelphia, PA, USA, 20–24 June 2016.

Hoekstra, J.; van Gent, R.; Ruigrok, R. Designing for safety: The 'free flight' air traffic management concept. Reliab. Eng. Syst. Saf. 2002, 75, 215–232.

Hongming Z. and Tianyang Y.. Taxonomy of Reinforcement Learning Algorithms, pages 125–133.

Springer Singapore, Singapore, 2020. ISBN 978-981-15-4095-0. doi: 10.1007/978-981-15-4095-0_3. URL https://doi.org/10.1007/978-981-15-4095-0_3.

Lyons, J.B. (2013). Being transparent about transparency: A model for human-robot interaction, AAAI Spring Symposium on Trust in Autonomous Systems, Palo Alto, CA, Mar. 25-27, 2013, pp. 48-53.

Jamieson,G.A., L. Wang, and H. F. Neyedli (2008). Developing human-machine interfaces to support appropriate trust and reliance on automated combat identification systems, Cognitive Eng. Lab, University of Toronto, Contract Report W7711-068000/001/TOR, Mar. 2008.

Jenie, Y.I., E. J. van Kampen, J. Ellerbroek, and J. M. Hoekstra (2017). Taxonomy of Conflict Detection and Resolution Approaches for Unmanned Aerial Vehicle in an Integrated Airspace. IEEE Trans. Intell. Transp. Syst. 2017, 18, pp. 558–567.

Jia Wu, Xiu-Yun Chen, HaoZhang, Li Dong Xiong, Hang Lei, Si-HaoDeng. Hyperparameter Optimization for Machine Learning Models Based on Bayesian Optimization. Journal of Electronic Science and Technology Volume 17, Issue 1, March 2019, Pages 26-40

Juozapaitis, Z., Koul,A., Ferm, A.,Erwig, M. & Doshi-Velez, F. (2019). Explainable Reinforcement Learning via Reward Decomposition, IJCAI/ECAI Workshop on Explainable Artificial Intelligence, 47-53, 2019

Juozapaitis,Z., A. Koul, A. Fern, M. Erwig, and F. Doshi-Velez (2019). Explainable reinforcement learning via reward decomposition, 28th IJCAI. Workshop on Explainable Artificial Intelligence., Macau, China, Aug. 11, 2019, pp. 47-53.

Kallus, K. W., Van Damme, D., & Dittmann, A. (1999). Integrated Task and Job Analysis of Air Traffic Controllers - Phase 2 : Task Analysis of En-route Controllers. Eurocontrol, 1–98.

Kim, K., Hwang, I., & Yang, B.-J. (2016). Classification of Conflict Resolution Methods using Data-Mining Techniques 16th AIAA Aviation Technology, Integration, and Operations Conference: American Institute of Aeronautics and Astronautics.

Kim, T. & P. Hinds (2006). Who should I blame? Effects of autonomy and transparency on attributions in human-robot interaction, 15th IEEE ROMAN, Hatfield, UK, Sep. 6-8, 2006, pp. 80-85.

Kirwan, B. & M. Flynn, (2002). Investigating air traffic controller conflict resolution strategies, Report asa.01.cora.2.del04-b.rs. EUROCONTROL. Brussels, Belgium, Mar. 2002.

Kirwan, B., & Flynn, M. (2002). Investigating Air Traffic Controller Conflict Resolution Strategies. In EUROCONTROL, ASA.01.CORA.2.DEL04-B.RS

Kistan, T., Gardi, A., & Sabatini, R. (2018). Machine Learning and Cognitive Ergonomics in Air Traffic Management: Recent Developments and Considerations for Certification. Aerospace, 5(4), 103.

Kotsiantis, S.B. Decision trees: a recent overview. Artificial Intelligence Review 39, 261–283 (2013). https://doi.org/10.1007/s10462-011-9272-4

Koyuncu, E. & B. Başpınar (2017). Demand and capacity balancing through probabilisticqQueuing theory and ground holding program for european air transportation network, Anadolu Uni. J. Sci. Technol. A – Appl. Sci. Engi., 18(2), 2017, pp. 360-374.

