AIAA SciTech Forum
8–12 January 2018, Kissimmee, Florida
2018 AIAA Information Systems-AIAA Infotech @ Aerospace

10.2514/6.2018-0074

Building Strategic Conformal Automation for Air Traffic Control Using Machine Learning

R.M. Regtuit* C. Borst† E. van Kampen‡

M.M. van Paassen§

Delft University of Technology, 2629HS Delft, The Netherlands

Acceptance of automation has been a bottleneck for successful introduction of automation in Air Traffic Control. Strategic conformal automation has been proven to increase automation acceptance, by creating a better match between automation and operator decision-making. In this paper strategic conformal automation for Air Traffic Control is designed using machine learning techniques. Rather than having pre-defined control strategies, which do not always match with individual operator decision-making, the automation is based on the operator's decision-making. Results show that when operators demonstrate their control strategies, machine learning techniques can identify these strategies and use them to learn similar control strategies. Apart from mimicking control strategies in identical traffic scenarios is it possible to use machine learning to solve similar, yet different conflicts by applying similar control strategies, without the need of human demonstrations for that particular conflict scenario. Future research should be done to investigate whether strategic conformal automation indeed increases automation acceptance, as well to investigate how the approach taken in this study can be applied to real-life traffic scenarios.

I. Introduction

Introducing automation in Air Traffic Control is not a straightforward task and it is foreseen that operator *acceptance* is one the largest obstacles.[1] An explanation why operators are reluctant to accept automation is that they do not always understand *why* certain solutions are proposed by the automation tool. One of the reasons is a mismatch in problem-solving style between automation and operator. It is therefore recommended to base automation strategies on human heuristics.[2]

This idea is supported by the view of Westin et al.[3] They suggest to design automation according to the principle of strategic conformance, a concept they define as *"the match in problem-solving style between decision aiding automation and the individual operator"*. This is an extension to the solution presented by Billings,[2] where not only differences between machine and human decision-making should be taken into account, but even more extreme, the differences between individual decision-making as well.

The idea behind strategic conformal automation is that the automation control strategies are similar to the individual operator's strategies. This eliminates the issue of not understanding *why* automation makes certain decisions, and can potentially increase the initial acceptance of automation. It is initial acceptance that is important in the first place, since trust in a system can only be developed by actually using it.[3]

In a study conducted by Hilburn et al.[1] the hypothesized benefits of strategic conformal automation have been investigated. Rather than developing a strategic conformal automation tool, they disguised operator's control strategies as automation strategies. They showed their participants two types of strategies: strategies that were their own (conformal) and strategies from their colleagues (non-conformal). Results indicated that conformal solutions were accepted more often and agreed with quicker than non-conformal solutions. This seems to confirm the hypothesis that strategic-conformal automation is better accepted than automation that is not.

*MSc student, Control and Simulation, Delft University of Technology.
†Assistant Professor, Control and Simulation, Delft University of Technology, AIAA member.
‡Assistant Professor, Control and Simulation, Delft University of Technology, AIAA member.
§Associate Professor, Control and Simulation, Delft University of Technology, AIAA member.

Where the work by Hilburn et al.[1] masked replays as "automation", this study will focus on the actual development of strategic conformal automation. To achieve this, data clustering is applied to identify human control strategies, whereas a special branch of machine learning will be used to replicate human control strategies: Reinforcement Learning. The replicated control strategies can be used to provide the ATCo with strategic conformal advisories, potentially leading to higher acceptance.

The paper is structured as follows: Section II discusses the task performed by Air Traffic Controllers (ATCos) and how they perform this, as well a motivation to take a machine learning approach to develop the strategic conformal automation. Section III discusses what steps are needed to develop strategic conformal automation, as well as the approach to identify and replicate human control strategies. Section IV will state the test case that is evaluated in this study and Section V shows the results of this test case. Section VI will discuss the sensitivity (robustness) of the approach used and a discussion of the results is given in Section VII. The study will be concluded in Section VIII.

II. Theoretical Motivation

This section provides theoretical motivation for the approach taken to design strategic conformal automation. In the first subsection the controller task is analyzed, to answer the question *what* task controllers fulfill and *how* they do this. We focus on Conflict Detection and Resolution (CD&R), the most important task of Air Traffic Controllers (ATCos).[4] The second subsection provides a motivation for taking a machine learning approach.

A. Conflict Detection & Resolution

During CD&R, ATCos are responsible for expediting air traffic as efficient as possible, without violating separation minimums. In doing so, four criteria are taken into account.[5] Sorted from most to least important, they are:

1. Violation of minimum separation standards
2. Deviations from standard operating procedures
3. Disorder that may result in cognitive work overload
4. Minimize the number of requests to the pilot

Conflict Detection

In light of the development of TCAS (Traffic Collision Avoidance System), many authors have studied how humans determine whether or not separation minimums are violated. When looking at a potential conflict on the radar screen, ATCos try to estimate what will be the Closest Point of Approach (CPA) between the aircraft. If the estimated CPA is smaller than the separation minimum, the ATCo will classify the aircraft to be in conflict. A literature study has been conducted from which several factors have been identified that influence the CPA estimation capabilities of the ATCos. It was found that when t_{CPA} increases, the ATCos lose accuracy in detecting conflicts. The same relation has been found for the conflict angle. Furthermore it has been found that when aircraft have unequal speeds, the ATCos have more trouble to estimate the CPA. The same is found for the spatial separation the aircraft.

Imagine two aircraft pairs with equal t_{CPA} but with a different speed. This leads to a different spatial separation between the two. The ATCo classifies the closer distanced as more urgent and will react different to it than the larger distanced conflict, despite the equal t_{CPA}. This is known as a "distance-over-speed" bias. Final parameter that has an influence on the conflict detection performance of the ATCos is their current workload. Under high workload conditions, the ATCo is more likely to classify an aircraft pair in conflict than under low workload conditions. Under low workload the ATCOs will apply a "wait-and-see" strategy if they are not sure yet whether an aircraft pair will be in conflict, but under high workload the ATCos are more proactive and will consider it a conflict and resolute the pair. Table 1 gives a summary of the factors that have an influence on the conflict detection task.

Table 1. Factors Influencing ATCo Conflict Detection Performance

States	**Motivation**	**Source**
Closest Point of Approach (CPA)	Operators try to estimate the CPA, to see whether a loss of separation is going to happen and thus whether they have to interfere or not.	
t_{CPA}	In a conflict detection task, ATCos predict up to 10 minutes ahead whether a conflict will occur or not. Succesful conflict detection will decrease when time until the conflict increases.	6 7 8
Conflict angle	When two aircraft are in potential conflict, increasing conflict angles will cause increasing troubles for ATCos to successfully determine whether or not a conflict will occur.	8 9 10
Speed difference	When two aircraft are in potential conflict, a speed difference will cause more troubles for ATCos to successfully determine whether or not a conflict will occur.	11
Distance on display	The distance on the radar screen is needed in order to take into account the ”distance-over-speed bias”	7
Workload	Workload has a negative effect on the ATCo's ability to successfully predict a conflict. Increasing workload will lead to lower conflict detection abilities.	8 12

Conflict Resolution

Studies on how ATCos resolve conflicts (i.e., conflict resolution) are more ambiguous than the ones investigating conflict detection. It is found that operators rely on their internal 'library' to find the best resolution for the given situation.[13] The library is constructed based on experience. Operators develop control strategies based on training and experience and find it hard to explain why they use certain strategies.

Research by Kirwan and Flynn[6] shows that in certain conflict geometries, operators have certain best practices (i.e., rules of thumb), which are shown in Table 2. The most important finding by Kirwan and Flynn is the fact ATCos resolve conflicts using a pair-wise approach. Rather than considering the whole traffic scenario as the problem to solve, ATCos look at individual aircraft pairs to come up with an appropriate resolution for that particular pair in conflict. After finding a suitable resolution, they check the impact on the total traffic in the sector.

In resolving conflicts, ATCos have three types of action they can apply. Sorted from most preferred to lesser preferred,[14] they are:

1. Altitude changes
2. Heading changes
3. Speed changes

Altitude changes are the preferred resolution mechanism, since they require the least amount of monitoring effort.[14] Speed changes are the least preferred, since they lack effectiveness due to the small speed envelope of commercial aircraft when flying at high altitudes.[6]

The results from the CD&R analysis give a good indication how to design strategic conformal automation for ATC. The criteria on which decisions are made have been successfully identified (see Table 1). However, how these parameters are used to make decisions is not clear from the literature. Since decisions are made from internal, mental 'libraries', it can be assumed that these are different for each controller. It is therefore required to identify for each controller separately what decisions they make, based on the found parameters.

