Human-interpretable input for Machine Learning in Tactical Air Traffic Control

Tiago Miguel Monteiro Nunes, Clark Borst and Erik-Jan van Kampen
Control & Simulation, Faculty of Aerospace Engineering
TU Delft, 2629 HS, Delft, Netherlands
contact: t.m.monteironunes@tudelft.nl

Brian Hillburn
Center for Human Performance Research
Oosteinde 263, 2272 AE Voorburg, NL
Contact: brian@chpr.nl

Carl Westin
Department of Science and Technology
Linköping University SE-601 74 Norrköping, Sweden
carl.westin@liu.se

***Abstract*—Increasing airspace demand requires an increase in effectiveness and efficiency of the ATC system. Automation, and specifically Machine Learning (ML), may present good prospects for increasing system performance and decreasing workload of ATCOs. AI, however, is typically a "black box" making it hard to include in a socio-technical environment. This exploratory research aims to increase operator trust and acceptance and move towards a more "cooperative" approach to automation in ATC. It focuses on building upon previous efforts by using two different approaches: Strategically Conformal AI and Explainable AI methods to AI-Human interactions. Strategic Conformance aims to increase acceptance by producing individual-sensitive advisories. Explainable AI focuses on producing more optimal solutions and providing a clear explanation for these solutions. In this article, we propose the use of a single visual representation for tactical conflict detection and resolution, called the Solution Space Diagram (SSD), to serve as a common ground for both explainable and conformal AI. Through this research, it has become clear that there needs to be a careful definition given both to optimality and conformance. Likewise, the training of the AI agents comes with requirements for a large amount of data to be available and displaying these solutions in a human-interpretable way, while maintaining optimality, has its own unique challenges to overcome.**

***Keywords*—Machine Learning; Human Machine Interaction; Decision Support Systems**

I. Introduction

All conflict resolution ATC tasks are currently performed manually, by Air Traffic Controllers (ATCOs), and thus are limited by the controller's performance and capabilities. An ATCO can only handle a limited number of aircraft simultaneously, making this the main bottleneck of the system under increased traffic densities and more stringent performance requirements. Even though the current SARS-COVID2 pandemic has led to a decrease in overall air traffic, this demand is expected to reach (and even surpass) pre-pandemic levels shortly.

One promising way to handle increased traffic densities and increased performance requirements is to increase the level of automation. The capabilities of advanced automated systems, such as Artificial Intelligence (AI) based agents, are promising [1] [2] [3] [4] [5] [6]. Therefore, it would seem that the key to success is to either fully automate the system or provide automated solutions to the ATCOs as advisories.

The first option is currently not being considered due to several factors such as liability and the automation not being reliable enough to predict and act upon every conceivable (or unexpected) scenario and conflict. Following this rationale, SESAR's ATM master plan suggests to a move towards supervised control, where humans play a central role in overseeing the automation's performance and intervene whenever desired or required.

Supervisory control, however, may also come with its own problems, as articulated by Bainbridge's pivotal paper called the "*Ironies of Automation*". Additionally, previous research has shown that ATCOs' distrust of automation risks it being disused. Thus having effective automation for solution advisories will be useless if it is underused by the people it is meant to support. Research indicates that this lack of trust comes from the ATCOs' limited understanding of the automation, fears concerning job security and the limited experience with AI. Since ATCOs are still the ones ultimately responsible for the decisions made in the ATC domain, they tend not to use solutions given by automation they cannot understand or properly monitor. This issue is compounded by the fact that AI agents tend to be opaque in terms of their function, being often referred to as "black boxes."

To increase operator trust and acceptance and move towards a more "cooperative" approach to automation in ATC, we suggest to adopt two different approaches: Strategically Conformal AI and Explainable AI methods. Strategically Conformal AI builds upon previous research by Westin et al. [7]. In that research, it was shown that having solutions presented to the ATCO that match personal preferences tends to increase acceptance. Therefore, Machine Learning (ML) can be used to learn how to replicate ATCO strategic preferences and give strategically conformal solution advisories. The underlying ML method to achieve strategic conformal AI has been initiated in the research by Van Rooijen et al. [8]. The main disadvantage of 100% conformal automation would be that mimicking an ATCO's control actions might include sub-optimal actions.

