

**Ecological interface design
supporting fault diagnosis of automated advice in a supervisory air traffic control task**

Borst, Clark; Bijsterbosch, Vincent A.; van Paassen, M. M.; Mulder, Max

DOI

[10.1007/s10111-017-0438-y](https://doi.org/10.1007/s10111-017-0438-y)

Publication date

2017

Document Version

Final published version

Published in

Cognition, Technology and Work

Citation (APA)

Borst, C., Bijsterbosch, V. A., van Paassen, M. M., & Mulder, M. (2017). Ecological interface design: supporting fault diagnosis of automated advice in a supervisory air traffic control task. *Cognition, Technology and Work*, 19(4), 545–560 . <https://doi.org/10.1007/s10111-017-0438-y>

Important note

To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

Ecological interface design: supporting fault diagnosis of automated advice in a supervisory air traffic control task

Clark Borst¹  · Vincent A. Bijsterbosch¹ · M. M. van Paassen¹ · Max Mulder¹

Received: 29 April 2017 / Accepted: 4 September 2017
© The Author(s) 2017. This article is an open access publication

Abstract Future air traffic control will have to rely on more advanced automation to support human controllers in their job of safely handling increased traffic volumes. A prerequisite for the success of such automation is that the data driving it are reliable. Current technology, however, still warrants human supervision in coping with (data) uncertainties and consequently in judging the quality and validity of machine decisions. In this study, ecological interface design was used to assist controllers in fault diagnosis of automated advice, using a prototype ecological interface (called the solution space diagram) for tactical conflict detection and resolution in the horizontal plane. Results from a human-in-the-loop simulation, in which sixteen participants were tasked with monitoring automation and intervening whenever required or desired, revealed a significant improvement in fault detection and diagnosis in a complex traffic scenario. Additionally, the experiment also exposed interesting interaction patterns between the participants and the advisory system, which seemed unrelated to the fault diagnosis task. Here, the explicit means-ends links appeared to have affected participants' control strategy, which was geared toward taking over control from automation, regardless of the fault condition. This result

suggests that in realizing effective human-automation teamwork, finding the right balance between offering more insight (e.g., through ecological interfaces) and striving for compliance with single (machine) advice is an avenue worth exploring further.

Keywords Ecological interface design · Air traffic control · Automation · Supervisory control · Sensor failure · Decision making

1 Introduction

Predicted air traffic growth, coupled with economic and environmental realities, forces the future air traffic management (ATM) system to become more optimized and strategic in nature (Consortium 2012). One important aspect of this modernization is the utilization of digital datalinks between airborne and ground systems via automatic dependent surveillance—broadcast (ADS-B). The most important benefit of a digital datalink over voice communication is that it facilitates the introduction of more advanced automation for efficiently streamlining aircraft flows, while maintaining safe separations. However, a prerequisite for the success of such automation is that the underlying data are reliable and accurate.

Field studies reported mixed findings about the accuracy of ADS-B position reports. On the one hand, it has been shown that ADS-B accuracy is already sufficient enough to meet separation standards and thus could eventually replace current radar technology (e.g., Jones 2003). On the other hand, several studies indicated that offsets between radar and ADS-B position reports could reach up to 7.5 nautical miles (Ali et al. 2013; Zhang et al. 2011; Smith and Cassell 2006). Despite the fact that continuous efforts

✉ Clark Borst
c.borst@tudelft.nl

Vincent A. Bijsterbosch
v.a.bijsterbosch@gmail.com

M. M. van Paassen
m.m.vanpaassen@tudelft.nl

Max Mulder
m.mulder@tudelft.nl

¹ Control and Simulation, Delft University of Technology, Kluyverweg 1, 2629 HS Delft, Netherlands

are being undertaken by the ATM community to improve the quality of ADS-B reports, such position offsets do provide an interesting case study for fault detection and diagnosis in an airspace where ADS-B technology is used to augment radar data with auxiliary aircraft data, such as the planned waypoint(s), estimated time of arrival, GPS and/or inertial navigation system positions and indicated air speed. In general, these auxiliary data contain essential information that would let a computer generate optimal solutions to traffic situations. But unreliable data would render such solutions error prone, demanding human supervision for judging the validity of machine-generated decisions and intervene whenever required.

To support humans in this supervisory control task, this article focuses on using ecological interface design (EID) in facilitating fault detection and diagnosis of automated advice in conflict detection and resolution (CD&R), within a simplified air traffic control (ATC) context. Here, a prototype ecological interface, called the solution space diagram (SSD), is used to study the impact of ambiguous data (i.e., radar data mixed with ADS-B data) on error propagation and fault detection and diagnosis performance. More specifically, the role of explicit (and amplified) ‘means-ends’ relationships between the aircraft plotted on the electronic radar display (source: radar data) and the functional information plotted within the SSD (source: ADS-B data) is investigated.

Note that the topic of EID and sensor failure has been studied before, albeit in process control for manual operations of power plants (Burns 2000; St-Cyr et al. 2013; Reising and Sanderson 2004). Here, the emphasis lies on studying the impact of explicit means-ends relations, as opposed to implicit means-ends relations, on judging the validity and quality of automated advice under data ambiguities within a highly automated operational context. The goal of this article is thus to complement aforementioned studies with new empirical insights about the merits of the EID approach in supervisory control tasks, where reduced task engagement, in conjunction with distractions caused by automation prompting action on its advice, could potentially conceal sensor failures.

2 Background

2.1 Ecological interface design

EID was first introduced by Kim J. Vicente and Jens Rasmussen some 25 years ago to increase safety in process control work domains (Vicente and Rasmussen 1992). Since that time, several books (e.g., Burns and Hajdukiewicz 2004; Bennett and Flach 2011) and numerous articles have explored EID in a variety of application

domains [see Borst et al. (2015) and McIlroy and Stanton (2015) for overviews]. In short, the EID framework is focused on making the deep structure (i.e., constraints) and relationships in a complex work domain directly visible to the system operator, enabling the operator to solve problems on skills-, rules- and knowledge-based behavioral levels.

Central in the development of an ecological interface is the abstraction hierarchy (AH), a functional model of the work domain, independent of specific end users (i.e., human and/or automated agents) and specific tasks. In other words, the AH specifies how the system works (i.e., underlying principles and physical laws) and what needs to be known to perform work, but not *how* to perform the work and by *whom*. The goal (and challenge) of EID is then to map the identified constraints and relationships of the AH onto an interface in order to facilitate productive thinking and problem-solving activities (Borst et al. 2015).

A generic template of the AH is shown in Fig. 1. At the top level, the functional purpose specifies the desired system outputs to the environment. The abstract function level typically contains the underlying laws of physics governing the work domain. At the generalized function level, the constraints of processes and information flows inside the system are described. The physical function level specifies processes related to sets of interacting components. Finally, at the bottom level, the physical form contains the specific states, shapes and locations of the objects in the system. It is argued that the AH is a psychologically relevant way to structure information, as it mimics how humans generally tend to solve problems (i.e., top-down reasoning) (Vicente 1999).

The relations between constraints at different levels of abstraction have been coined as ‘means-ends’ relations. In a means-end relation, information found at a specific level of abstraction is related to information at a lower level if it can answer *how* it is accomplished (*by means of...*) and related to higher-level information if it can answer *why* it is needed (*to serve the ends of...*). The importance of the AH,

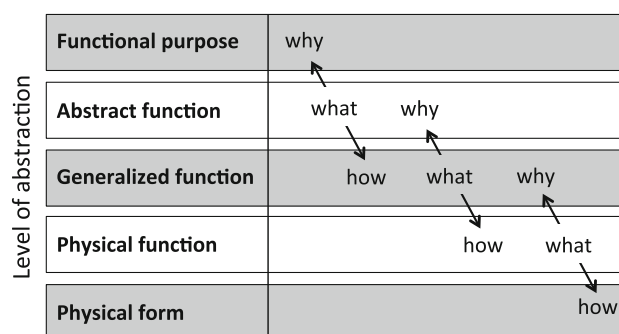


Fig. 1 Rasmussen’s AH, showing means-ends relationships between the levels of abstraction

and how well and complete its constraints and relations are represented on an interface, not only plays a critical role in the success of any ecological interface on decision making, but also on sensor/system failure detection and diagnosis, as will be discussed in the following section.