Kuchar, J.F. & L. C. Yang. (2000). A review of conflict detection and resolution modeling methods. IEEE Transactions on Intelligent Transportation Systems, 1(4), 2000, pp. 179-189.

Kuderer, M., S. Gulati, and W. Burgard. (2015). Learning driving styles for autonomous vehicles from demonstration, IEEE ICRA, Seattle, WA, May 26-30, 2015.

Lee, C.K.H. (1943). A review of applications of genetic algorithms in operations management, Engineering Applications of Artificial Intelligence, Volume 76, 2018, Pages 1-12, ISSN 0952-1976, https://doi.org/10.1016/j.engappai.2018.08.011.

H. Liang, H., Zhang, X., Zhang, J., Li, Q., Zhou, S., and Zhao, L., "A Novel Adaptive Resource Allocation Model Based on SMDP and Reinforcement Learning Algorithm in Vehicular Cloud System," in *IEEE*

Transactions on Vehicular Technology, vol. 68, no. 10, pp. 10018-10029, Oct. 2019, doi: 10.1109/TVT.2019.2937842.

Liang, M., Li, W., Delahaye, D., & Notry, P. (2019, 8-12 Sept. 2019). Policy Optimization in Automated Point Merge Trajectory Planning: An Artificial Intelligence-based Approach. Paper presented at the 2019 IEEE/AIAA 38th Digital Avionics Systems Conference (DASC).

Liu, Y., Y. Lee, an A. N. K. Chen. (2011). Evaluating the effects of task–individual–technology fit in multi-DSS models context: A two-phase view, Decision Support Syst., 51(3), 2011, pp. 688-700.

Manzey, D., Reichenbach, J., & Onnasch, L. (2012). Human performance consequences of automated decision aids: the impact of degree of automation and system experience. Journal of Cognitive Engineering and Decision Making, 6(1), 57–87. https://doi.org/10.1177/1555343411433844

Marcos, R., O. G. Cantú Ros, and R. Herranz (2017). Combining visual analytics and machine learning for route choice prediction: Application to pre-tactical traffic forecast, SID, Belgrade, Serbia, Nov. 28-30, 2017.

McCulloch, W.S., Pitts, W. A logical calculus of the ideas immanent in nervous activity. Bulletin of Mathematical Biophysics 5, 115–133 (1943). https://doi.org/10.1007/BF02478259

Mercer, J., Gabets, C., Gomez, A., Edwards, T., Bienert, N., Claudatos, L., & Homola, J. (2017). How important is conflict detection to the conflict resolution task? In N. A. Stanton, S. Landry, G. Di Bucchianico, & A. Vallicelli (Eds.), Advances in Human Aspects of Transportation (Vol. 484, pp. 103–115). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-41682-3_9

Mercer, J., Gomez, A., Gabets, C., Bienert, N., Edwards, T., Martin, L., ... Homola, J. (2016). Impact of automation support on the conflict resolution task in a human-in-the-loop air traffic control simulation. IFAC-PapersOnLine, 49(19), 36–41. https://doi.org/10.1016/j.ifacol.2016.10.458

Meeuwen, L. W. van, Jarodzka, H., Brand-Gruwel, S., Kirschner, P. a., de Bock, J. J. P. R., & van Merriënboer, J. J. G. (2014). Identification of effective visual problem-solving strategies in a complex visual domain. Learning and Instruction, 32, 10–21. https://doi.org/10.1016/j.learninstruc.2014.01.004

Mnih, V., K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, & M. Riedmiller (2013). Playing Atari with Deep Reinforcement Learning. NIPS Deep Learning Workshop 2013. arXiv:1312.5602

Mollinga, J., & van Hoof, H. (2020). An Autonomous Free Airspace En-route Controller using Deep Reinforcement Learning Techniques. arXiv eprints: 2007.01599.