To increase acceptance of AI models that generate more optimal control actions, giving the ATCo insights into the AI model might be helpful. Here, Explainable AI (XAI) methods would allow for explanations to be given to the ATCo. Ideally, XAI can allow for an AI agent to focus solely on optimality, while providing sufficient context for its solution such that the ATCO is able to understand and monitor the decision-making process of the automation.

In this article, we introduce a single visual representation to serve as inputs to achieve both conformal and optimal AI-based automation. The visual representation is the so-called Solution Space Diagram (SSD), which has originally been designed as a tactical decision-support tool for ATCos in performing (manual and supervisory) conflict detection and resolution (CD&R). The SSD is an interface that is based on Velocity Obstacles (VOs) and displays the free space maneuver space in terms of heading and speed of the aircraft [9], a graphical abstraction of it can be seen in Figure 1.

Being a successful tool in supporting ATCo decision-making, we hypothesize the SSD to be equally useful as a way to communicate the system state (including conflicts and potential solutions) to an artificial agent. Additionally, given that the SSD has been designed as a decision-support tool, the pixel input to the machine learning agents will be interpretable, thus contributing to XAI.

Within this research, the Conformal AI will be developed under a Supervised Learning (SL) framework whereas the XAI will be developed under a Reinforcement Learning (RL) framework. The reasons behind this choice will be expanded upon in section II. A Hybrid agent that is made up of both conformal and optimal considerations will also be presented and is currently a topic being explored.

Section II contains information on what automation is used and what kind of inputs it receives. The overall system setup is described in section III. Section IV contains an overview of the experiments to be conducted and the hypotheses to be tested in them. In section IV-B the results achieved so far will be presented and in section V a discussion on the results and the limitations of the approach is provided. This article ends with a series of conclusions in section VI.

II. Conformal and Optimal AI Agents

Within this research, two different AI agents will be used. For the Conformal automation, a SL agent based on Convolutional Neural Networks (CNN) will be used, since SL is particularly suited to learning to replicate rather than derive solutions. For the "Optimal" automation a RL agent will be used. RL agents are capable of learning how to improve their performance through interaction with the environment. One appealing feature of RL, specifically for this research, is that it works primarily through designing a reward function that shapes the RL agent's behaviour. Shaping this reward function is, however, not always trivial and may require careful considerations.

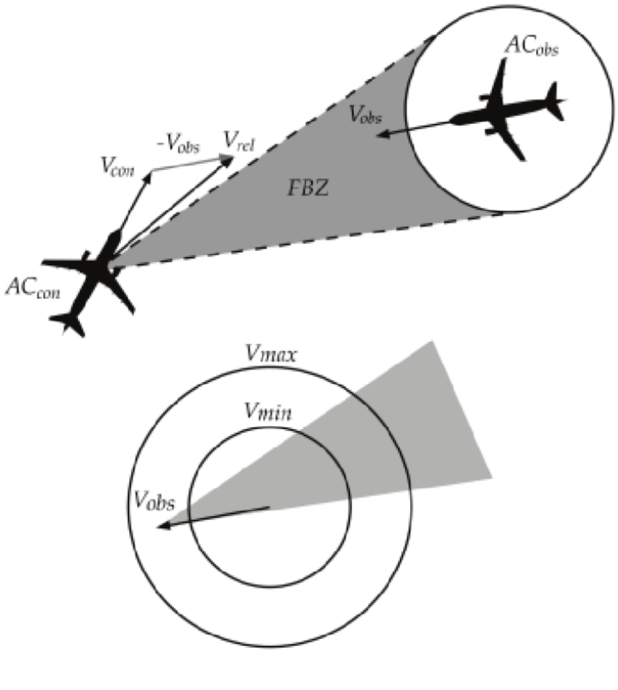

Figure 1. Graphical abstraction of an SSD, from [10]

A. SSD: Definition

Originally, The SSD was based on the principles underlying the Ecological Interface Design (EID) framework [11] [12] [13]. Therefore, it aims to provide information about the deep structure of physical constraints (and their relationships) of the traffic system to the ATCO. Results of previous experiments [7] suggest that the SSD can increase situation awareness and performance of an ATCO while performing both manual and supervisory CD&R tasks.