2.2 EID and fault diagnosis

A general concern about ecological interfaces has been that operators may continue to trust them even when the information driving them is unreliable (Vicente and Rasmussen 1992; Vicente et al. 1996; Vicente 2002). However, several empirical studies have proven otherwise [see Borst et al. (2015) for an overview]. For example, in process control, Reising and Sanderson investigated the differences of an ecological interface over a conventional piping and instrumentation diagram for minimal and maximal adequate instrumentation setups (Reising and Sanderson 2004). Results showed that the maximally adequate ecological interface showed the best failure diagnosis performance over the conventional interface. The main conclusion drawn in this research was that interfaces should display all relevant information to the operator, which becomes crucial in unanticipated events like sensor failures (Reising and Sanderson 2004). Other comparison studies between conventional and ecological interfaces in process control (Christoffersen et al. 1998; St-Cyr et al. 2013) and aviation (Borst et al. 2010) showed similar promising results.

In investigations focused more on the means-ends relations, Ham and Yoon (2001) compared three ecological displays of a pressurized water cooling control system of a nuclear power plant on fault detection performance. The display that explicitly visualized means-ends relations between the generalized and abstract function levels showed a significant increased operator performance, indicating an improved awareness of the system, enabling the human to solve unexpected situations (Ham and Yoon 2001).

Similarly, Burns (2000) investigated the effect of spatial and temporal display proximity of related work domain items on sensor failure detection and diagnosis. Results showed that a low level of integration provided the fastest fault *detection* time, but the most integrated condition resulted in the fastest and most accurate fault *diagnosis* performance. Interestingly, the most integrated display did not show more data or displayed it better, but ‘(...) it showed the data in relation to one another in a meaningful way’ (Burns 2000, p. 241). This helped particularly in diagnosing faults, which required reasoning and critical reflection on the feedback provided by the interface.

To summarize, an ecological interface is not vulnerable to sensor noise and faults by default, but how well and how complete the AH is mapped on the interface plays a fundamental role in diagnosing faults.

2.3 EID and supervisory control tasks

Current empirical investigations on EID and fault diagnosis mainly comprised manual control tasks, where the computer is used for information acquisition and integration to compose the visual image portrayed on the interface. When computers are entering the realm of decision making and decision execution, the involvement of the human operator in controlling a process diminishes, making it seem as if the human-machine interface becomes less important. Paradoxically, with more automation, the role of the human operator becomes more critical, not less (Carr 2014). Consequently, this implies that in highly automated work environments, the need for proper human-machine interfaces only becomes more important (Borst et al. 2015).

With more automation, the human will be pushed into the role of system supervisor, who has the responsibilities to oversee the system and intervene whenever the machine fails. In general, this means that the computer can calculate a specific (and optimal) solution and automatically execute it, unless the human vetoes. Such interaction and role division between human and automated agents are typically captured in levels of automation (LOA) taxonomies (Parasuraman et al. 2000). To successfully fulfill the role as system supervisor, it is thus essential that the human is able to judge the validity and quality of computer-generated advice.

Similar to the success of EID in sensor failure diagnosis, EID may offer a plausible solution in judging the validity and quality of computer-generated advice in supervisory control tasks. In terms of the AH, sensor failures on lower-level components can propagate into wrong computer decisions on higher functional levels. And without an interface that can help to dissect and ‘see through’ the machine’s advice, the more difficult it will become to evaluate its validity and quality against the system’s functional purpose. It is therefore expected that explicit means-ends relations will play an important role in overseeing machine activities and actions. For future air traffic control, which foresees an increased reliance on computers capable of making complex decisions, an ecological interface could help to provide insight into the rationality guiding the automation, resulting in more ‘transparent’ machines (Borst et al. 2015).

3 Ecological interface for ATC

3.1 Work domain of CD&R

In a nutshell, the job of an air traffic controller entails separating aircraft safely while organizing and expediting the flow of air traffic through a piece of airspace (i.e., sector) under his or her control. Controllers monitor aircraft

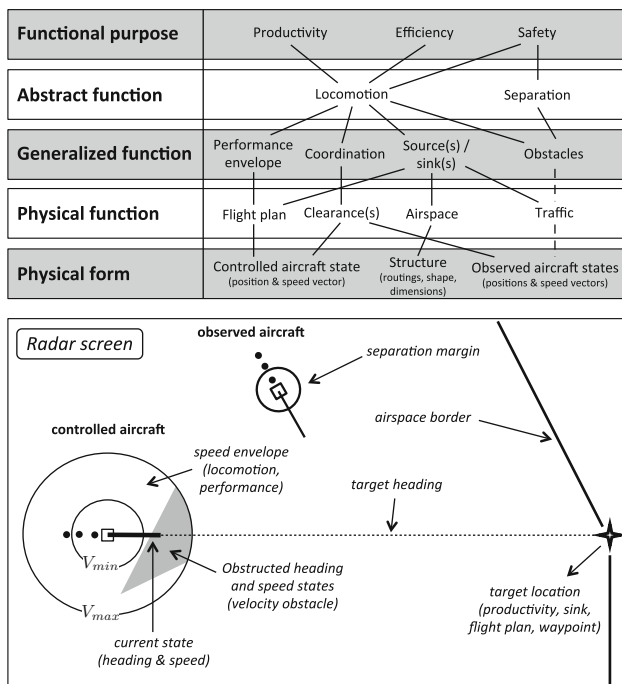


Fig. 2 Abstraction hierarchy, with means-ends relationships, of CD&R for air traffic control along with corresponding interface mappings on a PVD

movements and the separation by using a plan view display (PVD), i.e., an electronic radar screen. The criterium for safe separation is keeping aircraft outside each other's protected zone—a puck-shaped volume, having a radius of 5 nm horizontally and 1000 ft vertically.

In previous research, the work domain of airborne self-separation (i.e., CD&R for pilots) has been analyzed and summarized in an AH (Dam et al. 2008; Ellerbroek et al. 2013). This control problem is similar to the work domain of an air traffic controller, but with the difference that a controller is responsible for more than one aircraft. As such, an adaptation of the AH for self-separation has been made for air traffic control purposes. The resulting AH and the corresponding interface mappings on an augmented PVD are shown in Fig. 2 and will be explained in the following sections.

3.2 Solution space diagram

Central in the AH and the augmented radar screen is the portrayal of locomotion constraints for the controlled aircraft (see Fig. 2). The circular diagram, showing triangular velocity obstacles within the speed envelope of the selected aircraft, is called the solution space diagram (SSD). This diagram integrates several constraints found on lower levels of the AH into a presentation of how aircraft surrounding the controlled aircraft affects the solution space in terms of heading and speed. It is thus essentially a

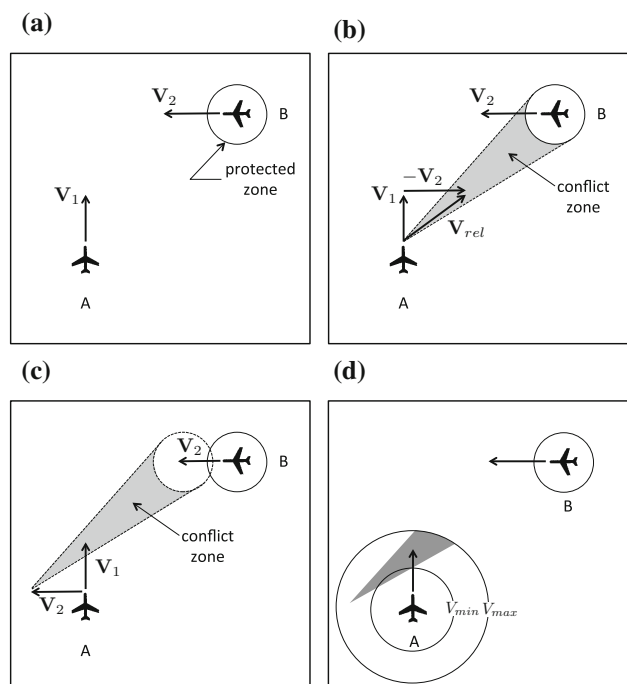


Fig. 3 The solution space diagram (SSD), showing the triangular velocity obstacle (i.e., conflict zone) formed by aircraft B within the speed envelope of the controlled aircraft A. **a** Traffic geometry, **b** conflict zone in relative space, **c** conflict zone in absolute space, **d** resulting solution space diagram for aircraft A

visualization of the abstract function level. The way the SSD is constructed is graphically explained step-by-step in Fig. 3. For more details on the design, the reader is referred to previous work (e.g., Dam et al. 2008; Ellerbroek et al. 2013; Mercado Velasco et al. 2015).