Table 2. Best practice resolutions used by ATCos

Resolution	Source
Minimize the number of aircraft to move.	6
Look for one key action that will resolve the situation.	6
Minimize additional track miles flown.	6
At cruising altitude the aircraft's speed envelop is small, therefore the speed cannot change much. Better do not use speed resolutions.	6
Use a pair-wise approach in resolving (potential) conflicts and check the impact of the resolution on other traffic afterwards.	6
In a crossing conflict, turn the slower aircraft behind the faster.	6
Solve the head-on conflict first	6
In a same track conflict, turn faster aircraft direct to route, so it will leave the sector before slower aircraft on same route.	6
Better to put aircraft behind than trying to go in between two aircraft.	6
Under high workload interfere earlier	13
Use of categorical resolutions.	6
5 nm separation probably not sufficient, need more to be safe.	6
Safety first	6
Stabilize until after crossing points	6

B. Machine Learning Approach

ATCos are sometimes inconsistent in their decision-making. Research has shown that in roughly 25% of the cases a different strategy is applied than the nominal one.[1] However, this claim is true for the "average" ATCo and it might be that some operators are more inconsistent than others. Regardless of these possible inconsistencies, a main strategy has to be identified for every operator. Since it is not known beforehand how inconsistent an operator is, machine learning techniques will be used to identify an operator's main strategy.

If it is identified how an ATCo resolves a certain type of conflict, this information can be used to create a strategic conformal strategy for that conflict. But if the conflict is altered a little, the same strategy might not resolve the conflict any more. One option is to analyze the strategy of the ATCo in this new scenario as well, but this would require many human demonstrations. For this reason reinforcement learning (RL) is used to replicate the demonstrated control strategies.

RL is a framework to *learn* control strategies based on received rewards from the environment, similar to how humans learn to complete a task.[15,16] The expected benefit of using RL is that in similar, yet different conflict situations, a similar control strategy can be found, without the need of an additional demonstration at that particular conflict geometry.

Another reason for using RL for creating strategic conformal automation is that some RL algorithms are model-free. Since only the decision parameters are known and not the model of what strategy to apply given the parameters, the problem is best solved model-free. Operators find it sometimes hard to explain why they came-up with a certain strategy given the traffic scenario, since they mainly rely on their "internal" library. They have learned via experience and training what to do under different traffic scenarios. By using the identified operator strategy as an input to the RL agent, a policy is learned that is similar to the ATCo's decision-making. The advantage of this method is that operators do not have to explain what control strategy they would use, but they rather demonstrate it. This takes away the problem that ATCos cannot always explain why they apply a particular strategy. From these demonstration the RL agent develops a policy which can be seen as the "internal" library of the automation. Furthermore has RL a history of using human demonstrations to learn control strategies (e.g.,[17]).

III. Methodology

This section will elaborate on the methodology used to create strategic conformal automation for ATCo's. First the general approach is discussed, followed by a more in-depth explanation of the intermediate steps taken.

A. Conceptual Design

To make automation conformal to the ATCo's decision-making, it is crucial to identify what strategies the controller uses. Since literature does not provide a basis to generate resolutions, operator demonstrations will be used. To do so, a number of steps has to be completed. Figure 1 shows the method applied in this study, to base automation on the individual's controlling style. First the ATCo is asked to resolve a number of traffic scenarios. The information that is retrieved from these scenarios is evaluated to identify the most used strategy per traffic scenario. This strategy is then used in a reinforcement learning environment, that replicates the demonstrated strategy as good as possible.

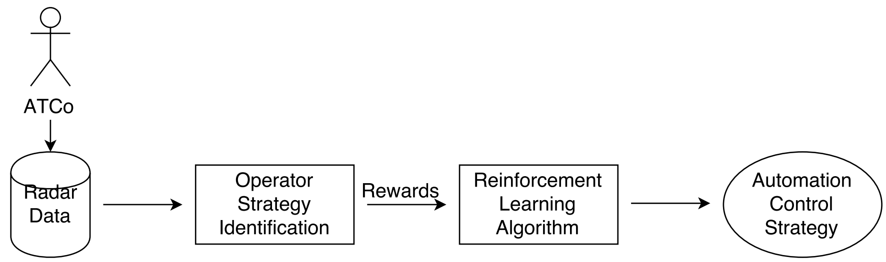


Figure 1. Conceptual Design for Building Strategic-Conformal Automation.

B. Scenario Data

Strategic conformal automation is different for each operator and it is therefore crucial to know what strategies the individual operator, for which the automation will be designed, uses. Rather than conducting interviews to find out what strategies to apply in different traffic scenarios, the ATCos are given a traffic scenario and asked to resolve it. Since research has shown that ATCos are not always consistent in their decision-making, a single demonstration of a certain traffic situation is not sufficient. Multiple runs for a certain scenario are required, in order to identify the most used strategy in that traffic scenario.

During these runs the states responsible for the decision-making (see Table 1) are logged throughout the demonstration. The outcome of the demonstrations are thus traces of logged states, which are unique for the strategies applied by the operator. These traces are used by the identification algorithm to distinguish the ATCo's main strategy from possible inconsistent strategies.

C. Strategy Identification

To replicate the operator's decision-making, it is of importance to know what control strategies are used by the operator in the first place. From operational data state traces can be constructed, which are used to identify different control strategies. However, due to inconsistencies in the ATCo's decision-making, these inconsistencies must be distinguished from the main strategy. Since the nature of the inconsistencies is not known beforehand, the strategy identification problem becomes more complicated. It could be that some operators use one main strategy, accompanied by a lesser-used second strategy. But it could also be that an operator uses one main strategy, in combination with, for example, three lesser-used strategies. What does not change is the fact that the identification algorithm must be able to detect and estimate all of the strategies correctly. Given the nature of the problem, the task of identifying the different strategies will be done using unsupervised machine learning, more specifically clustering. The goal of the algorithm is to determine the number of strategies used by the operator, as well as to estimate the average of each of those strategies.

Data clustering is a technique to group data in groups, based on the (dis)similarity of data points. These (dis)similarities are a result of the control actions applied by the ATCos during the CD&R task. One of the criteria taken into account by the ATCos in CD&R tasks in to minimize the requests to the pilots, hence they want to resolve the conflict with as little control actions possible. Therefore a CD&R maneuver usually consists out of two control actions: one to resolve the conflict and one to realign the deviated aircraft back to its correct sector exit point. Looking at those two actions, it is the action to resolve the conflict that is the most characteristic for the strategy applied by the ATCo, hence the (dis)similarities resulting from this action can be best used for strategy identification.

To classify the demonstrated traces into different strategies, the set of states directly after the heading change to resolve the conflicts are used. This means that every state trace is abstracted to this single data point. Having multiple runs per scenario, this leads to a cloud of data points, on which the clustering analysis will be performed.

In the cluster analysis two types of clustering algorithms have been evaluated, k-means clustering,[18] a hard clustering technique, and Gustafson-Kessel clustering,[19] a soft clustering technique, respectively.

In k-means clustering, the algorithm minimizes the distance between k cluster centers and data points. In this application the Euclidean distance norm is used. Data points are assigned to a cluster based on their distance relative to the cluster means. This is indicated by the membership degree μ_{ij}, stating the membership degree of the i^{th} data point to the j^{th} cluster. In hard clustering methods data points are associated to a single cluster, hence the membership degree $\mu_{ij} \in \{0, 1\}$.

Due to the different units of the states, changes are that certain states have a disproportional effect on the clustering results. To tackle this potential problem, all data are normalized first, before running the k-means clustering algorithm.

In Gustafson-Kessel clustering, the algorithm minimizes the distance between k cluster centers and data points as well. Different to k-means clustering is that it is a soft-clustering method, meaning that data points can be (partially) member of multiple clusters (e.g., 20% to cluster 1, 80% to cluster 2), hence $\mu_{ij} \in [0, 1]$. What is also different to k-means clustering is that the distance norm is based upon the covariance matrix, making it suitable to fit non-Gaussian distribution better. Since the shape of the clusters is unknown, Gustafson-Kessel clustering was expected to work well for this application.

Where both methods differ in the way data points are associaed to different clusters, are they similar to the fact that they both require the number of clusters k in which data must be grouped. For doing so, several techniques have been developed to indicate how well the data set is clustered. The elbow method[20] and average silhouette method[21] have been investigated for this application.

The elbow method visualizes the sum of the within-cluster sum of squares for different values of k. The idea behind the method is that when a natural division of data is found, the sum of squares will not decrease much more when adding more clusters. This can be seen as a "kink" in the plot. The optimal amount of clusters is thus located at this kink. However the method somewhat is not always unambiguous[22] and requires human intervention to find the optimal number of clusters.

The average silhouette width method gives an indication of how similar data points in a cluster are versus how different they are compared to data points in other clusters. The higher this contrast is, the higher the average silhouette width is. The optimal number of clusters to group the data set in is found for the k that maximizes the average silhouette width.

To find the best method for identifying control strategies, five data sets have been created, in which different control strategies have been applied. The number of strategies per data set was known beforehand, such that the performance of the mentioned methods could be evaluated.

Table 3. Number of data sets identified correctly

	Elbow method	Average Silhouette Width Method
K-means	3/5	4/5
Gustafson-Kessel	2/5	1/5

The results shown in Table 3 indicate that k-means clustering in combination with the average silhouette width method performs best. This method is therefore used to identify human control strategies. Solving k-means clustering problems is done via an iterative solver, given by Algorithm 1.