MUFASA (2013). E.02.08 – MUFASA Final Project Report WP-E. SESAR SJU.

Murdoch, W.J., C. Singh, K. Kumbier, R. Abbasi-Asl, and B. Yu (2019). Definitions, methods, and applications in interpretable machine learning, PNAS USA, 116(44), 2019, pp. 22071-22080.

Murdoch, W.J., P. J. Liu, and B. Yu (2018). Beyond word importance: Contextual decomposition to extract interactions from LSTMs, arXiv e-prinst: 1801.05453.

Nguyen, T.T., Nguyen, N.D., and Nahavandi, S. (2020). Deep Reinforcement Learning for Multiagent Systems: A Review of Challenges, Solutions, and Applications. IEEE Trans. Cybern, 50(9), 3826-3839. doi:10.1109/TCYB.2020.2977374

Ohneiser, O., Gürlük, H., Jauer, M.-L., Szöllősi, Á., & Balló, D. (2019, Dec. 2-5). Please have a Look here: Successful Guidance of Air Traffic Controller's Attention. Paper presented at the 9th SESAR Innovation Days, Athen, Greece.

Ong, H. Y., & Kochenderfer, M. J. (2017). Markov Decision Process-Based Distributed Conflict Resolution for Drone Air Traffic Management. Journal of Guidance, Control, and Dynamics, 40(1), 69-80. doi:10.2514/1.g001822

Parasuraman, R. and Jiang, Y. (2012). Individual differences in cognition, affect, and performance: Behavioral, neuroimaging, and molecular genetic approaches, Neuroimage, 59(1), 70-82.

Parasuraman, R., Sheridan, T.B. & Wickens, C.D. (2000). A model for types and levels of human interaction with automation. IEEE Transactions on Systems Man and Cybernetics- Part A: Systems and Humans. 30(3), 286-297.

Pelegrín, M., & D'Ambrosio, C. (2020). Airspace conflict resolution: A unifying mathematical framework and review. Retrieved from https://hal.archives-ouvertes.fr/hal-02902566

Pham, D., Tran, N. P., Alam, S., Duong, V., & Delahaye, D. (2019a). A machine learning approach for conflict resolution in dense traffic scenarios with uncertainties. Paper presented at the 13th USA/Europe ATM R&D Seminar, Jun. 18-19, Vienne, Austria.

Pham, D., Tran, N. P., Goh, S. K., Alam, S., & Duong, V. (2019b). Reinforcement Learning for Two-Aircraft Conflict Resolution in the Presence of Uncertainty. Paper presented at the 2019 IEEE-RIVF International Conference on Computing and Communication Technologies (RIVF).

Piera, M. A., Ramos, J. J., Ortiz, R. M., & Radanovic, M. (2018, 6-7 Aug. 2018). A Negotiation Mechanism for a Consensus in Air Traffic Conflict Resolution. Paper presented at the 2018 International Conference on Advances in Big Data, Computing and Data Communication Systems (icABCD).

Prevot, T., Homola, J., & Mercer, J. (2008). Human-in-the-loop evaluation of ground-based automated separation assurance for NEXTGEN. In The 26th Congress of ICAS and 8th AIAA ATIO(p. 8885). Retrieved from http://arc.aiaa.org/doi/pdf/10.2514/6.2008-8885

Rantanen, E. M., & Nunes, A. (2005). Hierarchical Conflict Detection in Air Traffic Control Hierarchical Conflict Detection in Air Traffic Control. The International Journal of Aviation Psychology, 15(4), 339–362. https://doi.org/10.1207/s15327108ijap1504

Rantanen, E. M., & Wickens, C. D. (2012). Conflict Resolution Maneuvers in Air Traffic Control: Investigation of Operational Data. The International Journal of Aviation Psychology, 22(3), 266–281. https://doi.org/10.1080/10508414.2012.691048

Rao, A.S. & M. P. Georgeff (1995).BDI agents: From theory to practice, 1st ICMAS, San Fransisco, CA, Jun. 12-14, 1995.