The SSD enables controllers to detect conflicts (i.e., when the speed vector of a controlled aircraft lies inside a conflict zone) and avoid a loss of separation by giving heading and/or speed clearances to the controlled aircraft that will direct the speed vector outside a conflict zone. Any clearance that will move the speed vector into an unobstructed area will lead to safe separation, but may not always be optimal. That is, a safe and productive clearance would direct an aircraft into a safe area that is closest to the planned destination waypoint. A safe and efficient clearance would be one that results in the smallest state change and the least additional track miles relative to the initial state and planned route. Thus, any combination that balances safety (e.g., adopting margins), efficiency and productivity would be possible. The SSD does not dictate any specific balance, but leaves it up to the controller (and his/her expertise) to decide on the best possible strategy, warranted by situation demands.

Linking the conflict zones to their corresponding aircraft on the radar screen is encoded *implicitly* in the SSD, as indicated by the dashed lines in the AH shown in Fig. 2.

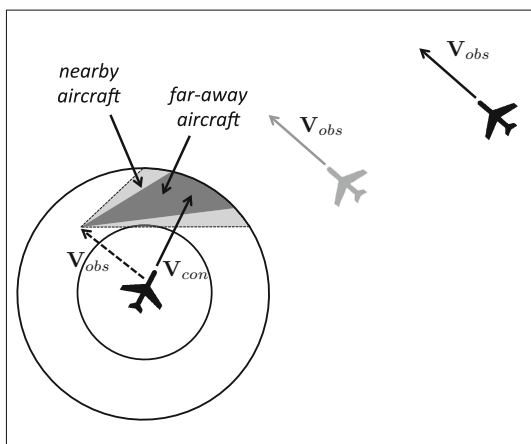


Fig. 4 Implicit means-ends relations between conflict zones and aircraft plots shown on the radar screen

That is, from the shape and orientation of the conflict zone a controller can reason on the locations, flight directions and proximities of neighboring aircraft. In Fig. 4 it can be seen that the cone of the triangle points toward, at a slight offset, the neighboring aircraft and the width of the triangle is large for nearby aircraft and small for far-away aircraft. Additionally, drawing an imaginary line from the aircraft blip toward the tip of the triangle indicates the absolute speed vector of a neighboring aircraft. As such, with the shape and orientation of the conflict zones, a controller would be able to link aircraft to their corresponding conflict zones. Thus, in this way the controller is able to move from higher-level functional information down toward lower-level objects.

3.3 Information requirements and error propagation

Composing the SSD requires information from sensors. In ATC, the sensors for surveillance are the primary and secondary radar systems that, combined, can gather aircraft position, groundspeed, altitude (in flight levels) and call-sign. This information is insufficient to construct the SSD, as it requires accurate information about the aircraft speed envelope (in indicated and true airspeed), destination waypoint(s), flight direction and current velocity.

Currently, the ATM system is undergoing a modernization phase where aircraft is being equipped with ADS-B that can broadcast such information to airborne and ground systems via digital datalinks. Given that continuous efforts are being undertaken to improve ADS-B in terms of reliability and accuracy, it is very likely that position and direction information will still remain available from ground-based radar systems, and that ADS-B is used to augment radar data with auxiliary data, such as GPS position, destination waypoint(s), speed envelopes and

speed vectors. Consequently, this implies that discrepancies between ADS-B and the radar image may arise, resulting in an ambiguity between the aircraft position shown on the PVD (source: surveillance radar) and the representation of the conflict zone (source: ADS-B).

Several studies in Europe and Asia have reported frequently occurring ADS-B position errors reaching up to 7.5 nm (Ali et al. 2013; Smith and Cassell 2006; Zhang et al. 2011). The main causes for ADS-B not meeting their performance standards are: (1) frequency congestion due to other avionics using the same 1090-MHz frequency spectrum, (2) delays in the broadcasted messages and (3) missed update cycles, resulting mostly in in-trail position errors.

In Fig. 5 it can be seen how an in-trail ADS-B position error can propagate into a misalignment of an aircraft conflict zone and its corresponding radar position. Interestingly, the ambiguity between the conflict zone orientation and the radar position creates a false solution space in between the two conflict zones. That is, placing the speed vector of the controlled aircraft into this area will eventually result in a loss of separation with the bottom-right aircraft.

Identifying and diagnosing the validity of the solution space requires the controller to link the conflict zones to the aircraft plots on the PVD. In this case, the ADS-B error would be relatively easy to spot because of the low traffic density. However, one can imagine that under increased traffic density and complexity, the error will be obscured due to a more complex SSD, demanding a more explicit representation of means-ends relations.

Another factor complicating the identification of a position error is the distance between the controlled and observed aircraft. In Fig. 6, it is can be seen that the larger the distance d , the smaller the visual offset angle $\Delta\theta$ of the conflict zone. At distances $d > 50$ nm, an in-trail position offset of $\epsilon = 7.5$ nm will not be noticeable anymore by visually inspecting the SSD. For aircraft separation purposes, the look-ahead time will generally encompass

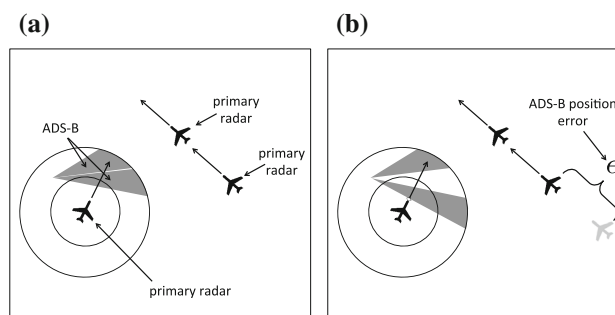


Fig. 5 The effect of an ADS-B in-trail position error (ϵ) on the visualization of the conflict zone within the SSD. **a** Without ADS-B error, **b** with ADS-B error

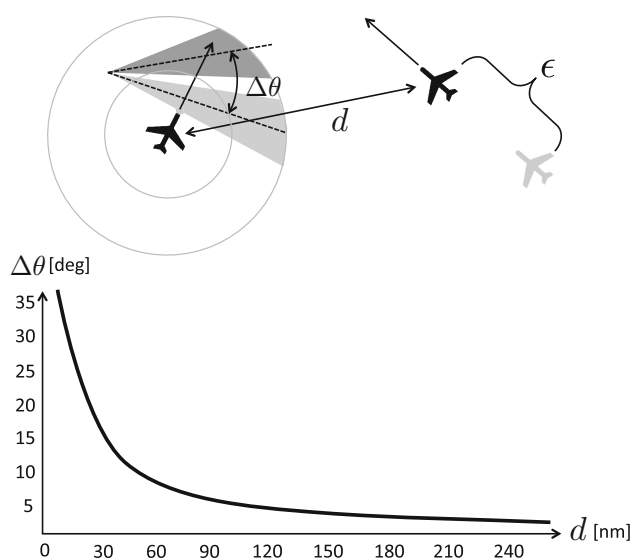


Fig. 6 Sensitivity analysis of the visual offset angle $\Delta\theta$ as a function of distance d , at a fixed in-trail position offset ϵ

5 min, corresponding to approximately 40 nm for a medium class commercial aircraft at cruising speed. Thus also here a more explicit representation of means-ends relations will presumably be helpful in distinguishing between nearby (i.e., high priority) and far-away (i.e., low priority) aircraft/conflicts. In terms of human performance, this could mean the difference between taking immediate action versus adopting a ‘wait-and-see’ strategy.

3.4 Toward explicit means-ends relations

In the study of Burns (2000), means-ends relations were made salient with close spatial and temporal proximity of related display elements. In the SSD, related elements (i.e., aircraft and their conflict zones) already have a close proximity on the electronic radar screen, allowing a controller to link them together, as illustrated in Fig. 4. With more traffic, however, the SSD can also become more complex and cluttered (e.g., overlapping conflict zones), potentially diminishing the benefit of close proximity on fault detection and diagnosis. To negate this effect, it may be required to further amplify the relations between aircraft blips and their corresponding conflict zones.

One way to amplify the means-ends links is by making use of the mouse cursor device, as illustrated in Fig. 7. To support top-down linking, clicking on a conflict zone in the SSD will highlight the corresponding aircraft on the radar screen. To support bottom-up linking, hovering the mouse cursor over an aircraft on the radar screen will highlight its corresponding velocity obstacle. This could enable a controller to more easily match triangles to aircraft, thus expediting the detection of errors/mismatches, especially in more complex traffic scenarios.