Initialize k means arbitrarily;
repeat
Associate data points to cluster, by analyzing to which mean they are the closest:
$c_i = \arg\min_j ||x_i - v_j||, \quad 1 \leq i \leq N, \quad 1 \leq j \leq k$;
Membership degree of data point i to cluster j:
$\mu_{ij} = \{c_i = j\}, \quad 1 \leq i \leq N, \quad 1 \leq j \leq k$;
Compute new cluster means based on membership degree of the data points:
$v_j = \frac{\sum_{i=1}^{N} \mu_{ij}\ x_i}{\sum_{i=1}^{N} 1}, \quad 1 \leq j \leq k$;
until *Convergence is reached*;

Algorithm 1: K-means clustering iteration scheme.

D. Strategy Replication

To replicate the identified control strategies reinforcement learning (RL) is used. RL is well suited for replicating human control strategies, since it is based on animal learning[15,16] and its history with human demonstations (e.g.,[17]). Moreover is it possible to use RL in a model-free environment and still find optimal control policies, meaning it can replicate any type of strategy demonstrated by the ATCos, as long as it is physically possible.

To replicate the ATCo's strategy, Q-Learning will be used,[23,24] a temporal-difference based, model-free RL algorithm. Q-Learning is suited to replicate the demonstrated strategies, due to its model-free approach, which is required since it is not known what strategies to apply given the decision parameters.

To make the trade-off between exploration and exploitation, an ϵ-greedy method is used. The action selection probabilities are given by Eq. (1).

$$a_t = \begin{cases} \arg\max Q(s_t, a_t) & \text{with probability } 1 \text{ - } \epsilon \\ \text{random} & \text{with probability } \epsilon \end{cases} \tag{1}$$

Combining the ϵ-greedy method with Q-Learning will lead to Algorithm 2.

Initialize Q(s,a) arbitrarily;
repeat
Initialize s;
repeat
Take action a, observe r, $s^{'}$;
Choose $a^{'}$ from $s^{'}$ using policy derived from Q (i.e., ϵ-greedy);
$Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma\ max_{a'} Q(s^{'}, a^{'}) - Q(s,a)]$;
$s \leftarrow s^{'}; a \leftarrow a^{'}$;
until s *is terminal*;
until *number of desired episodes is reached*;

Algorithm 2: Q-Learning, an off-policy algorithm[23,25]

To make the learning as efficient as possible, the states in Algorithm 2 are ones that have a direct impact on the reward function, hence a direct impact on the decision-making of the agent. Such states are known as features[26] and the state vector thus only consist out of elements that build the reward function.[27] Looking at the CD&R task, the states mentioned in Table 1 should be used as features. The actions available to the agent consist of altitude, heading and speed changes.

The reward function is responsible for the produced policy and therefore a crucial component in the RL framework. The goal of the ATC automation is to mimic the demonstrated strategies, so the estimated state trace is used as a reference signal, which has to be tracked (as good as possible) by the agent. By tracking the reference signal, the same states will be visited by the RL agent as has been done by the operator, hence the operator's strategy will be replicated. To track this reference state trace, the trace is discretized in N

points and the current state of the agent is compared to all of the points on the reference line. For each of the points, the reward is given by computing the absolute error between the agent's states and the reference states and are multiplied by a weight:

$$r_{reference} = \max\ \left(\alpha \cdot |states_{agent} - states_{reference}|\ \right) \tag{2}$$

Given the N points in which the reference line is discretized, this also gives N rewards. The reward that is fed back to the agent, is the maximum of those N points. Tuning the weights in the reward weight vector α, gives the designer the opportunity to track certain states better than others.

Tracking the reference signal is not the only consideration to take into account. As mentioned before have operators the top priority to ensure that aircraft do not violate separation minimums, while they also try to limit the requests to the pilots. Those two demands give rise to two additional components in the reward function. The first additional term in the reward function is a Loss of Separation (LOS) penalty, which penalizes any LOS that will occur. This to ensure that the policies produced by the agent are free of LOS situations. The reward function that penalizes LOS sitations:

$$r_{LOS} = \begin{cases} 0 & \text{if no LOS} \\ \alpha_{LOS} & \text{if LOS} \end{cases} \tag{3}$$

To take into account the preference of limiting the requests to the pilot, any actions that require a request to the pilot are penalized as well. This assures that the reference signal is tracked as good as possible, while minimizing the requests to the pilots. The reward function that penalizes pilot requests is given:

$$r_{action} = \begin{cases} 0 & \text{if no pilot request} \\ \alpha_{action} & \text{if pilot request} \end{cases} \tag{4}$$

Tracking of the estimated operator strategy, the avoidance of LOS situations and the desire to minimize the number of pilot requests form the complete reward functions:

$$reward = r_{reference} + r_{LOS} + r_{action} \tag{5}$$

IV. Test Case

To test the effectiveness of the machine learning approach a test case is set up. The goal of the test case is to evaluate every aspect in the conceptual design. This section covers the scope and definition of the test case. First, the scope of the test case is given, followed by the definition of the traffic scenario. Next, the states and actions on which decision-making is based will be given. The section will be concluded with the applied operator strategies.

A. Scope

For some traffic situations there exist best practices of what is the best strategy to apply (see Table 2). However there are many more traffic scenarios for which such best practices do not exist. Due to absence of such best practices it can be expected that the diversity in controller strategies is higher, hence those are the most challenging traffic scenarios to automate. The traffic scenario is therefore designed such that it does not have a best practice.

Furthermore has it been found that ATCos prefer altitude resolution the most, since they require the least amount of monitoring effort. To make the problem even more challenging, the possibility to alter the altitude is removed. The CD&R task is thus performed in the horizontal plane. Since commercial aircraft have small speed envelopes, speed resolutions are not effective and are therefore hardly used (see Table 2). Speed changes are therefore removed as well, hence the CD&R task has to be completed using heading changes only.

Due to the absence of a best practice and the limitation to only use heading changes, it is expected to see more inconsistency in control strategies. It can be therefore considered a difficult scenario to develop strategic conformal automation for. If the machine learning algorithms can develop strategic conformal automation for this challenging test case, it is expected that easier traffic scenarios can be automated as well.

B. Conflict Definition

In the test case an operator demonstrates how he or she resolves a 90 degrees conflict, where both aircraft have identical speeds, identical characteristics and the CPA distance is zero nautical miles. For such a conflict no rule of thumb exists (unbiased conflict), hence a bias towards a specific strategy is not expected. If there would have been a speed difference or a non-zero CPA distance, rules of thumb on what strategy to apply exist (biased conflict), and therefore there is an expected bias towards certain strategies. Since the conflict in the test case is unbiased, it can be expected that operators are somewhat inconsistent in their decision on what strategy to use. Figure 2 visualizes the conflict geometry, where the aircraft are shown in the starting position of the scenario. Operators are asked to make sure that the aircraft leave the sector at the correct exit point, without violating separation minimums.

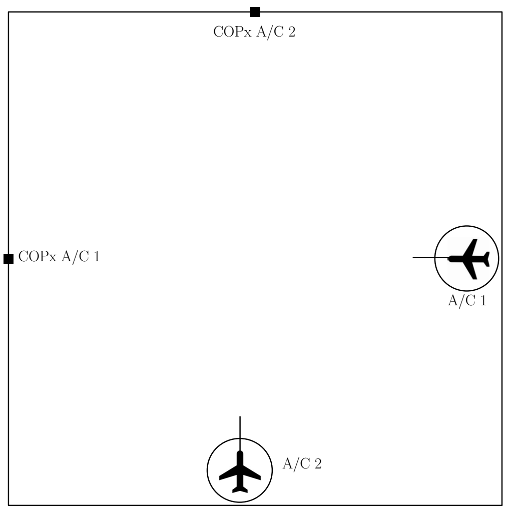


Figure 2. The scenario evaluated in the test case. Aircraft positions show the starting point of the simulation.

One of the motivations for using RL in the automation design is the expected benefit that similar scenarios can be solved similarly, without the need of additional human demonstration. To define what conflicts can be considered similar to the one shown in Figure 2, the conflict types defined by the International Civil Aviation Organization (ICAO) are used as a guideline. The 90 degrees conflict is a so-called crossing conflict. Given the definition of ICAO that conflicts with conflict angles between 45 and 135 degrees (and between 225 and 315 degrees) are all considered crossing conflicts (see Figure 3), this domain is used to evaluate whether the demonstration at 90 degrees can be used to resolve similar conflicts without the need for additional demonstrations.

C. State Selection

The test case consists of a single scenario that will be evaluated. Throughout this scenario some of the parameters that influence decision-making, see Table 1, remain constant. Since the parameters remain constant, they do not have an impact on the decision-making, hence they are not required as states in the RL environment.