Regtuit, R. M., Borst, C., Kampen, E.-J. v., & Paassen, M. M. v. (2018). Building Strategic Conformal Automation for Air Traffic Control Using Machine Learning.

Regtuit, R.M., C. Borst, E.-J. van Kampen, and M. M. van Paassen (2018). Building strategic conformal automation for air traffic control using machine learning, AIAA Information Systems-AIAA Infotech at Aerospace, Kissimmee, FL, Jan. 9-12, 2018.

RHEA. (1998). Final report. Role of the human in the evolution of ATM systems (WP8, RHEA/NL/WPR/8/04).

Ribeiro, M.T., S. Singh, C. Guestrin (2016).Why should I trust you?: Explaining the predictions of any classifier, 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, Aug. 2016, pp. 1135–1144.

Ribeiro, V. F., de Almeida Rodrigues, H. T., de Faria, V. B., Li, W., & Garcia, R. C. (2020). Conflict Detection and Resolution with Local Search Algorithms for 4D-Navigation in ATM. In A. Abraham, A. Cherukuri, P. Melin, & N. Gandhi (Eds.), Intelligent Systems Design and Applications. (Vol. ISDA 2018. Advances in Intelligent Systems and Computing): Springer, Cham.

Ribeiro,M., J. Ellerbroek, & J. M. Hoekstra (2020). Review of Conflict Resolution Methods for Manned and Unmanned Aviation. Aerospace, 7(6), 79, 2020.

Ricci, F., Rokach, L., & Shapira, B. (2015). Recommender systems handbook (F. Ricci, L. Rokach, & B. Shapira Eds. 2 ed.). New York, NY: Springer.

Rooijen, S. J. van, Ellerbroek, J., Borst, C., & van Kampen, E. (2019, Jun. 17-21). Conformal automation for air traffic control using convolutional neural networks. Paper presented at the 13th USA/Europe ATM R&D Seminar, Vienna, Austria.

Rooijen, S.J. van, J. Ellerbroek, C. Borst, and E.-J. van Kampen (2020). Toward individual-sensitive automation for Air Traffic Control using convolutional neural networks. J. Air Transp., 28(3), 1-9.

Rummelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. Nature, 323, 533--536. https://doi.org/10.1038/323533a0

Sathya, ZR. & A. Abraham (2013). Comparison of Supervised and Unsupervised Learning Algorithms for Pattern Classification (IJARAI) International Journal of Advanced Research in Artificial Intelligence, Vol. 2, No. 2, 2013

Sathyan, A., Ernest, N., Lavigne, L., Cazaurang, F., Kumar, M., & Cohen, K. (2017). A Genetic Fuzzy Logic Based Approach to Solving the Aircraft Conflict Resolution Problem AIAA Information Systems-AIAA Infotech @ Aerospace: American Institute of Aeronautics and Astronautics.

Seamster, T. L., Redding, R. E., Cannon, J. R., Ryder, J. M., & Janine, A. (1993). Cognitive Task Analysis of Expertise in Air Traffic Control. The International Journal of Aviation Psychology, 3(4), 257–283. https://doi.org/10.1207/s15327108ijap0304

Seifert C. et al. (2017) Visualizations of Deep Neural Networks in Computer Vision: A Survey. In: Cerquitelli T., Quercia D., Pasquale F. (eds) Transparent Data Mining for Big and Small Data. Studies in Big Data, vol 32. Springer, Cham. https://doi.org/10.1007/978-3-319-54024-5_6

Selkowitz, A.R., S. G. Lakhmani, J. Y. C. Chen, and M. Boyce (2015). The effects of agent transparency on human interaction with an autonomous robotic agent, HFES Annual Meeting, 59(1), 2015, pp. 806-810.