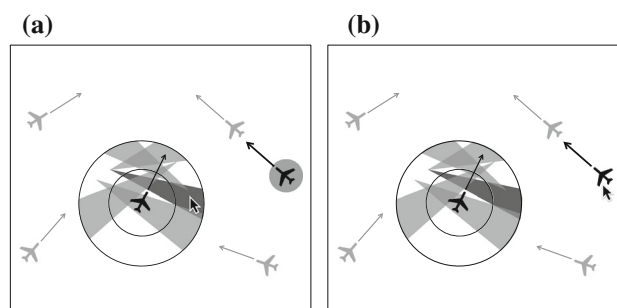


Fig. 7 Explicit means-ends relations, by either **a** clicking on a conflict zone in the SSD (top-down) or **b** by hovering the mouse cursor over an aircraft (bottom-up)

4 Experiment design

4.1 Participants

Sixteen participants volunteered in the experiment, all students and staff at the Control and Simulation Department, Faculty of Aerospace Engineering, Delft University of Technology. All participants were familiar with both the ATC domain (and ‘best practices’ in CD&R) and the SSD from previous experiments and courses, but none of them had professional ATC experience (see Table 1).

Given the goal and nature of this experiment (i.e., studying sensor failures and their impact on judging automated advice when using an ecological display), prior knowledge of and experience with the SSD were required. Concretely, this meant that participants were aware of how the SSD is constructed (i.e., what aircraft information is needed), what information is portrayed, and how to use the SSD to control aircraft, all of which allowed for reduced training time. Note that, in general, ecological interfaces are not intuitive by default and would always require some form of training and deep understanding before people can exploit the power of such representations (Borst et al. 2015).

4.2 Tasks and instructions

The control task of the participants, as illustrated in Fig. 8, was to monitor automation that would occasionally give advice on how to solve a particular conflict (i.e., CD&R) or how to clear an aircraft to its designated exit point. The advisories remained valid for 30 s. During that time, participants needed to diagnose the validity of the advisories (by inspecting the SSDs) and rate their quality, or, level of agreement, by dragging a slider in the advisory dialog window (see Fig. 8). Finally, advisories could be either accepted or rejected by clicking on one of two buttons in the dialog window. In case no accept/reject action was undertaken within those 30 s, the automation would always

Table 1 Participant background information

Profile	9 M.Sc. students, 4 Ph.D. students, 2 Assistant Profs., 1 Full Prof.
Age	22–47 years (average 27)
SSD experiments	1–6 (average 2)

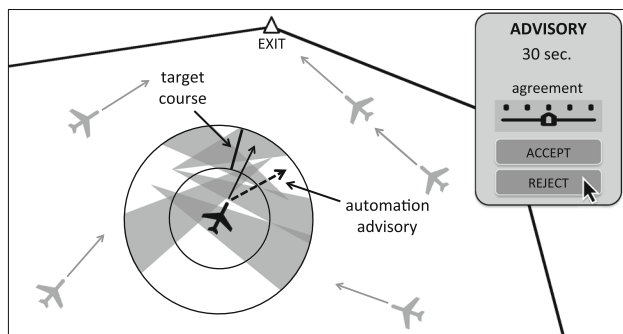


Fig. 8 Simulator screen, showing the control task of participants in which they needed to assess the quality and validity of automated advice and either accept or reject it

automatically execute its intended advice. This level of automation closely resembled ‘Management-by-Exception’ (Parasuraman et al. 2000).

During the supervisory control task, the solution spaces of all aircraft could be examined, but aircraft could not be controlled manually. However, only when an aircraft received an advisory, that particular aircraft became available for manual control, irrespective of the advisory being accepted, rejected or expired. Following an advisory, the color of the subjected aircraft would become, and remain, blue as an indication of that aircraft being available for manual control.

Participants were told that in case of accepting a conflict resolution advisory, the automation would not steer it back to its exit waypoint some time later. Thus accepting such an advisory always required at least one manual control action further on to put the aircraft back on its desired course. Upon rejecting a conflict resolution advisory, at least two manual control actions were required (i.e., resolving the conflict and clearing the aircraft to its exit point). For an exit clearance advisory, either no (accept) or one (reject) manual control action was required. For aircraft under manual control, participants could give heading and/or speed clearances by clicking and dragging the speed vector within the SSD, followed by pressing ENTER on the keyboard to confirm the clearance.

Finally, it was emphasized to the participants that they had to carefully inspect the advisories based on the SSD and the overall traffic situation shown on the radar display. They were told that during the experiment, errors could occur in the position reports needed to construct the SSD (and thus also the advisory), resulting in a mismatch between the radar positions of the conflict zones. Given the

prior SSD knowledge of the participants, they knew that position mismatches could either manifest in off-track or in-trail errors, each having a different effect on the displayed conflict zones (see Figs. 4, 5 for off-track and in-trail offsets, respectively). Participants were, however, unaware of how many aircraft featured an error as well as the exact nature of the position error.

4.3 Independent variables

In the experiment, three independent variables were defined:

1. Availability of amplified means-ends links, with levels ‘Off’ and ‘On’ (between participants),
2. Sensor failure, having levels ‘No Fault’ and ‘Fault’ (within participants) and
3. Scenario complexity, featuring levels ‘Low’ and ‘High’ (within participants).

The rationale for making the amplified means-ends links a between-participant variable was to prevent participants from signaling the absence of the means-ends feature as a system failure. Note that based on an inquiry on the types and number of previous experiments the participants have been involved with (see Table 1), an effort was undertaken to form two balanced groups in order to prevent their prior experiences confound the means-ends manipulation.

The sensor fault always featured an ADS-B in-trail position offset of 7.5 nm, which was found in literature to be a realistic, frequently occurring error. In the scenarios with a sensor failure, only one aircraft emitted incorrect ADS-B position reports and would affect the solution spaces of aircraft receiving an advisory. Additionally, the in-trail position error always made the ADS-B position report lag behind the radar plot (see Fig. 5).

Scenario complexity was a derivative of structured versus unstructured air traffic flows. By keeping the number of aircraft inside the sector approximately equal between two complexity levels, the average conflict-free solution space for the unstructured, high complexity situation was smaller. The rationale for the two traffic structures was that a position offset of an aircraft flying in a stream of multiple aircraft would be easier to spot within a SSD (see Fig. 5) than aircraft flying from and in different directions (see Fig. 7).

4.4 Traffic scenarios and automation

The two levels of scenario complexities were determined by the sector shape, routing structure and the location of crossing points (see Fig. 9).

The sector used for the high complexity condition featured a more unstructured routing network with crossing points distributed over the airspace. This would require participants to divide their focus of attention, or area of interest, potentially inciting more workload. Additionally, a more unstructured airspace would result in aircraft having a reduced available solution space, thus making it more difficult to resolve potential conflicts.

The sector representing low complexity had a more organized airway system with crossing points clustered around the center of the airspace. Finally, each sector (and thus experimental condition) featured on average the same number of aircraft that were inside the sector simultaneously and had approximately the same size.

Two runs with each of the two sectors were performed, i.e., one run for each failure condition. To prevent scenario recognition, dummy scenarios were used in between actual measurement scenarios and measurement scenarios were rotated 180°. For example, conditions ‘High Complexity-No Fault’ and ‘High Complexity-Fault’ both featured sector 2, but rotated over 180°.

In order to test multiple traffic situations per trial, and to keep the trials repeatable and interesting, the simulation ran at three times faster than real time. This resulted in a traffic scenario of 585 s, which ran for 195 s in the simulation. This was chosen such that four consecutive advisories of 30 s could be given without any overlap, with 15 s initial adjustment time, 15 s in between advisories and 15 s manual run-out time after the last advisory. In the scenarios including a sensor failure, three out of four advisories would be affected by this failure and would thus be incorrect.

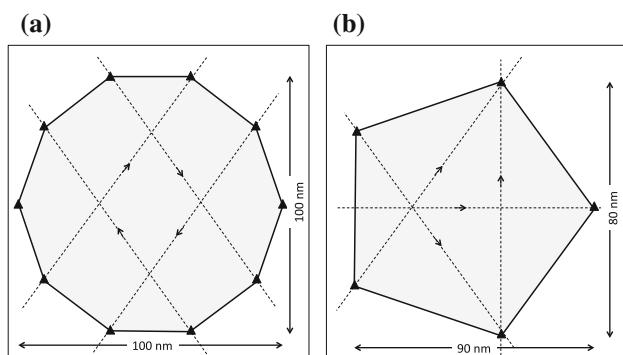


Fig. 9 Sectors used in the simulator trials to manipulate complexity. **a** Sector 1: low complexity, **b** sector 2: high complexity

Table 2 The number and type of advisories encoded in the experiment conditions

Advisory	Means ends off/on				
	Low complexity		High complexity		
	No fault	Fault	No fault	Fault	
Conflict					
Correct	2 (2)	–	2 (2)	–	4 (4)
Incorrect	–	3 (6)	–	3 (6)	6 (12)
Exit					
Correct	2 (0)	1 (0)	2 (0)	1 (0)	6 (0)
Incorrect	–	–	–	–	–
Total	4 (2)	4 (6)	4 (2)	4 (6)	16 (16)

The numbers between brackets represent the minimum number of required manual control actions after the advisories

All advisories were scripted rather than being generated by an algorithm. This simplification was facilitated by both the predefined traffic scenarios and the simulated high LOA (supervisory control task), eliminating the need for designing and tuning a complex CD&R algorithm. This also ensured that each participant received the exact same advisory at the exact same time. However, participants were told that advisories were in fact generated by a computer.