Since the agent can only apply heading changes, the speed difference will remain constant during the scenario, hence it has no impact on the decision-making of the agent. The distance on the display is needed to take into account the distance-over-speed bias. However, since only one scenario is tested, the distance-over-speed bias will not have an impact, hence the distance on the display can be omitted as well. The final parameter that has no impact on the decision-making in the test case is the workload. In the single scenario evaluated the workload does not change, hence it is not needed as a state. This will lead to the following states and actions in the Q-Learning environment:

- States:
  - CPA
  - t_{CPA}

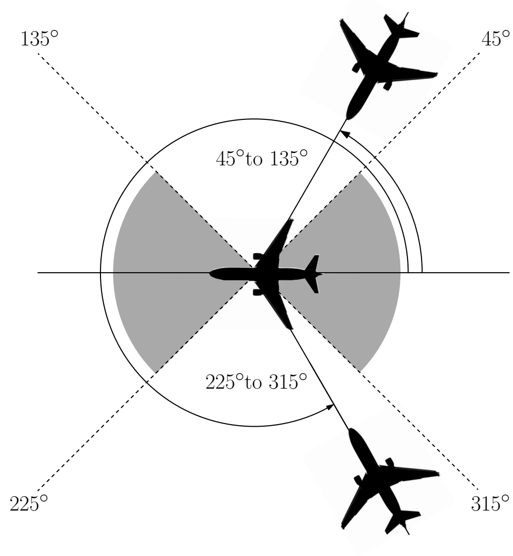


Figure 3. Crossing Conflict Definition by the Internation Civil Aviation Organization.[28]

  - Conflict Angle

- Actions:

  - Heading Changes

To create a policy similar to the operator's decision-making, the estimated state trace from the operator is used as a reference by the RL agent. For this application, the following states are present in the estimated operator profile:

- CPA
- t_{CPA}
- Conflict Angle
- COPx alignment

Comparing these to the states of the RL agent, it can be observed that in the reward function, the COPx (Change Over Point: exit) alignment is added. COPx alignment is included, since the agent should resolve the conflict in similar style as the operator, but must also ensure that the aircraft leaves the sector at the correct COPx. To ensure that this demand is communicated to the agent, the COPx alignment is included.

D. Operator Strategies

The automation bases it control strategy on the demonstrations from the ATCos. In this test case a data set has been created, in which an ATCo that is not entirely consistent is modeled. To create this data set, an operator is asked to resolve the test case scenario, using the following three strategies:

- Main strategy: Late intervention, with small separation buffer, where the aircraft is sent behind the other;
- Second strategy: Early intervention, with large separation buffer, where the aircraft is sent behind the other; and
- Third strategy: Late intervention, with large separation buffer, where the aircraft is sent in front of the other.

Furthermore the operator is asked to only maneuver A/C 2, such that only heading changes of the second aircraft have to be evaluated by the RL agent. If the operator would have the possibility to also maneuver A/C 1, the number of heading changes available to the RL agent is doubled, and COPx alignment of A/C 1 has to be considered as well. This requires the agent to evaluate four times as many action-pairs.

V. Results

This section will discuss the results of building strategic conformal automation for the ATCo and traffic scenario described in the previous section. First, the results of the strategy identification are presented. This is followed by the results of replicating the operator strategy in an identical traffic situation.

A. Strategy Identification

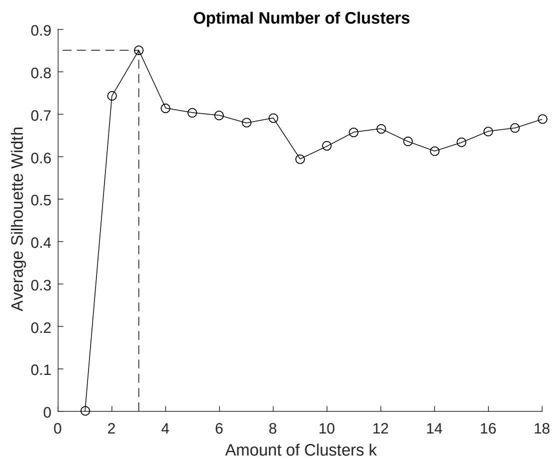


Figure 4. The three identified strategies.

In the test case three different strategies were applied: two strategies that sent one aircraft behind the other (one early intervention what large separation buffer and one late intervention with minimal separation buffer) and one strategy where the aircraft was sent in front of the other, which had a late intervention time and large separation buffer. Looking in Figure 4, it can be seen that the identification algorithm did identify three different strategies. This can be concluded from the fact that the average silhouette width for three clusters achieves a maximum value. This is a good indication that the identification algorithm is suitable for identifying human control strategies.

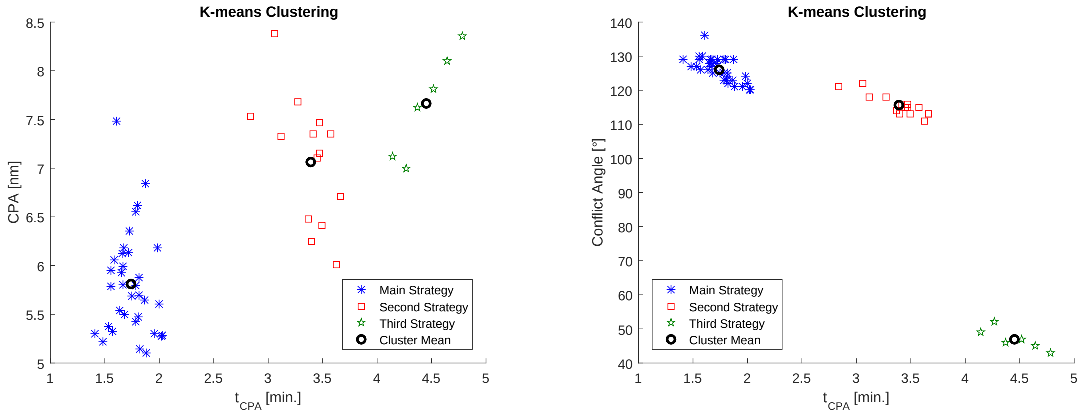
Looking at Figure 5, one can see the results of the found clusters in the CPA-t_{CPA} plane, as well as the conflict angle-t_{CPA} plane. Visual inspection of the clustering result shows that the data points are correctly grouped in the three clusters.

The outcome of the clustering algorithm divides the demonstrated state traces in different strategies. For each of these strategies the state traces are used to estimate the average state trace per strategy. The results of these estimated state traces are shown in Figure 6. The three strategies that have been used in the test case are correctly identified by the algorithm.

B. Strategy Replication

The found main strategy can be used in the RL framework to find a policy that replicates the estimated controller strategy. The policy is formed through the rewards received from the environment, hence the values of the weights in the reward functions have a direct influence on the formed policy. The RL parameters and reward weights to replicate the ATCo's strategy are given in Table 4.

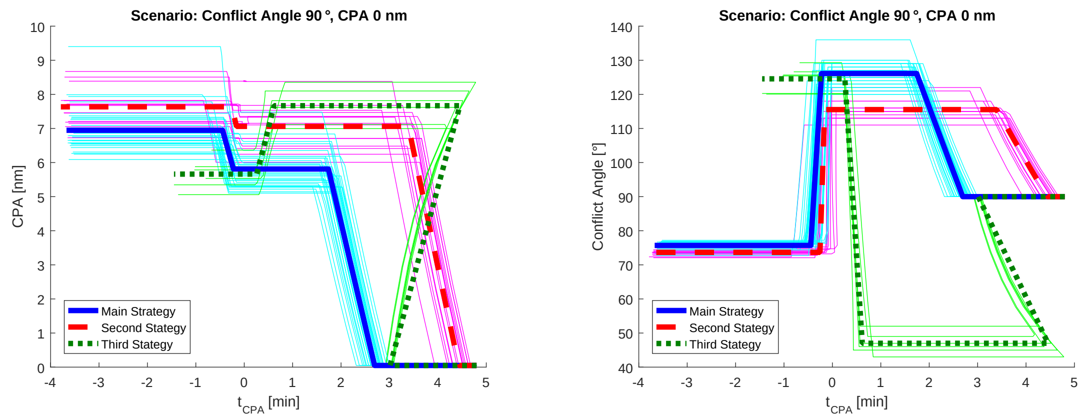
Figure 7 shows the result of replicating the estimated main strategy. It can be noticed that the RL policy is similar to the estimated operator's profile. Since it is rather hard to get a clear picture of what strategy is applied from the state traces directly, a schematic visualization of the strategy is shown in Figure 8.



(a) The data points distributed over the three clusters, as seen in the t_{CPA} - conflict angle plane.

(b) The data points distributed over the three clusters, as seen in the t_{CPA} - conflict angle plane.

Figure 5. The states directly after the first heading change, clustered according to their strategies.



(a) The estimated average strategy traces, as seen in the t_{CPA} - conflict angle plane.

(b) The estimated average strategy traces, as seen in the t_{CPA} - conflict angle plane.

Figure 6. The maneuver points clustered according to their strategies.