The SSD construction is as illustrated in Figure 2 where two aircraft, AC_{con} and AC_{obs} can be seen, and shows the solution space of the controlled aircraft. This means that it not only provides information on the current conflict, but also on the constraints an ATCO should respect when giving a resolution command, such as the speed envelope and the desired direction to the exit waypoint. This information allows the ATCO to consider both the efficiency of the command when it comes to flight path deviation and also the robustness and safety of that command. It has also been shown, in previous research, to work adequately as an input to ML agents [8], [14].

B. Supervised Learning

The system used for the conformal agent is based upon the work of [8]. This system provides the ATCO with personalized solutions for conflicting aircraft. As mentioned above, the artificial agent receives a cropped SSD as input. This SSD is sized as a 128x64 pixel image. The reason for this is that, in

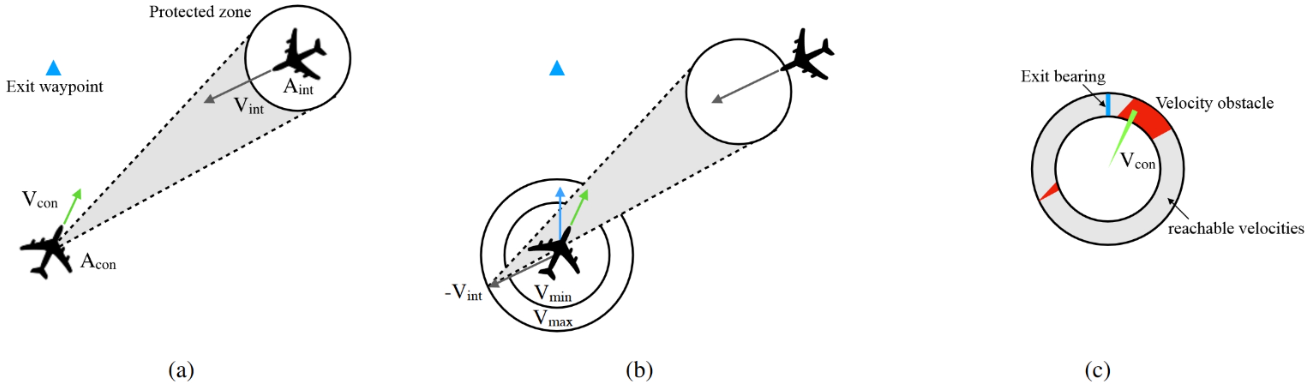

Figure 2. Construction of the SSD: a) The set of conflicting velocity vectors is shown by the grey area, b) this area is displaced by the velocity vector of the intruding aircraft, c) The minimum and maximum velocity of the controlled aircraft limit the solution area creating the SSD. From [8]

TABLE I
SPECIFICATION OF NN LAYERS FOR THE SL MODEL

Convolutional	Filter size: 2x2	Stride: 1x1	ReLU
Max-pooling	Size: 2x2	Stride: 2x2	ReLU
First dense layer	Dropout rate: 20%	-	ReLU
Second dense layer	-	-	Softmax

experiments, it was shown that having the full SSD versus only the top half does not further increase performance and it also makes training more cumbersome. The SSD image is rotated such that the current heading always upwards. An example of this cropped SSD can be seen in Figure 3. The system's output is a solution advisory to the ATCO in the form of a heading deviation from the current heading.

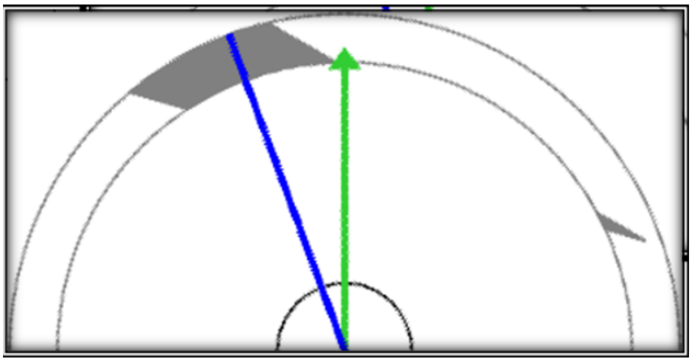

Figure 3. Example of the cropped SSD, showing the speed envelope, the conflict zone, the exit direction (blue) and the current speed vector (green). From [14]

The CNN model is made up of a series of three convolutional layers interspersed with max pooling-layers, two dense layers were then added at the end of the last max-pooling layer. The output of the model has a size of 3. The parameters of the hidden layers can be seen in table I

A hyper-parameter tuning process followed. The resulting hyper parameters can be found in table II.