Finally, in Table 2 an overview is provided of the number and type of advisories, organized by experiment condition. In this table, it is also indicated how many manual control actions were minimally required after accepting and rejecting conflict resolution and exit clearance advisories.

4.5 Control variables

The control variables in the experiment were as follows:

- *Degrees of freedom* All aircrafts were located on the same altitude (flight level 290) and could not change their altitude. Thus the CD&R task took place in the horizontal plane only, making it a 2D control task. Note that this simplification ensured more comparable results between participants as they could not change altitude to resolve conflicts whenever they vetoed an advisory.
- *Aircraft type* All aircrafts were of the same type, having an equal speed envelope (180–240 kts indicated airspeed) and turns at a fixed bank angle of 30°.
- *Aircraft count* On average, all scenarios featured 11 aircraft simultaneously inside the sector at all times.
- *Level of automation (LOA)* The chosen LOA for this experiment was fixed at ‘Management-by-Exception’ (Parasuraman et al. 2000), which meant that the advisory would automatically be implemented unless the participant vetoed. The main reason for supervisory

control through Management-by-Exception, instead of complete manual control, was to not only to simulate a highly automated operational environment, but also to keep the evolution of traffic scenarios comparable between participants as much as possible.

- *Automation advisories* All scripted advisories featured a fixed expiration time of 30 s.
- *Interface* The controllers had always access to the SSD. That is, whenever they selected an aircraft, the SSD for that aircraft opened and could be inspected, irrespective of having received an advisory or not.

4.6 Dependent measures

The dependent measures in the experiment were as follows:

- *Correct accept/reject scores* measured if participants accepted the advisory or wanted to implement their own solution(s). These scores have also been used as a proxy for the failure detection performance.
- *Advisory agreement rating* measured the level of agreement with the given advisory, which was measured by a slider bar with scale 0–100 before responding to the advisory.
- *Advisory response time* measured the time between initiation of and response (i.e., accept or reject) to an advisory. An expired advisory would be measured as a 30-s response time.
- *Number of SSD inspections* was recorded to measure how often SSDs were opened, and furthermore how many means-ends inspections were utilized and in what way (top-down versus bottom-up).
- *Sensor failure diagnosis* was measured by using verbal comments (requiring participants to think aloud during the trials) and were noted when the correct nature of the sensor failure (i.e., an in-trail position error lagging behind the radar plot) was detected and the corresponding aircraft identified.
- *Workload ratings* measured the overall perceived workload per trial and were measured using a slider bar with scale 0–100 at the end of each scenario.
- *Control strategy* was measured by eliciting a participant's main strategy from verbal comments and manual control performance.

4.7 Procedure

The experiment started with a briefing in combination with a fixed set of ten training runs, in which the basic working principles of the interface, the automation and the details of the task were discussed. The training scenarios gradually

built up in complexity to the level of the actual trial scenarios. In training, only two sensor failures occurred to demonstrate the importance of carefully inspecting the validity of the SSD (and the given advice) compared to the radar positions.

After the briefing and training, participants engaged in seven measurement scenarios of about 3 min. Of these seven scenarios, four were actual measurement scenarios according to the independent variables and they were mixed with three dummy scenarios (without advisories). The purpose of the dummy scenarios was twofold: (1) prevent scenario recognition and (2) make advisories and sensor failures appear as rare events. After each scenario, participants indicated their perceived workload rating.

When the experiment was finished, a short debriefing was administered, in which participants could provide overall feedback on the simulation and their experience and adopted control strategies. In total, the experiment took about 2.5 hours per participant.

4.8 Hypotheses

It was hypothesized that the availability of explicit means-ends relations, compared to implicit means-ends relations, would result in: (1) an increased number of correctly accepted and rejected advisories, (2) higher agreement ratings for correct advisories and lower ratings for incorrect advisories, (3) lower advisory response times, (4) improved sensor failure diagnosis, (5) reduced number of SSD inspections and (6) a lower number of manual control actions. It was further expected that these results would be more pronounced in the high complexity scenario.

The main rationale for these hypotheses was that the explicit means-ends relations would enable participants to gain a better insight into the traffic situation, leading to more effective fault detection and manual control performance.

5 Results

5.1 Advisory acceptance and rejection

The cumulative advisory acceptance and rejection counts, categorized by experiment condition, are shown in Fig. 10. Note that the maximum count for correctly accepted advisories in scenarios without sensor failure was 32 (four correct advisories times eight participants per group), and for scenarios with a sensor failure, this number was eight (one correct advisory times eight participants per group). For the rejection counts, the maximum count was 24 (three incorrect advisories times eight participants per group).

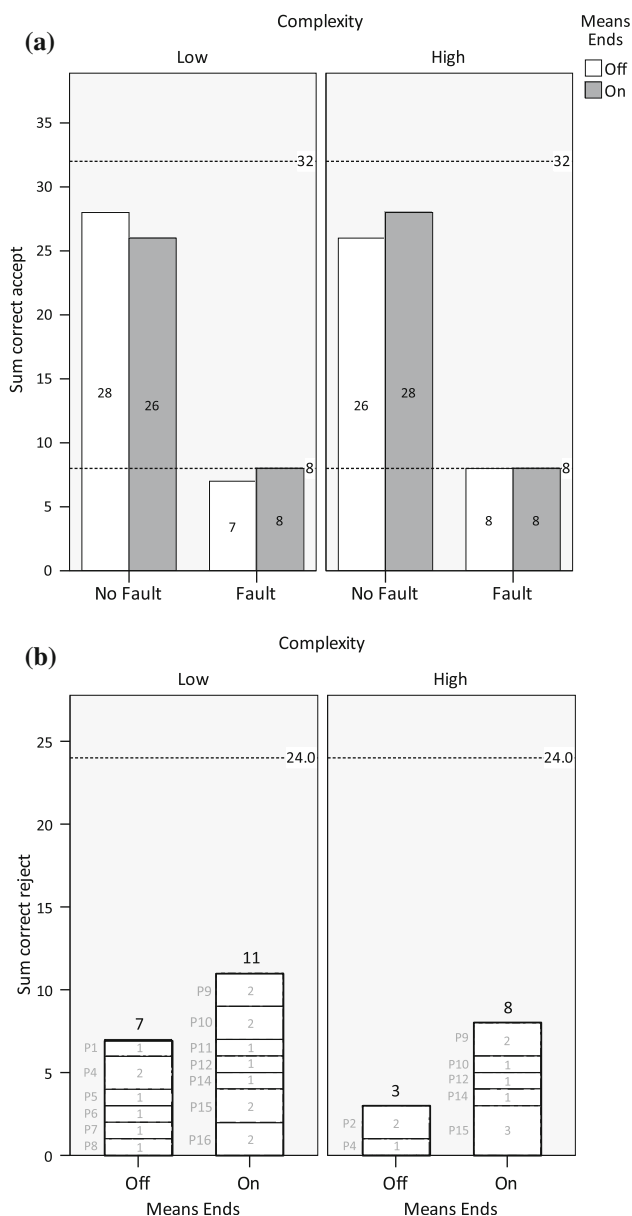


Fig. 10 Number of correctly accepted and rejected advisories, with the rejection counts broken down per participant. **a** Correctly accepted advisories, **b** correctly rejected advisories

In Fig. 10a it can be seen that both means-ends groups accepted the majority of correct advice. As hypothesized, Fig. 10b reveals that group with explicit means-ends links rejected more incorrect advice. Statistically, however, neither significant main nor interaction effects have been found for both the acceptance and rejection counts. From Fig. 10b it is also clear that quite often incorrect advisories have been accepted, given the low total number of rejections. This result can be partially explained by the observed control strategies, in which accepting advisories was often used as a gateway to gain manual control over aircraft (see Sect. 5.8). As such, the acceptance and rejection counts

cannot be considered as good proxies for solely the failure detection performance.