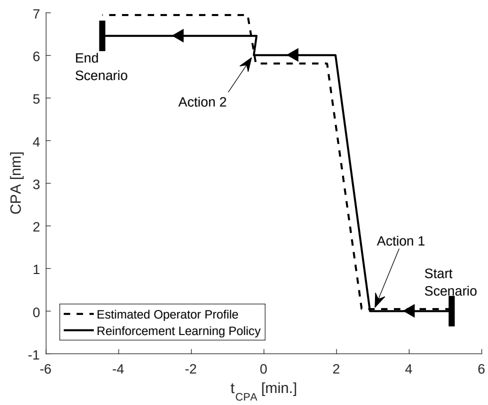
The two heading changes cause the CPA and conflict angle to jump, as can be seen in Figure 7. Action 1 is applied at approximately 3 minutes before the CPA is reached, shifting the CPA from zero nautical miles to approximately six nautical miles (and thus resolves the conflict). Slightly after the CPA is reached ($t_{CPA} = 0$), action 2 is applied to realign the aircraft to its exit point.

VI. Sensitivity Analysis

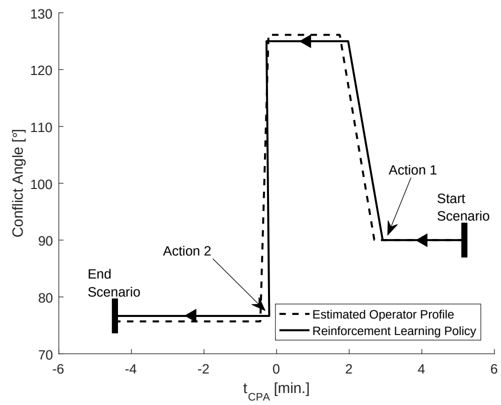
An important reason to use RL for creating strategic conformal automation is the potential benefit of finding similar control strategies in similar conflict geometries, without the need for a demonstration of that particular geometry. This section discusses the sensitivity of the identification algorithm first, followed by the results of using the demonstration from the test case in the crossing conflict domain (see Figure 3). More specifically, the two most extreme cases are discussed: an unbiased conflict at 45 degrees and an unbiased conflict at 135 degrees.

A. Strategy Identification

The identification algorithm did correctly identify the three strategies applied in the test case. Interesting to see is from what point onward the identification is not capable anymore to correctly identify the used



(a) The estimated average strategy (dotted) being replicated by the
RL policy (solid), as seen in the t_{CPA} - conflict angle plane.



(b) The estimated average strategy (dotted) being replicated by the
RL policy (solid), as seen in the t_{CPA} - conflict angle plane.

Figure 7. Estimated controller strategy replicated by the RL agent.

Table 4. The used parameters to replicate the demonstrated strategy.

Q-Learning Settings		Reward Function Weights	
episodes	6,000,000	α_{CPA}	-2
ϵ	0.10	$\alpha_{t_{CPA}}$	-1
α	0.05	α_{CA}	-0.05
γ	0.95	α_{COPx}	-7
		α_{LOS}	-2
		α_{action}	-4

strategies. Multiple data sets of ATCo strategies have been created to test different combinations of strategies, as well as a varying number of strategies per data set, to discover what the limitations of the strategy identification algorithm are.

In a data set that used *four* strategies, in which the aircraft were sent in front of the other (early intervention with small buffer, early intervention with large buffer, late intervention with small buffer and late intervention with large buffer) all four strategies were identified correctly.

In another data set that used *five* strategies, in which the aircraft were sent behind the other (early intervention with small buffer, early intervention with medium buffer, late intervention with small buffer, late intervention with medium buffer and late intervention with large buffer) all five strategies were identified correctly as well.

However, when the data sets were combined, yielding a total of *nine* strategies, the identification algorithm did not identify the nine strategies, but rather found two: one strategy that was the average of all the strategies to send the aircraft in front and one strategy that was the average of the strategies that sent the aircraft behind the other. Clearly, when the ATCo becomes very inconsistent (in this case nine different strategies), the identification algorithm cannot estimate the strategies correctly anymore.

B. Conflict Angle of 45 Degrees

Figure 9 shows the state trace of the RL policy compared to the estimated operator strategy. As one can see there is a big offset when looking in the t_{CPA} - conflict angle plane (Figure 9 (b)). This due to the fact that the starting conflict angle has shifted from 90 degrees to 45 degrees, hence the RL policy line starts below the average controller strategy in the 90 degree scenario. However, the shape of both lines is similar, they

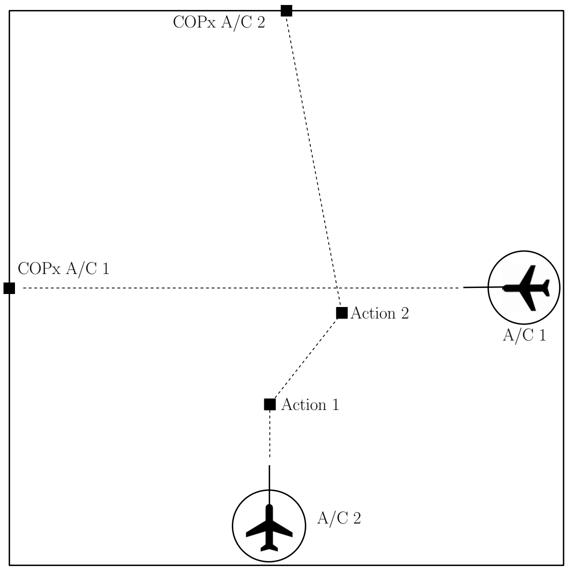


Figure 8. RL policy to resolve an unbiased conflict at 90 degrees.

both increase the conflict angle after the first action applied, indicating that in both cases the controlled aircraft is sent behind the other aircraft.

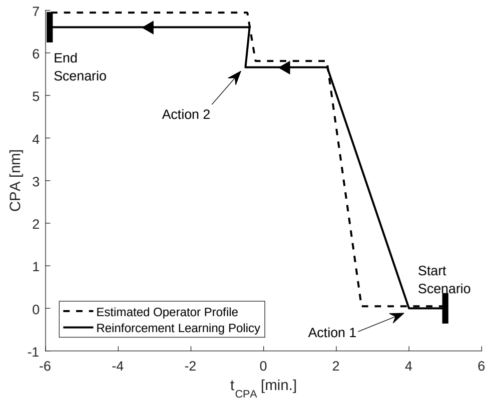
Another observation that can be made from Figure 9 is that the time of intervention has changed. This is most clear from Figure 9 (a), where the time of intervention of the RL agent (approximately $t_{CPA} = 4min$) differed with the demonstrated time of intervention (approximately $t_{CPA} = 3min$). This is the consequence of the altered conflict geometry. With the reduced conflict angle the velocity vectors of the aircraft become more parallel, hence the relative velocity reduces. This mean that given a certain t_{CPA}, the spatial separation between the two aircraft is lower than for higher conflict angles. If the RL agent would wait for the demonstrated t_{CPA}, the aircraft will be so close to each other, that applying a similar resolution as demonstrated will probably lead to a LOS. For this reason the first action of the RL policy has been applied earlier. However, despite interfering earlier, the separation margin after action 1 is still very similar to the demonstrated strategy, as well as the strategy to send the aircraft behind the other. Also the timing of when to realign the aircraft with its exit point is similar. It can therefore be argued that the policy learned at a conflict angle of 45 degrees is still similar to the one demonstrated by the operator at 90 degrees. Figure 10 shows a schematic representation of the CD&R maneuver as would be seen on the radar screen of the ATCo. The RL parameters and reward weights to create the policy for this scenario are given in Table 5.

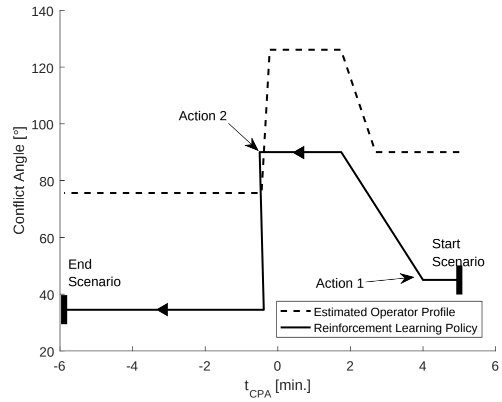
Table 5. The used parameters to create the RL policy at a conflict angle of 45 degrees.

Q-Learning Settings		Reward Function Weights	
episodes	6,000,000	α_{CPA}	-2
ϵ	0.10	$\alpha_{t_{CPA}}$	-1
α	0.05	α_{CA}	-0.05
γ	0.95	α_{COPx}	-7
		α_{LOS}	-2
		α_{action}	-6

C. Conflict Angle of 135 Degrees

Figure 11 shows the RL policy of a 135 degrees conflict compared to the estimated operator strategy at a 90 degrees conflict. Looking in the CPA-t_{CPA} plane (Figure 11(a)), it could be seen that the timing and





(a) The estimated average strategy (dotted) being replicated by the RL policy (solid), as seen in the t_{CPA} - conflict angle plane.