SESAR (2019) European ATM Master Plan. Digitalising Europe’s Aviation Infrastructure. Executive View. 2020 Edition. SESAR Joint Undertaking, Luxembourg: Publications Office of the EU, Dec. 17.

Sheridan, T.B. (1992). Telerobotics, Automation, and Human Supervisory Control, Cambridge, MA: MIT Press.

Silver, D., Huang, A., Maddison, C. et al. Mastering the game of Go with deep neural networks and tree search. Nature 529, 484–489 (2016). https://doi.org/10.1038/nature16961

Simon, H.A. (1956). Rational choice and the structure of the environment, Psychol. Rev., 63(2), 1956, pp. 129-138.

Sipe, A., & Moore, J. (2009). Air traffic functions in the NextGen and SESAR airspace. In Digital Avionics Systems Conference, 2009. DASC’09. IEEE/AIAA 28th(pp. 2. A. 6-1-2. A. 6-7). IEEE. Retrieved from http://ieeexplore.ieee.org/abstract/document/5347554/

Sutton, R.S. & A.G. Barto (2017). Reinforcement Learning: An Introduction. The MIT Press, Cambridge, Massachusetts, 2nd edition, 11 2017.

Szalma, J.L. (20098). Individual differences in human–technology interaction: Incorporating variation in human characteristics into human factors and ergonomics research and design, Theor. Issues Ergon. Sci., 10(5), 2009, pp. 381-397.

Theodorou, A., R. H. Wortham, and J. J. Bryson (2017). Designing and implementing transparency for real time inspection of autonomous robots, Connect. Sci., 29(3), 2017, pp. 230-241.

Tony, L. A., Ghose, D., & Chakravarthy, A. (2020). CONCORD: A UAV Conflict Resolution System using Correlated Equilibrium based Decision Making AIAA Scitech 2020 Forum: American Institute of Aeronautics and Astronautics.

Tran, N. P., Pham, D., Goh, S. K., Alam, S., & Duong, V. (2020). An Interactive Conflict Solver for Learning Air Traffic Conflict Resolutions. Journal of Aerospace Information Systems, 17(6), 271-277. doi:10.2514/1.I010807

Trapsiliwati, F., Wickens, C., Chen, C. & Qu, X. (2017). Transparency and conflict resolution automation reliability in air traffic control. In Proceedings of the International Symposium for Aviation Psychology (ISAP). pp 420-424. Dayton, Ohio: Wright State University / ISAP.

Volpe Lovato, A., Hora Fontes, C., Embiruçu, M., & Kalid, R. (2018). A fuzzy modeling approach to optimize control and decision making in conflict management in air traffic control. Computers & Industrial Engineering, 115, 167-189. doi:https://doi.org/10.1016/j.cie.2017.11.008

Vouros, G., J. M. Cordero, P. Costas, and G. Fuchs (2018). Data-driven aircraft trajectory prediction research (DART): Final project results report, D4.5, 2018. Retrieved from: https://www.sesarju.eu/projects/dart [Online].

Wang, N., D. V. Pynadath, and S. G. Hill (2016). Trust calibration within a human-robot team: Comparing automatically generated explanations, 11th ACM/IEEE International Conference on Human Robot Interaction, Christchurch, New Zealand, Mar. 7-10, 2016, pp. 109-116.

Wang, Z., Li, H., Wang, J., & Shen, F. (2019a). Deep reinforcement learning based conflict detection and resolution in air traffic control. IET Intelligent Transport Systems, 13(6), 1041-1047. doi:10.1049/iet-its.2018.5357 Wang, Z.-y., Liang, M., Delahaye, D., & Wu, W. (2019b). Learning Real Trajectory Data to Enhance Conflict Detection Accuracy in Closest Point of Approach. Paper presented at the 13th USA/Europe ATM R&D Seminar, Jun. 17-20, Vienna, Austria.

Wang, Z., Liang, M., & Delahaye, D. (2020). Data-driven Conflict Detection Enhancement in 3D Airspace with Machine Learning. Paper presented at the 2020 International Conference on Artificial Intelligence and Data Analytics for Air Transportation (AIDA-AT), Feb. 3-4.