TABLE II
PARAMETERS OF THE SL MODEL

Parameters	Value
Optimization Algorithm	Adam
Loss function	Cross-entropy
Train/validate ratio	80% / 20%
Batch size	4
Epochs	6
Learning rate	0.01
Dropout rate	20%
Input shape	128x64 pixels

C. *Reinforcement Learning*

For the Optimal part of the system, a RL approach is used. Namely, two different agents have been developed. The first one, based on the Q-learning architecture, works as a "Supervisor." It handles the optimization of the parameters of a deterministic conflict resolution algorithm. The reasoning behind this is that Q-Learning is a simple framework for RL, using a well-performing deterministic algorithm with this RL agent as a supervisor to improve performance provides a good middle ground between full RL and full deterministic control. The second agent is based on Deep Q-Learning from Demonstrations (DQfD). This agent was selected because, in preliminary research, it was shown to work well in this environment [14] and with the SSD as input. Another advantage of DQfD is that it can make use of a pre-training phase, where it is exposed and trains on "expert demonstrations." This allows the agent to learn a good initial policy for the training, thus speeding up training and requiring less interaction with the environment itself before converging to a good policy.

As with the SL system, the DQfD agent also receives the SSD as input. Likewise, it also outputs a solution advisory in the form of a heading change. A description of DQfD can be found in [14]. The high-level idea of its use in this research is to use it in conjunction with demonstrations from a deterministic conflict resolution algorithm to pre-train it without need for much direct interaction with the ATM simulator. This makes

TABLE III
PARAMETERS OF THE SL MODEL

Parameters	Value
Optimization Algorithm	Adam
Learning rate (α)	0.00001
Batch size (n)	128
ϵ	1.0 to 0.05 (decaying)
Loss-function	$L(Q) = L_{DQN}(Q) + \lambda_1 L_{(n-step)}(Q) + \lambda_2 L_E(Q) + \lambda_3 L_{L2}(Q)$
Input size	128 X 64 X 1 (after pre-processing of SSD)
Output	6 possible heading actions

generating data easier. For a full description of Q-learning, the reader is referred to [15]. Both Q-learning and DQfD are tried and tested methods as well as being efficient and successful. The RL model relies on a set of 10 layers. The architecture of the network is a Dueling-DQN network. This architecture is capable of achieving a better performance than a regular DQN network in most tasks as shown in [16]. For this research, a dueling network with two heads, one that estimates the state value and one that estimates the action advantage, is used. These two heads share the same feature learning module. This feature learning module is made up of the CNNs as described originally in [17]. The composition of the value and advantage streams are the same and correspond to a fully-connected layer with 512 neurons. Due to the different pre-processing operations performed on the image, the overall output of the feature learning module has a size of 3072. This is, thus, the size of input of both streams.

As with the SL model, a hyper-parameter tuning process is followed. The results of which are described in table III

D. Hybrid Automation

Within this research, a hybrid automation agent is also considered. This agent takes information from both the optimal and conformal models and creates, according to a given set of parameters, a "middle-ground" between conformance and optimality. The rationale behind this system is that while conformance is desirable to increase acceptance, it is also desirable to have a higher level of optimality. Having a piece of automation that can make better decisions than the human operator, but still remain strategically conformal, would, in our view, be the best possible approach.

E. Limitations of using an SSD input

There are limitations to using an SSD as input to a ML agent. Namely, the SSD is limited in the information it is able to display. This is readily apparent when one looks at the original design of the SSD. It lacked information on the target heading and the time to loss of separation. Additionally, it is only useful for heading and/or speed solutions, thus excluding altitude. These limitations can be somewhat mitigated through a careful re-design. The target heading can be added in a pre-processing stage to the SSD as another colored line. The different parts of the SSD conflict zones can be given different colors that allow the agent to detect different times to loss-of-separation. Including altitude information is, however, more difficult to embed in the 2D velocity obstacle representation and may require other design considerations to be explored in future research.