Due to the low number of correctly rejected advisories, it is worthwhile to inspect the individual participant contributions. It can be observed that in the high complexity scenario, more participants of the means-ends group contributed to correct rejections of faulty advice as compared to the means-ends ‘Off’ group. More interestingly, only a few participants (i.e., P2 in group means-ends ‘Off’ and P9 and P15 in group means-ends ‘On’) were successful in rejecting (almost) all incorrect advice. To advance on the way the means-ends linking was used, P9 and P15 were among the participants who predominately used top-down linking by clicking on the triangles (see Fig. 13a). Apparently, for these participants, this strategy led to more successful rejections of incorrect advice. Also note that in the means-ends group, P13 is not at all represented, meaning that this participant always wrongfully accepted incorrect advice. This was also true for P3 in the means-ends ‘Off’ group.

5.2 Advisory agreement

The normalized agreement ratings are shown in Fig. 11. A three-way mixed ANOVA only revealed a significant main effect of sensor failure ($F(1, 14) = 7.017, p = 0.019$) and a significant complexity \times means-ends interaction effect ($F(1, 14) = 5.643, p = 0.032$), but no main effect of means-ends. Thus, a fault condition led to lower advisory agreements and in the low complexity condition, and the agreement ratings of the means-ends group were generally lower, irrespective of fault condition.

On a critical note, the reliability of the agreement ratings can be questioned, given the relatively large spread in the data. This is especially true for the means-ends ‘Off’ group. Similar to the acceptance/rejection counts, the control strategies could explain this spread. That is, not all participants may have been equally thorough in evaluating the advice, especially in the high complexity condition due to experienced time pressure to inspect a more complex SSD and act upon the advisory.

5.3 Advisory response time

The advisory response time was defined as the time between the start of the advisory and it being either accepted, rejected or expired. In the entire experiment, however, not a single advisory expired.

The distributions of the average participants’ response times, categorized by experiment condition, are provided in Fig. 12. Due to violations of the ANOVA assumptions, nonparametric Kruskal–Wallis and Friedman tests were

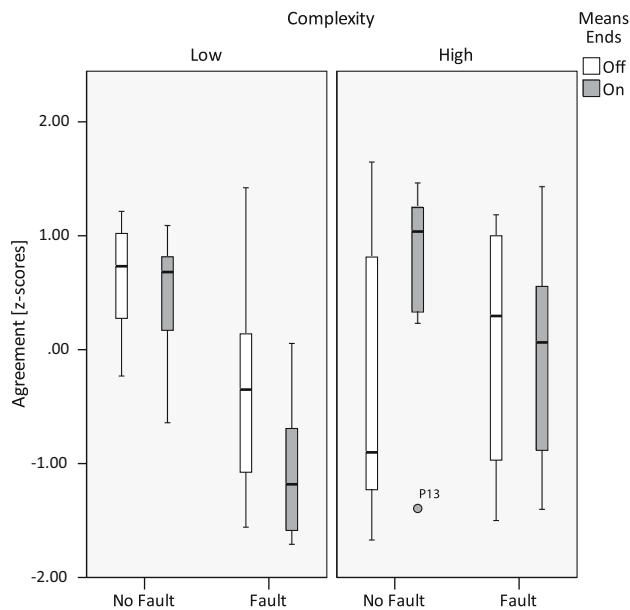


Fig. 11 Advisory agreement

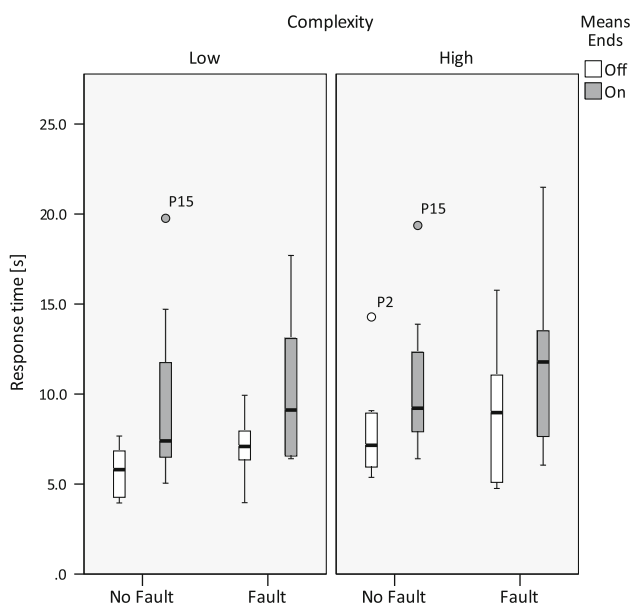


Fig. 12 Average participant advisory response time

conducted for between- and within-participant effects, respectively.

Results revealed only a significant effect of explicit means-ends relations ($H(1) = 3.982, p = 0.046$) in the ‘Low Complexity-Fault’ condition. Here, the means-ends group took longer to act upon an advisory, which runs counter to what was hypothesized. A Friedman test reported a significant effect of the within-participant manipulations ($\chi^2(3) = 9.375, p = 0.025$), where pairwise comparisons (adopting a Bonferroni correction) showed a

significant difference between the ‘Low Complexity-No Fault’ and ‘High Complexity-Fault’ conditions.

The relatively high (variability in) response times were mainly caused by attention switches. Whenever participants were busy manually controlling an aircraft that previously received an advisory, automation could prompt for action on an advisory for another aircraft. In most cases, participants first finished working with the aircraft under manual control before acting upon the advisory. This behavior was observed in both participant groups. In the means-ends ‘On’ group, the unexpected increased activity (discussed in Sect. 5.4) in dissecting the traffic situation (discussed in Sect. 5.8) was responsible for increased response times.

5.4 SSD inspections

The total number of SSD inspections was counted and is displayed in Fig. 13a. Due to violations of the ANOVA assumption, nonparametric tests were conducted. Kruskal–Wallis revealed only a significant effect of means-ends in the ‘Low Complexity-Fault’ condition ($H(1) = 4.431, p = 0.035$), where the means-ends group inspected significantly more aircraft SSDs in contrast to what was hypothesized. A Friedman test on the within-participant manipulations was significant ($\chi^2(3) = 8.353, p = 0.039$). However, pairwise comparisons did not confirm significant differences between conditions after adopting a Bonferroni correction. It can also be observed that the spread for the means-ends group is quite large, especially in the low complexity condition, indicating that not all participants inspected the SSDs equally frequently. A Levene’s test, however, did not mark these spread patterns to be significantly different.

The increased activity of participants in the means-ends ‘On’ group was unexpected and counter to the hypothesis. It was expected that explicit means-ends links would make the search for neighboring aircraft that could be affected by an advisory, more efficient and thus reduce the need to open the SSDs of (many) other aircraft. Instead, it was observed that the means-ends links seemed to have encouraged participants to inspect the SSDs of more aircraft more frequently. This behavior can be partially explained by the observed control strategy as will be discussed in Sect. 5.8.

Another interesting result was found when investigating how the explicit means-ends links were used. Comparing the counts in Fig. 13b with the ones in Fig. 13c, it is clear that hovering (i.e., bottom-up linking) was more frequently used than clicking on the triangles (i.e., top-down linking). This was rather unexpected, because bottom-up linking would take more effort, especially in the complex scenario