(b) The estimated average strategy (dotted) being replicated by the RL policy (solid), as seen in the t_{CPA} - conflict angle plane.

Figure 9. Estimated controller strategy replicated by the RL agent.

separation margin of action 1 are similar to the demonstrated strategy and the timing of when to realign the aircraft again is also similar. This suggests similar strategies.

However, when looking at Figure 11(b) a remarkable result is found. Now that the conflict is changed to 135 degrees, the starting point of the RL policy traces starts above the estimated controller strategy at 90 degrees. Where in the 45 degrees scenario the shape was still similar, the results in Figure 11(b) are not anymore. Rather than increasing the conflict angle with action 1, the conflict angle is now decreased. This indicates that rather than sending the aircraft behind the other, the aircraft is now sent in front of the other. Even though timing and separation margins are very similar to the demonstrated strategy, the direction (i.e., send in front rather than behind the aircraft) is changed. This is instantly noticed by the operator, and it can be questioned whether or not this strategy is still considered to be strategic conformal.

The change of direction is caused due to the fact that the initial conflict angle of 135 degrees is higher than is reached throughout the scenario demonstrated by the operator. If one looks at the CPA after action 1 is applied, one can see that the CPA of the RL policy is equal to the CPA of the demonstration. There are two strategies to reach this CPA distance: one strategy sending the aircraft behind the other (as done in the demonstration) and one to send the aircraft in front (the found RL policy). Reason why the RL agent decides to send the aircraft in front, has to do with the following: since the initial conflict angle of 135 degrees is already higher than the maximum conflict angle achieved throughout the demonstration, a strategy to send the aircraft behind the other will increase this error even more. This will lead to a big penalty in terms of tracking the conflict angle, while a resolution to send the aircraft in front reduces the conflict angle, hence the absolute conflict angle error is lower. For this reason is the direction of the solution changed.

So the point by which the solutions change "direction" has to do with the maximum conflict angle encountered in the demonstration. If the operator would have used a bigger heading change in his demonstrations and therefore lifting the maximum conflict angle above 135 degrees, the solution of the RL agent would probably be to send the aircraft behind. So depending on the operator strategy, the solution will change direction after starting at a too high conflict angle.

In this case the solution to send the aircraft in front of the other is more efficient in terms of additional miles flown. The found solution can be considered as a "better" solution compared to what the operator demonstrated in that regard. However, in terms of strategic conformance the solution might be wrong, since it does not match with the demonstrated strategy.

A schematic representation of the strategy observed on the radar screen is shown in Figure 12.

The RL parameters and reward weights to create the policy at this scenario are given in Table 6.

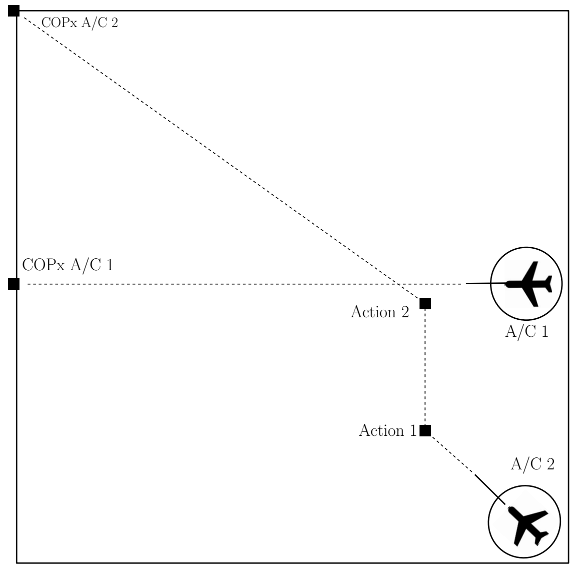


Figure 10. RL policy to resolve an unbiased conflict at 45 degrees.

Table 6. The used parameters to create the RL policy at a conflict angle of 135 degrees.

Q-Learning Settings		Reward Function Weights	
episodes	6,000,000	α_{CPA}	-2
ϵ	0.10	$\alpha_{t_{CPA}}$	-1
α	0.05	α_{CA}	-0.03
γ	0.95	α_{COPx}	-10
		α_{LOS}	-2
		α_{action}	-8

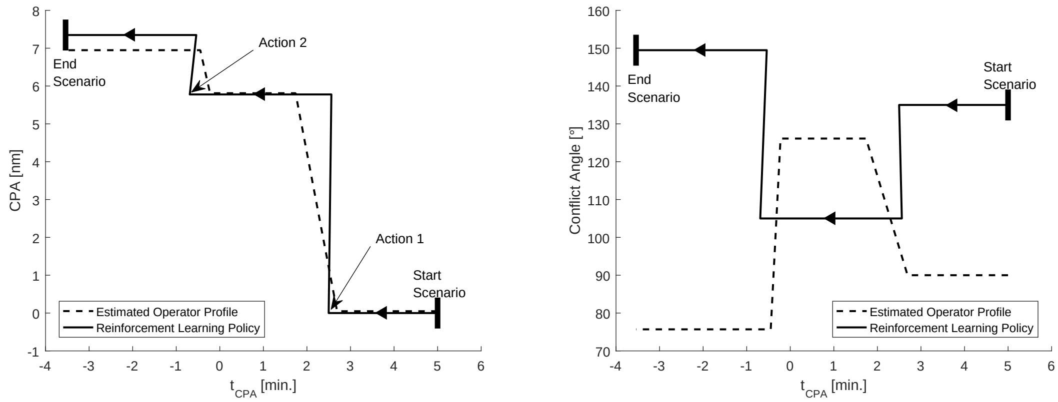
VII. Discussion

This section will discuss the results presented in Sections V and VI. First the results of the machine learning approach are discussed followed by the results of the human strategy identification. Afterwards the results of replicating the strategy in a similar conflict geometry are discussed, followed by a discussion of the sensitivity analysis.

A. Machine Learning Approach

The results have shown that a machine learning approach for creating strategic conformal automation is feasible. Inconsistent data sets are not a problem for identifying used control strategies. Furthermore, it is possible to use RL for creating control strategies in both "nominal" traffic scenarios, as well as scenarios that differed from the demonstrated scenarios. A drawback of using machine learning is that many of the decisions are made by the algorithm itself, without any interference of the operator. In terms of strategy identification this could mean that an operator finds a different number of strategies compared to the clustering algorithm for example.

Another drawback of the method is that it is hard to verify that the learned policy is both safe and similar to the ATCo's decision-making. Part of the problem is that the policy is stored in a big look-up table, from which the agent selects the "best action" based on the current states. The action associated with the highest expected rewards is the action that the agent will select. Although it is extremely simple to evaluate the action to apply at the current state, it is more difficult to evaluate the complete strategy following from the



(a) The estimated average strategy (dotted) being replicated by the RL policy (solid), as seen in the t_{CPA} - conflict angle plane.

(b) The estimated average strategy (dotted) being replicated by the RL policy (solid), as seen in the t_{CPA} - conflict angle plane.

Figure 11. Estimated controller strategy replicated by the RL agent.

current state. It is therefore hard to determine whether or not the developed policies will violate separation minimums. An option to get better inside in the RL policy is to visualize it using behavior trees.[?]

In the implementation of the automation in this paper, it has been chosen to train the agent based on the most used strategy by the operator; the lesser used strategies are not taken into account in the developed policy. However, from the strategy identification algorithm it is found that the main strategy is used in 62% of the demonstrations, whereas the second and third strategy are used in 27% and 11% of the cases, respectively. To take into account all demonstrated strategies, an option would be to train a policy for each of the identified strategies. The automation will then select one of the developed policies, based on the probabilities demonstrated by the operator. Disadvantage of this probabilistic approach is that the automation loses its deterministic character, which might influence the ATCo's acceptance. Another downside is that if the ATCo would make a mistake in the demonstrations, the identification algorithm might detect this as an additional strategy, and this mistake is then also included in the automation.

The most crucial aspect of the strategic conformal automation is the (potential) increase in automation acceptance. Where this paper discusses an approach how to design strategic conformal automation, it does not include an analysis whether or not the chosen approach leads to higher acceptance of automation (see Figure 13). It is important to investigate the effect of the current approach on automation acceptance, since this is the reason why strategic conformal automation is developed in the first place.

B. Strategy Identification

The results of this study are promising in order to identify human control strategies, since the three deliberately used strategies in the test case have been identified. Both the amount of strategies (three) as well as the estimate of each of the strategies has been identified.

Apart from the data set used in the test case, different data sets have been constructed and used to test the identification algorithm. From these data sets it was found that when operators used up to five different strategies in a single scenario, the found strategies were indeed the strategies used by the operator. However when the operators applied more than five different strategies for resolving the same conflict, the identification algorithm loses performance. In the data sets tested, this meant that strategies were grouped in strategies sending aircraft behind the other and strategies sending the aircraft in front of the other.