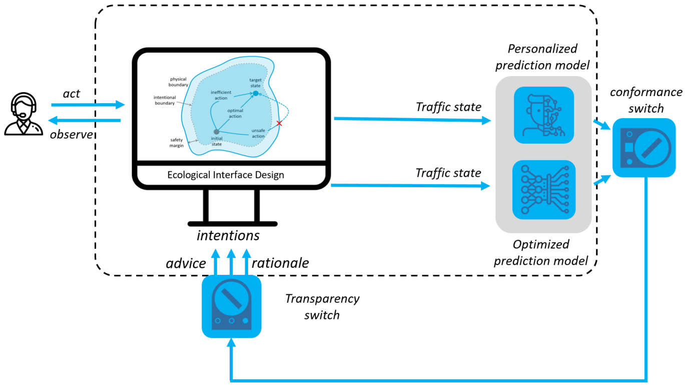

Figure 4. Conceptual diagram of the different components and their interactions.

One other limitation of the SSD is that the it is only a "1-step-ahead" interface, i.e. it only displays the action that avoids the conflict and does not consider what and when the correct action should be to "steer-back" to the target heading. This can, again, be somewhat mitigated through several methods such as giving the RL agent penalties for a longer flight path, thus implicitly rewarding it for finding better ways to perform the avoidance and steer-back actions. It is, nonetheless, a limitation of the interface itself. Future research will look at using the s-called Travel Space Representation (TSR) [18] as a visual input as it is a sort of "integrated SSD" and includes both the avoidance action as well as the "steer-back" action. It does this, specifically, by presenting the solution to the conflict while already considering a "steer-back" command for the aircraft to resume its normal flight after the conflict is solved.

III. SYSTEM SETUP FOR VALIDATIONS

The envisioned interaction between the different types of automation, the ATM simulator, and the human operator (ATCo) can be seen in Figure 4. Validation experiments are yet to be conducted under the umbrella of the MAHALO SESAR exploratory research project. The "switches" presented in the figure above will be static as part of the experimental design as opposed to real-time shifts. For each Human-in-the-Loop (HITL) experiment, the positions of these switches needs to carefully calibrated. The "transparency" switch controls exactly how much of an explanation (if any) is given to the operator (e.g., by showing/hiding parts of the SSD). The "conformance" switch influences how individually-sensitive the output of the system will be.

A. Data Generation

Data will be generated based on the conditions described above. The BlueSky ATM simulator [19] is used to generate data and train the RL agent as it is capable of generating large numbers of automated conflict resolution example using Modified Voltage Potential (MVP). The ATM simulator

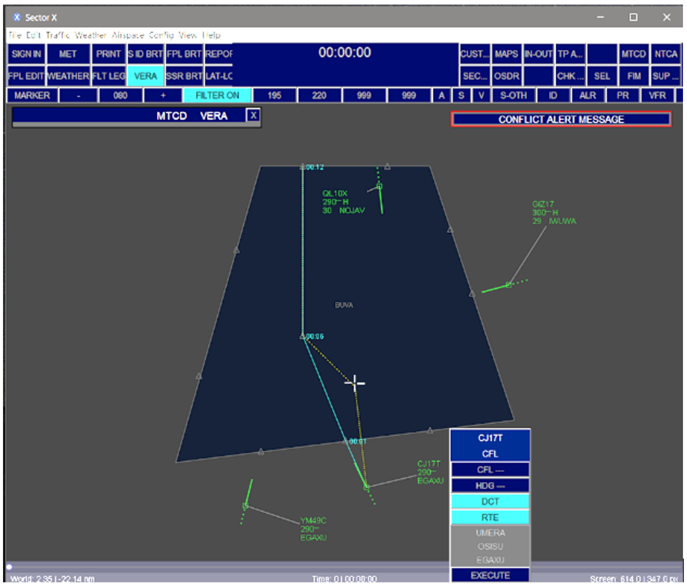

Figure 5. State of the art PVD implemented in SectorX

SectorX (see Figure 5) , a Java-based medium fidelity ATC simulator for HITL research, is used to generate data for the SL agent through a simplified heading solver implemented in the code. The reason for using two different simulators is that BlueSky is implemented in Python. Since the RL agent requires interaction with the system in order to learn better, it is helpful to have a simulator that is implemented in Python as this is the de-facto ML programming language.