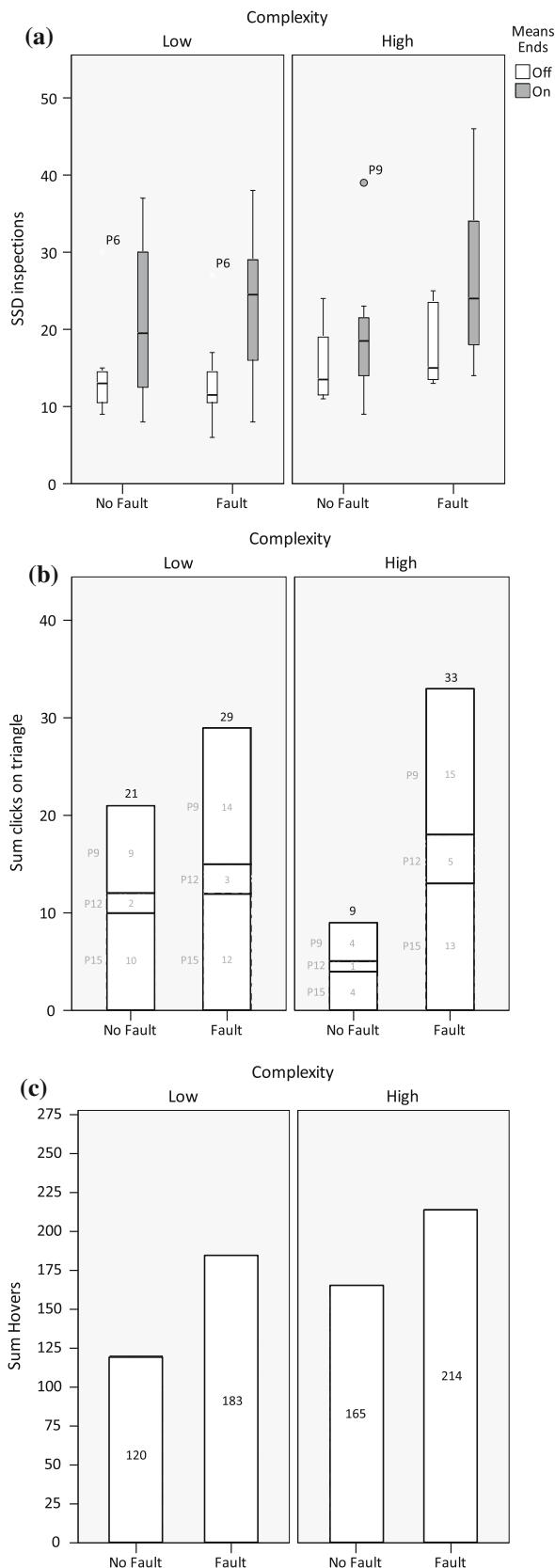


Fig. 13 SSD inspections and means-ends usage. **a** SSD inspections, **b** top-down linking, **c** bottom-up linking

where aircraft was more scattered around the airspace. The fastest way to use the means-ends links would be to first click on the triangles close to the advisory. This would narrow the search down to one or more (in case of overlapping triangles) aircraft on the PVD. Then, the search could be finalized by hovering over the remaining aircraft to link each triangle to its aircraft. Note that in the briefing (and training) prior to the experiment, participants were not instructed on this strategy to avoid biasing the results. Only three participants (P9, P12 and P15) discovered and adopted the preferred strategy, and two of them (P9 and P15) were also the ones who correctly rejected the majority of faulty advice (see Fig. 10b).

5.5 Sensor failure diagnosis

Successful sensor failure diagnosis was established through verbal comments during the experiment, facilitated by participants (in both groups) thinking aloud. A detection was judged to have occurred when the participants found the one aircraft exhibiting a sensor failure and could explain what the nature of the failure was (i.e., in-trail position offset lagging behind the radar position).

The cumulative numbers of successful detections are shown in Fig. 14, where a maximum cumulative result of eight was possible (one ADS-B failure per scenario times eight participants), indicated by the dotted line. Kruskal–Wallis only revealed a significant main effect of means-ends in the high complexity condition ($H(1) = 4.01, p = 0.046$). Interestingly, despite the high success rate for the means-ends group in this condition, this was not reflected in rejection counts. This is yet another indication that the interaction with the advisories itself is not a good proxy for the fault detection performance.

Another interesting observation was that incorrect sensor failure detection involved participants thinking that there was an off-track position error, which would make the triangle appear wider or sharper than necessary (depending on the presumed proximity between the selected and the observed aircraft). In their opinion, this made resolution advisories not necessarily incorrect, but inefficient by taking either too much or too little buffer in avoiding conflicts. This caused them to initially accept an advisory, followed by a manual control action to improve its ‘efficiency.’ This action, however, triggered more manual control inputs moments later, as the real failure had not been properly detected.

5.6 Workload

After each scenario, participants submitted a workload score by means of a slider bar with values varying from 0 (low perceived workload) to 100 (extreme high perceived

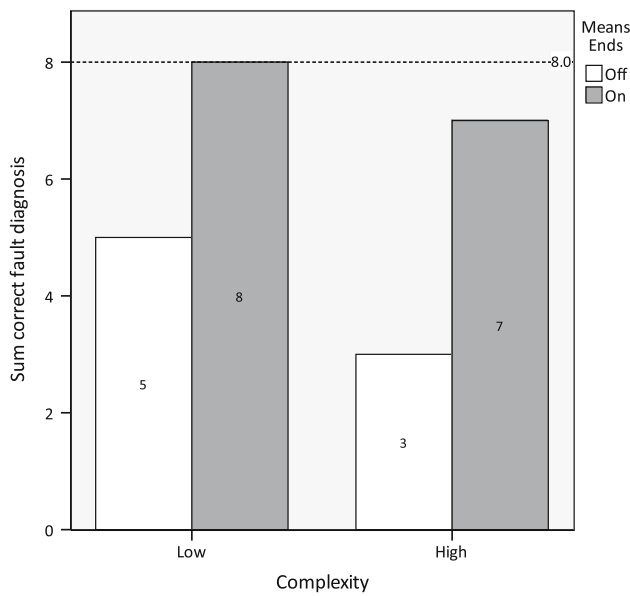


Fig. 14 Correct failure diagnosis

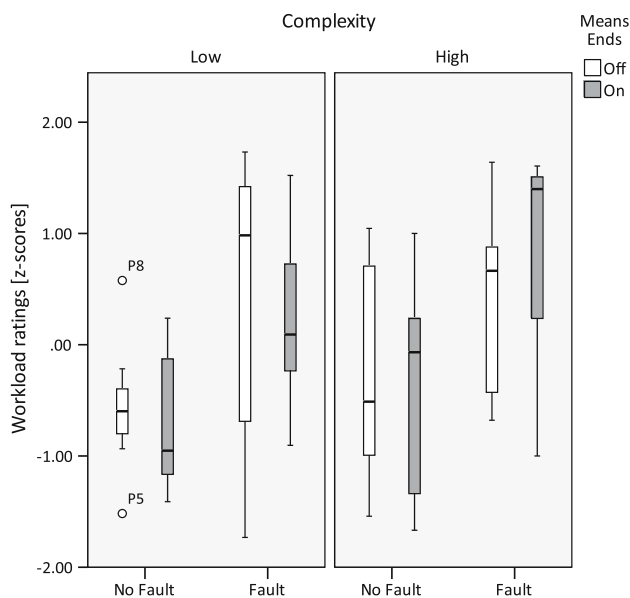


Fig. 15 Normalized workload rating

workload). The normalized workload ratings can be seen in Fig. 15, which reveals that a fault condition resulted in higher workload. A three-way mixed ANOVA indeed revealed a significant main effect of failure ($F(1, 14) = 19.327, p < 0.05$), but also reported neither main nor interaction effects of the means-ends and complexity manipulations. Despite these results, the perceived workload was relatively low, given the nature of the supervisory control task. The participants needed to monitor the traffic scenarios and could only manipulate it when a advisory would pop up for an aircraft that either needed

an exit clearance or a conflict resolution. Note that the choice for this particular control task was made intentionally in order to investigate how low workload conditions (and reduced vigilance) would impact failure diagnosis.

5.7 Manual control performance

Ideally, each participant required minimally 16 manual control actions over the entire experiment (see Table 2), totaling 128 commands per means-ends group. In Table 3, it can be seen that many more commands were given in both groups. The group with explicit means-ends gave fewer commands in total, as hypothesized. This result, however, was not statistically significant.

Interestingly, the means-ends manipulation seemed to have reduced the number of heading-only (HDG) and combined (COMB) commands and increased the number of speed-only (SPD) commands. Note that a combined command featured a single ATC instruction containing both a speed and a heading clearance. The rationale for including such a clearance was that it might say something about the participants' efficiency in 'communicating' with the aircraft.

A graphical depiction of the number and type of commands, distributed over the experiment conditions, is shown in Fig. 16. Recall from Table 2 that in the 'No Fault' condition a minimum of 16 manual control actions was required and for the 'Fault' condition 48 control actions. This means that in the failure condition, approximately 20% more commands were given than necessary, whereas this increase is about 55% in the nonfailure condition. However, the differences in number of commands between the complexity levels within one failure condition were not significant.