However this limitation is not a real practical concern. When operators use more than five different strategies in a conflict geometry, this is an indication that they are not consistent in their decision-making. Since it is operator consistency in the first place that is required to introduce strategic conformal automation

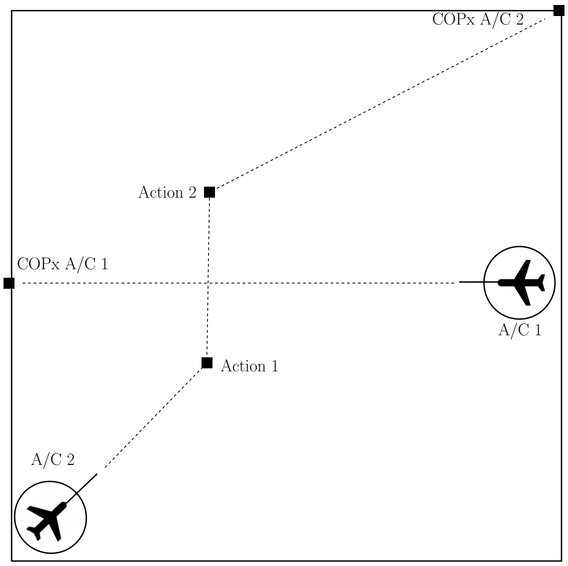


Figure 12. RL policy to resolve an unbiased conflict at 135 degrees.

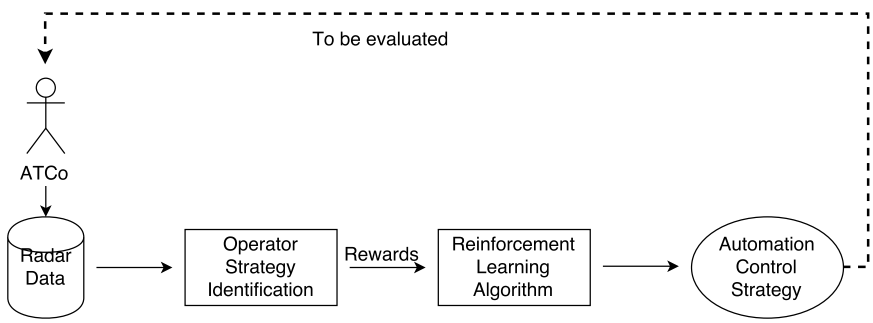


Figure 13. Future research should investigate the effect of the strategic conformal automation on automation acceptance.

successfully,[3] these operators can be considered unsuitable for strategic conformal automation.

Another aspect to consider is the number of demonstrations needed for the same conflict geometry. The dataset in this paper consisted of 55 demonstrations of the same scenario. However when developing automation for professional ATCos, it is unlikely that one has the resources to let ATCos perform scenarios that many times as well, so future research should focus on how many demonstrations per scenario are required, to still identify the used ATCo strategies correctly. Multiple demonstrations are still required, to account for the possible operator inconsistencies.

The most important aspect that still has to be investigated is how many different geometries should be resolved by the operator, in order to reconstruct strategic conformal strategies throughout the complete domain. Different conflict geometries are expected to lead to different resolutions. The International Civil Aviation Organization (ICAO) classifies three different types of conflicts: crossing conflicts, reciprocal track conflicts and same track conflicts.[28] This is a very general division however, in which ATCos could have different strategies in different parts of the defined conflict domains.

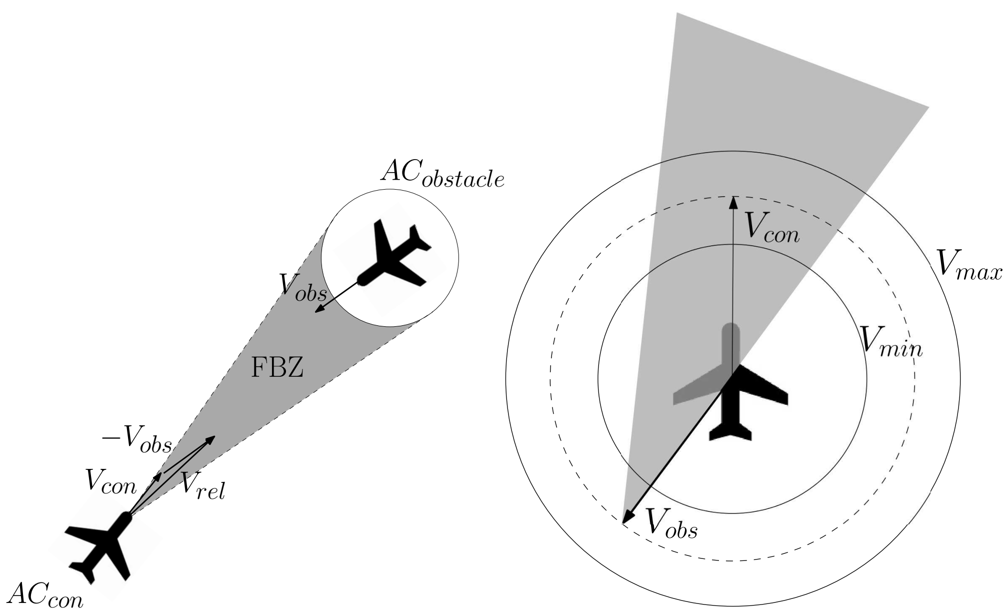
If one looks at initial CPA distance, different behavior can be expected for biased conflicts versus unbiased conflicts. And not every biased conflict is the same, potentially leading to different strategies for different types of biased conflicts. Ideally one would have demonstrations for every possible conflict geometry, however this is simply not possible.

But not only the conflict geometry determines what strategies are applied by the ATCos. In the test case scenario both aircraft had identical performance characteristics, but in real-life air traffic consists of multiple types of aircraft, each with their own characteristics. These performance characteristic are expected to influence the decision-making of the ATCo. If one aircraft can maneuver easier than the other, it is likely that this influences the ATCo in his control strategy.

Apart from aircraft differences, there could be sector specific procedures, which could potentially lead to different control strategies per sector, requiring a different RL policy for each sector as well. These additional parameters could all influence the ATCos' decision-making, but are not taken into account by the factors found in literature (see Table 1). Future implementations of the strategic conformal automation should therefore take these considerations into account as well.

C. Strategy Replication

The strategy replication in an identical traffic scenario looks promising. The RL policy approximated the estimated ATCo's strategy rather well. Given a simulation time step of 15 seconds and the discretized implementation of the Q-Learning algorithm, an exact match between RL policy and the demonstration is hard to achieve. By reducing the simulation time step and increasing the resolution of both states and actions, a better approximation of the operator's strategy will be achieved, at the cost of increasing number of action-values to be evaluated.[29]



(a) Forbidden Beam Zone (FBZ). (b) SSD for AC_{con}.

Figure 14. A two aircraft conflict, showing the FBZ and the SSD for AC_{con}. Adapted from[30]

.

The RL policy is found by the agent by exploring what actions lead to the highest expected future rewards. The learning problem is considered as a model-free problem, but for some of the heading changes it is known beforehand that they are not sufficient to resolve the conflict. In the work by Rahman et al.[30] a solution space diagram (SSD) is used in CD&R tasks, which presents to the ATCo which heading changes will resolve the conflict and which will not.

Figure 14 shows an example of an SSD output in a two aircraft conflict situation. A loss of separation will occur if the relative velocity vector of the two aircraft is inside the Forbidden Beam Zone (FBZ). As the figure shows will a LOS occur, since the relative velocity vector is within the FBZ. Looking at the corresponding SSD display this can be observed as well, since the velocity vector of the controlled aircraft (V_{con}), lies within the shaded area of the SSD. This shaded area contains the set of heading and speed combination that will lead to a LOS. So in order to resolve the conflict, the ATCo has to make sure that the aircraft's velocity vector is outside this shaded grey area.

The information from the SSD display can be used by the RL agent as well, since the actions that are not capable of resolving the conflict are not the ones that will appear in the final policy. This will reduce the amount of state-actions to be evaluated, such that the learning can be done in fewer episodes. Downside of adding this information to the RL environment is that the simplicity of the RL algorithm is sacrificed for the sake of less episodes required. Moreover is there the possibility that these additional constraints will interfere with the learning process of the RL agent, leading to different outcomes of the algorithm.

This will be illustrated using an example: Looking at the scenario replicated in the test case, see Figure 7, it can be observed that the operator does not react immediately to the conflict, but rather waits a few minutes before resolving the conflict. When using the SSD information, the agent will get the information that the zero degrees heading change does not resolve the conflict, since the CPA remains at zero nautical miles. Therefore a zero degree heading change will be seen as an action that does not resolve the conflict and hence should not be applied. Looking at the human demonstrations however, it can be seen that the ATCo does not apply a heading change immediately, despite being in the shaded area of the SSD. So apparently it is not a problem to be in some parts of the SSD.

One option to allow the RL agent to be in certain parts of the shaded area of the SSD, is by adding an additional rule, that states when the t_{CPA} is in a certain interval, the RL agent is allowed to use heading change that do not resolve the conflict directly. However, this directs the automation design towards a rule-based approach, eliminating the simplicity of the RL approach. Those additional rules are not needed at all, since the RL agent can learn by itself which heading changes do or do not resolve the conflict, it only requires some additional computational resources. If one would like to use SSD information to speed up the learning process, it should be thoroughly investigated how this information should be used.