B. System Interactions

Since two different simulators are used, several software interfaces need to be developed to translate scenarios from BlueSky to SectorX and vice versa. Whenever a solution from the RL agent is required in the experiment, for example, the scenario is converted into BlueSky, the RL agent's advisory is calculated and is converted back into SectorX. While these interfaces are important for the proper functioning of the overall system, they will not be described in depth as they are only background applications.

IV. Experiments

A. Overview

In order to validate the assumptions made and the ML agents created, experiments will be performed in the future. These experiments are carefully designed in order to effectively study the effects of conformance and explainability. Namely, three experiments are to be conducted during this research.

Simulation 1 will take place at TU Delft and will serve to validate and test the implementation of the techniques described. This experiment will involve novices (university students) in generating data and testing the interfaces and data collection protocols.

Simulations 2a and 2b will be performed in Italy and Sweden, respectively, with professional ATCos as participants. These experiments are each split into two stages: a Conformance Pre-test and a Final experiment. During the Conformance Pre-test, data will be collected on the way ATCOs solve conflicts such that the conformal automation can be properly trained. The Final experiment will happen some weeks after the Conformance Pre-test and will involve exposing ATCOs to conformal advisories (trained from the pre-test data) and optimal advisories (from the RL model) in a supervisory control setting. During these experiments several hypotheses will be tested:

- Controller self-reported acceptance of advisories is higher for solutions that conform to the controller's chosen solution. Likewise, it will be higher in the presence of a high transparency display format.
- Assuming that they differ, conformal solutions are expected to provide a higher level of acceptance/agreement than optimized solutions.
- Optimal solutions will be associated with lower reported trust than will conformal solutions.
- Increases in either conformance and/or transparency will be associated with a decreased controller reported workload and acceptance.

Two independent variables will be studied: Transparency and Conformance. Conformance will be provided in three levels: Personalized prediction model, Group prediction model and Non-conformal model.

Likewise, Transparency will be provided in three levels: No Transparency, Domain Transparency (Highlighting of the relevant part of the solution space) and Agent Transparency (agent provides a text-based explanation for choosing one solution over another, for example, by explaining that solution A leads to a lower flight path deviation than solution B). Since all solutions can be generated before the experiments are actually ran, developing ML models capable of generating advisories in real-time is not a concern for this research. This removes some constraints for choosing the agents, but also for performing the experiment since there is not a specific need for high-end hardware.

B. Expected Results

The main results obtained thus far relate to the development of the different ML agents, their interfaces and the presentation within Sector X. The ML agents have been tested for their reliability and accuracy. As explained above, run time is not a concern for this research. Previous research as well as preliminary results gathered throughout the development of these systems demonstrate that the approach chosen is adequate. The main results will be obtained after the Main experiment when trust and acceptance will be measured and the hypotheses can be tested.

V. Discussion

While the progress seen in this research is promising, we acknowledge several limitations. These limitations have to do with the traffic scenarios being considered, the actions the agents were allowed to take and the input given to the ML agents.

As stated above, the traffic scenarios considered were limited. Future research should look into more complex forms of traffic involving en-route as well as terminal traffic and weather effects. The limited set of allowed actions (only horizontal resolutions in terms of heading and/or speed) also contributes to a reduced level of realism. For a better idea of the effect a system like this would have on real air traffic control, the set of actions the automation can take has to be expanded. In a first step, automation could be allowed to also issue altitude commands (which are the most common resolution commands in en-route airspace). These limitations were mainly due to a need to reduce traffic complexity to test the effects of conformance and explainability alone.

The input given to the ML agents was chosen to be the SSD as it is a simple, yet effective, interface that has been proven to work well as a decision-support tool. Ideally, an interface that provides the ML agent with more information would be ideal. One of the main limitations of the SSD is that it does not include information on steering back to the target waypoint. Another limitation is that it only displays information on the horizontal domain. This research thus provides a first step in showing how to possibly achieve conformal and optimal AI using on a visual, human-interpretable state representation as inputs to ML models. Nevertheless, it is essential that either the SSD is adapted to contain more information or that another interface is used if this approach is to be followed in the future.

VI. Conclusions

This article presented an overview of the current approach being taken by the MAHALO team to develop an Artificial Intelligence agent that is also capable of communicating with a human ATCO using a visual representation of the traffic state, called the Solution Space Diagram. The main focus of this research is to explore how conformal and optimal AI models can be made transparent in order to foster acceptance of such models. Throughout both a literature review and several workshops, it has been seen that AI, and in particular ML, is a topic that is currently generating much interest in the ATC community.