Wen, H., Li, H., Wang, Z., Hou, X., & He, K. (2019, 14-15 Dec. 2019). Application of DDPG-based Collision Avoidance Algorithm in Air Traffic Control. Paper presented at the 2019 12th International Symposium on Computational Intelligence and Design (ISCID).

Westin, C., Borst, C. & Hilburn, B. (2016a). Strategic conformance: Overcoming acceptance issues of decision aiding automation? IEEE Trans. Human–Mach. Syst., 46(1), 2016, pp. 41-52.

Westin, C., C. Borst, and B. Hilburn (2016b). Automation transparency and personalized decision support: Air traffic controller interaction with a resolution advisory system, In IFAC-Papers Online, Kyoto, Japan, 2016 pp. 201-206.

Woods, D.D. (1996). Decomposing automation: Apparent simplicity, real complexity, in Automation and human performance: Theory and applications, R. Parasuraman and M. Mouloua, Eds., ed Hillsdale, NJ, England: Lawrence Erlbaum Associates, 1996, pp. 3-17.

Xu-rui, J., Ming-gong, W., Xiang-xi, W., & Ze-kun, W. (2018, 26-28 May 2018). Application of ensemble learning algorithm in aircraft probabilistic conflict detection of free flight. Paper presented at the 2018 International Conference on Artificial Intelligence and Big Data (ICAIBD).

Yang, J.; Yin, D.; Niu, Y.; Shen, L. Distributed Cooperative Onboard Planning for the Conflict Resolution of Unmanned Aerial Vehicles. J. Guid. Control. Dyn. 2019, 42, 272–283.

Yang, Y., Cai, K.-q., & Prandini, M. (2017). Fast Algorithm Based on Computational Geometry for Probabilistic Aircraft Conflict Detection. Paper presented at the International Conference on Robotics and Artificial Intelligence, Shanghai, China.

Acronyms

AI	Artificial Intelligence
ANN	Artificial Neural Network
APP	Approach Control
BP	Bilevel Programming
CARD	Conflict and Risk Display
CD	Conflict Detection
CD&R	Conflict Detection and Resolution
CNN	Convolutional Neural Network
CPA	Closest Point of Approach
CR	Conflict Resolution
CWA	Cognitive Work Analysis

D2MAV	Deep Distributed Multi-Agent Variable
DDPG	Deep Deterministic Policy Gradient
DEP	Departure Control
DRL	Deep Reinforcement Learning
EID	Ecological Interface Design
FFNN	Feed-Forward Neural Network
GBM	Gradient Boosting Machines
KNN	K-Nearest Neighbour
LOA	Level/s of Automation
LSTM	Long Short-Term Memory
MAHALO	Modernising ATM via Human-Automation Learning Optimisation
MDP	Markov Decision Process
MINLP	Mixed-Integer Nonlinear Programming
MIQCP	Mixed-Integer Quadratic Constrained Programming
MIQP	Mixed-Integer Quadratic Programming
ML	Machine Learning
MLR	Multiple Linear Regression
MSAW	Minimum Safe Altitude Warning
MTCD	Medium-Term Conflict Detection
MUFASA	Multidimensional Framework for Advanced SESAR Automation
NN	Neural Network
POMDP	Partially Observable Markov Decision Process
POSH	Parallel-rooted, Ordered, Slip-stack Hierarchical
PPO	Proximal Policy Optimization
RF	Random Forests

RL	Reinforcement Learning
SA	Situation Awareness (or Simulated Annealing)
SAT	Situation Awareness-based Agent Transparency
SL	Supervised Learning
SOAR	State of the Art Report
STCA	Short-Term Conflict Alert
SVM	Support Vector Machine
TCT	Tactical Controller Tool
TMA	Terminal Maneuvering Area
UI	User Interface
UL/USL	Unsupervised Learning