The traffic scenario evaluated in this paper consisted of two aircraft, since ATCos resolve conflicts in a pair-wise manner. In real-life traffic scenarios however, traffic scenarios consist of multiple aircraft in a sector, leading to multiple aircraft pairs. The way to resolve conflicts is still the same, since the policy applies to an aircraft pair, however it could be that two pairs require an action at the same time. One option to deal with such a situation is to apply two actions at the same time, but this could be confusing for the operator. Another option could be to apply the control actions sequentially, but then the problem occurs of which of the actions to apply first.

Another (potential) problem that should be looked into is that resolving one conflict via the developed policy could create another conflict elsewhere. Before the approach discussed in this paper can be applied to real-life applications, this kind of challenges have to be overcome first.

D. Sensitivity Analysis

The results from the sensitivity analysis showed that demonstrations from one conflict geometry can be used to successfully resolve similar conflicts (i.e., they do not violate the separation minimums and with minimum number of pilot requests). However, due to the different conflict geometry, the applied strategy by the RL agent is not identical to the one demonstrated by the operator. While using the ICAO definition to define what conflicts can be considered similar, there does not exist a metric to classify resolution based on their similarity. The easiest classifier to determine similarity between strategies is to look whether resolutions have the same direction (e.g., send the aircraft behind the other), but the time when to interfere and the separation margin are also important criteria. However, due to the lack of an objective metric, resolutions in this paper are judged on similarity by looking at the traces. To judge resolutions on their similarity, a metric should be found or developed, such that manual inspection of every test case is not required.

Another aspect that requires additional research is the weight selection for the components in the reward function. It can be seen that in the cases with a conflict angle of 45 degrees and 135 degrees, the used weights of the reward functions differ. The weights in this paper have been found using a trial and error approach, but for real-life applications this is a very time consuming activity given the many regions to train for. It is recommended that in future automation development the weight selection can be done via an automated process.

VIII. Conclusion

This study has taken a first step to investigate how machine learning can be used to create strategic conformal automation for Air Traffic Control. A method has been developed to translate human decision-making into machine learning decision-making. An algorithm that combines clustering and reinforcement learning has been proposed, to identify and replicate human control strategies from logged human control strategies.

From a test case it was found that the algorithm is capable of identifying and replicating human control strategies. The logged control strategies been used to create similar control strategies for similar conflict geometries. It was found that the strategies in those scenarios looked similar by visual inspection, but without a metric to objectively confirm this statement, it is hard to make this claim. The results however seem to confirm that machine learning is a possible methodology to create strategic conformal automation for Air Traffic Control. The developed control strategies could be used to provide the ATCo with strategic conformal advisories.

References

[1]Hilburn, B., Westin, C., and Borst, C., "Will Controllers Accept a Machine That Thinks Like They Think ? The Role of Strategic Conformance in Decision Aiding Automation," *Air Traffic Control Quarterly*, Vol. 22, No. 2, 2014, pp. 115–136.

[2]Billings, C. E. C. E., *Aviation automation : the search for a human-centered approach*, Mahwah, N.J. : Lawrence Erlbaum Associates Publishers, 1997, Includes bibliographical references (p. 337-348) and indexes.

[3]Westin, C., Borst, C., and Hilburn, B., "Strategic Conformance: Overcoming Acceptance Issues of Decision Aiding Automation?" *IEEE Transactions on Human-Machine Systems*, , No. February 2016, 2015.

[4]Bekier, M., Molesworth, B. R. C., and Williamson, A., "Tipping point: The narrow path between automation acceptance and rejection in air traffic management," *Safety Science*, Vol. 50, No. 2, 2012, pp. 259–265.

[5]Seamster, T. L., Redding, R. E., Cannon, J. R., Ryder, J. M., and Purcell, J. A., "Cognitive task analysis of expertise in air traffic control," *The International Journal of Aviation Psychology*, Vol. 3, No. 4, 1993, pp. 257–283.

[6]Kirwan, B. and Flynn, M., "Investigating Air Traffic Controller Conflict Resolution Strategies," Tech. rep., Eurocontrol, Brussels, 2002.

[7]Xu, X., Rantanen, E. M., and Wickens, C. D., "Estimation of conflict risk using cockpit displays of traffic information," Tech. rep., University of Illinois at Urbana-Champaign, Aviation Human Factors Division, Urbana-Champaign 1 Airport Road Savoy, Illinois 61874, 2004.

[8]Remington, R. W., Johnston, J. C., Ruthruff, E., Gold, M., and Romera, M., "Visual search in complex displays: factors affecting conflict detection by air traffic controllers." *Human factors*, Vol. 42, No. 3, 2000, pp. 349–366.

[9]Bisseret, A., "Application of signal detection theory to decision making in supervisory control The effect of the operator's experience," *Ergonomics*, Vol. 24, No. 2, 1981, pp. 81–94.

[10]Kimball, K. A., "Estimation of intersection of two converging targets as a function of speed and angle of target movement," *Perceptual and Motor Skills*, Vol. 30, No. 1, 1970, pp. 303–310.

[11]Law, D. J., Pellegrino, J. W., Mitchell, S. R., Fischer, S. C., McDonald, T. P., and Hunt, E. B., "Perceptual and cognitive factors governing performance in comparative arrival-time judgments." *Journal of Experimental Psychology: Human Perception and Performance*, Vol. 19, No. 6, 1993, pp. 1183–1199.

[12]Galster, S. M., Duley, J. A., Masalonis, A. J., and Parasuraman, R., "Air traffic controller performance and workload under mature free flight: Conflict detection and resolution of aircraft self-separation," *International Journal of Aviation Psychology*, 2001, pp. 71–93.

[13]Kallus, K. W., Van Damme, D., and Dittmann, A., "Integrated Task and Job Analysis of Air Traffic Controllers - Phase 2 : Task Analysis of En-route Controllers," Tech. rep., Eurocontrol, 1999.

[14]Rantanen, E. M. and Nunes, A., "Hierarchical conflict detection in air traffic control," *The International Journal of Aviation Psychology*, Vol. 15, No. 4, 2005, pp. 339–362.

[15]Samuel, A. L., "Some studies in machine learning using the game of Checkers," *IBM Journal of Research and Development*, 1959, pp. 71–105.

[16]Klopf, A. H., "Brain function and adaptive systems: a heterostatic theory," Tech. rep., DTIC Document, 1972.

[17]Schaal, S., "Learning from demonstration," *Advances in neural information processing systems*, 1997, pp. 1040–1046.

[18]Hartigan, J. A. and Wong, M. A., "Algorithm AS 136: A k-means clustering algorithm," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, Vol. 28, No. 1, 1979, pp. 100–108.

[19]Gustafson, D. E. and Kessel, W. C., "Fuzzy clustering with a fuzzy covariance matrix," *Decision and Control including the 17th Symposium on Adaptive Processes, 1978 IEEE Conference on*, Jan 1978, pp. 761–766.

[20]Thorndike, R. L., "Who belongs in the family?" *Psychometrika*, Vol. 18, No. 4, 1953, pp. 267–276.

[21]Rousseeuw, P. J., "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of computational and applied mathematics*, Vol. 20, 1987, pp. 53–65.

[22]KETCHEN, D. J. and SHOOK, C. L., "THE APPLICATION OF CLUSTER ANALYSIS IN STRATEGIC MANAGEMENT RESEARCH: AN ANALYSIS AND CRITIQUE," *Strategic Management Journal*, Vol. 17, No. 6, 1996, pp. 441–458.

[23]Watkins, C. J. C. H., *Learning from delayed rewards*, Ph.D. thesis, University of Cambridge England, 1989.

[24]Watkins, C. J. C. H. and Dayan, P., "Q-learning," *Machine learning*, Vol. 8, No. 3-4, 1992, pp. 279–292.

[25]Sutton, R. S. and Barto, A. G., *Reinforcement Learning: An Introduction*, MIT Press, Cambridge, MA, USA, 1st ed., 1998.

[26]Hutter, M., "Feature reinforcement learning: Part I. unstructured MDPs," *Journal of Artificial General Intelligence*, Vol. 1, No. 1, 2009, pp. 3–24.

[27]Ziebart, B. D., Maas, A. L., Bagnell, J. A., and Dey, A. K., "Maximum Entropy Inverse Reinforcement Learning." *AAAI*, 2008, pp. 1433–1438.

[28]International Civil Aviation Organization, "Air Traffic Management," Tech. rep., ICAO, nov 2001.

[29]Bellman, R., *Dynamic Programming*, Princeton University Press, Princeton, NJ, USA, 1st ed., 1957.

[30]Rahman, S. M. A., Borst, C., Mulder, M., and Van Paassen, M., "Solution space diagram in conflict detection scenarios," *Jurnal Teknologi, 75 (8) 2015*, 2015.