While it is too early to provide conclusions about our suggested approach, the current stage of research offers optimistic results in terms technical feasibility. The next steps in the development will be the upcoming experiments, where the impact of manipulating levels of transparency and conformance on operator acceptance and human-automation interaction will be explored.

References

[1] A. N. Aneesh, L. Shine, R. Pradeep, and V. Sajith, “Real-time traffic light detection and recognition based on deep retinanet for self driving cars,” in *2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT)*, vol. 1, 2019, pp. 1554–1557.

[2] G. Bohouta and V. Këpuska, “Next-generation of virtual personal assistants (microsoft cortana, apple siri, amazon alexa and google home),” in *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*, 01 2018.

[3] X.-Y. Zhang, C.-L. Liu, and C. Y. Suen, “Towards robust pattern recognition: A review,” *Proceedings of the IEEE*, vol. 108, no. 6, pp. 894–922, 2020.

[4] I. Oh, S. Rho *et al.*, “Creating pro-level ai for a real-time fighting game using deep reinforcement learning,” 2020.

[5] Y.-L. Shen, C.-Y. Huang *et al.*, “Reinforcement learning based speech enhancement for robust speech recognition,” in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 6750–6754.

[6] H. Wen, H. Li *et al.*, “Application of ddpg-based collision avoidance algorithm in air traffic control,” in *2019 12th International Symposium on Computational Intelligence and Design (ISCID)*, vol. 1, 2019, pp. 130–133.

[7] C. Westin, B. Hilburn, and C. Borst, “The effect of strategic conformance on acceptance of automated advice: Concluding the mufasa project,” *SIDs 2013 - Proceedings of the SESAR Innovation Days*, 01 2013.

[8] S. Rooijen, J. Ellerbroek, C. Borst, and E.-J. Van Kampen, “Conformal automation for air traffic control using convolutional neural networks,” in *Proceedings of the 13th USA/Europe Air Traffic Management Research and Development Seminar 2019, ATM 2019*, 06 2019.

[9] J. d’Engelbronner, C. Borst *et al.*, “Solution-space-based analysis of dynamic air traffic controller workload,” *Journal of Aircraft*, 04 2015.

[10] S. M. Abdul Rahman, C. Borst, and others., “Solution space diagram in conflict detection scenarios,” *Jurnal Teknologi*, vol. 75, pp. 53–58, 08 2015.

[11] M. M. Van Paassen, C. Borst *et al.*, “Ecological interface design for vehicle locomotion control,” *IEEE Transactions on Human-Machine Systems*, vol. 48.5, pp. 541–555, 08 2018.

[12] K. Vicente and J. Rasmussen, “Ecological interface design: theoretical foundations,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 22, no. 4, pp. 589–606, 1992.

[13] S. Dam, M. Mulder, and M. M. Van Paassen, “Ecological interface design of a tactical airborne separation assistance tool,” *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 38, pp. 1221 – 1233, 12 2008.

[14] M. Hermans, “Towards explainable automation for air traffic control using deep q-learning from demonstrations and reward decomposition,” MSc Thesis, TU Delft, 2021.

[15] T. Hester, M. Vecerik, O. Pietquin, M. Lanctot, T. Schaul, B. Piot, D. Horgan, J. Quan, A. Sendonaris, I. Osband, G. Dulac-Arnold, J. Agapiou, J. Leibo, and A. Gruslys, “Deep q-learning from demonstrations,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, Apr. 2018. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/11757

[16] Z. Wang, T. Schaul *et al.*, “Dueling network architectures for deep reinforcement learning,” 2016.

[17] V. Mnih, K. Kavukcuoglu *et al.*, “Human-level control through deep reinforcement learning,” *Nature*, vol. 518, pp. 529–533, 2015.

[18] R. Klomp, C. Borst, M. Mulder, and G. Praetorius, “Ecological interface design: Control space robustness in future trajectory-based air traffic control decision support,” vol. 2014, 10 2014, pp. 329–334.

[19] J. Hoekstra and J. Ellerbroek, “Bluesky atc simulator project: an open data and open source approach,” 06 2016.