When considering the type and number of commands per participant (see Fig. 17), the preference for more SPD clearances in the means-ends group becomes apparent. However, it also becomes clear that not all participants contributed equally to the overall increase in manual control actions. Some participants (i.e., P3, P11 and P12) gave even less clearances than minimally required. They often did not steer aircraft back to their exit point after solving a conflict. Recall that after a conflict resolution advisory, the

Table 3 Number of commands, categorized by type and means-ends condition

Commands	Means ends off	Means ends on	Totals
SPD	24	39	63
HDG	90	73	163
COMB	68	55	123
	182	167	349

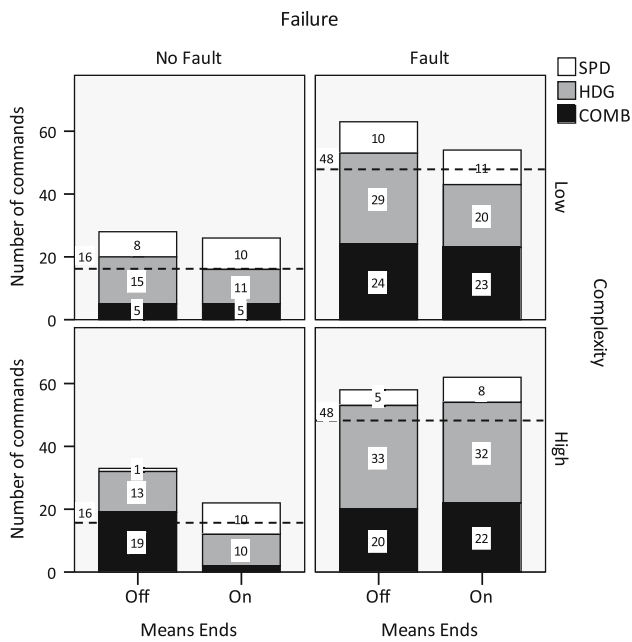


Fig. 16 Total number of commands, categorized by type and experiment condition

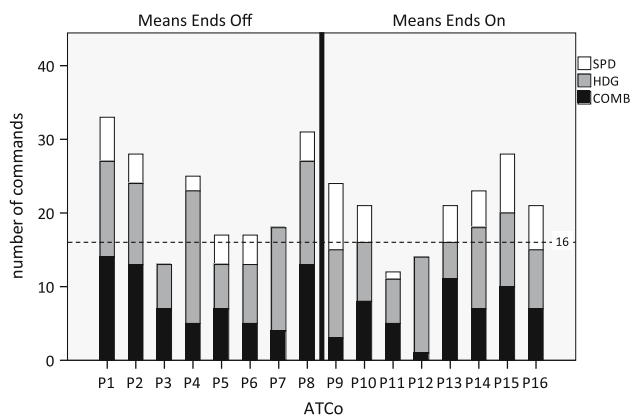


Fig. 17 Number and types of commands per participant

automation would not issue an exit clearance advice and thus it was up to the discretion of the participants to steer aircraft back on course. A few other participants (e.g., P1 and P8) almost doubled the number of clearances as they were continuously trying to ‘optimize’ their conflict resolutions and exit clearances. Note that such individual differences can be expected when using ecological interfaces, as these displays do not dictate one particular course of action.

5.8 Control strategies

Besides the manual control performances, participant strategies were elicited from observations and verbal comments during and after the experiment. A qualitative

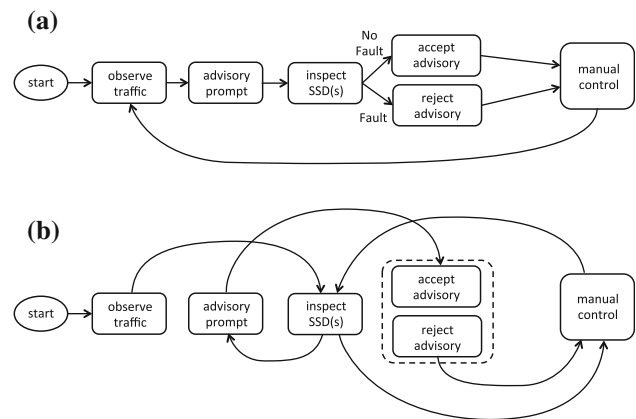


Fig. 18 Qualitative illustrations of observed control strategies. **a** Expected (nominal) behavior, **b** unexpected (observed) behavior

depiction of two main control strategies is provided in Fig. 18 as flow maps. The nominal strategy illustrated in Fig. 18a was the anticipated/designed strategy for failure detection and diagnosis. However, only six participants followed this strategy (i.e., P3, P5, P6, P7, P11 and P12). This observation is also reflected in the advisory response times (Fig. 12), the SSD inspections (Fig. 13a) and the number of control actions (Fig. 17).

All other participants (i.e., the majority within the means-ends ‘On’ group) followed a more complicated strategy, which was unexpected given the instructions and the limitations of the traffic simulator. In its most succinct form, this strategy was geared toward gaining manual control over the traffic scenario as much as possible. That is, as soon as a simulator trial commenced, participants immediately began to inspect all SSDs and used the means-ends linking when this was available, to proactively scan for potential problems and solutions. Then, the accept/reject buttons were used as gateways to gain manual control over aircraft and work around the automation’s advice. Here, the more convenient positioning of the ‘accept’ button above the ‘reject’ button (see Fig. 8) may have been responsible for causing a bias toward accepting advice.

The ‘quickly-gaining-manual-control’ strategy also led to some frustration among participants in the means-ends group. The designed limitation of the simulator only allowed interaction with aircraft that received an advisory, whereas some participants ideally wanted to solve a conflict by interacting with another aircraft. Here, the explicit means-ends links could highlight other, and perhaps more convenient, aircraft to interact with. In that sense, the limitations of the simulator did not allow participants to fully exploit their plans. Instead, they had to fall back to a back-up strategy that involved temporarily vectoring the aircraft under manual control into conflict zones of (far-away) aircraft that could not be controlled. This back-up

strategy often required follow-up control actions and speed clearances to solve new conflicts that they deliberately created, which explains the higher number of manual control actions than was minimally required.

6 Discussion

The goal of this research was to investigate the impact of explicit means-ends relationships on fault diagnosis of automated advice. Here, the focus was on diagnosing rare failure events in a supervisory air traffic control setting. Guided by previous studies, it was reasonable to assume that explicit means-ends links would make the fault detection and supervisory control tasks more efficient and effective.

The results revealed that the fault detection performance, as established through verbal comments, was indeed significantly (and positively) affected by the explicit means-ends links in the high complexity scenario. The majority of other measurements, especially the interaction with the advisory system, were either inconclusive (due to a lack of statistical significance) or ran counter to the hypotheses. These results were mainly caused by unexpected, but interesting interaction patterns of the participants with both the advisory system and ecological interface. These patterns appeared to be unrelated to the fault diagnosis task and geared toward taking control over from automation. As a result, several participants in the means-ends group were more active than anticipated and exploited the means-ends links to work around the automation's limitations. In combination with the limited sample size, not many significant results were found due to the variability between participants, which caused spread in the data. Note that variability between participants is expected when using ecological interfaces, because such interfaces do not dictate any specific course of action. However, the level of variability that was observed was unexpected, given the instructions and the carefully designed limitations of the simulator. In hindsight, two factors may have contributed to these findings.

First of all, a supervisory control context, in which decision authority shifts toward a computer, is not always well appreciated by human operators (Bekier et al. 2012). This can lead to people rejecting any form of automated decision support. Additionally, introducing a higher level of automation into socio-technical work domains generally involves making difficult and intertwined trade-offs and decisions to find the right balance between human and machine authority and autonomy (e.g., Dekker and Woods 1999). Despite the carefully designed experiment, it can thus be argued that still too much control authority was allocated to the participants, making it difficult to study the

phenomenon of interest. Note that the outcomes of ATC experiments are generally very difficult to control, as any decision made by a participant can affect the evolution of traffic situations in unexpected ways.

Second, all advisories were scripted and thus the same for all participants. Although this was deemed necessary for the sake of experimental control, research has also indicated that acceptance problems may arise when computer advice does not match the operator's way of working (Westin et al. 2016). As evidenced by the manual control performance, participants in both groups did seem to prefer different types of clearances. To mitigate fighting against the automation, it may be better to provide advisories in line with the operator's preferences. This, however, would be difficult to tune as this concept hinges on how consistent each person reacts to the same situation.

The observed strategy of 'fighting against the automation' cannot solely be attributed to the chosen level of automation and the implementation of the scripted advisories, however. This strategy has to be considered in light of the decision aid that was used, i.e., the SSD. This interface is intended to give participants more insight into the traffic situation and enable them to see more solutions than just the one that was offered by the automation. In case of explicit means-ends relations, participants were able to elicit more ways to solve problems, leading them to implement (for example) more speed clearances than the group without the explicit means-ends links.

Our experiment has exposed a potential dilemma regarding the use of ecological interfaces in highly automated control environments. On the one hand, constraint-based interfaces could facilitate automation transparency (Borst et al. 2015), allowing operators to better judge the validity and quality of specific computer advice. On the other hand, ecological interfaces reveal all feasible control actions within the work domain constraints, thereby increasing the chance that people will disagree with advice that wants to push them into one specific direction. Therefore, finding the right balance between offering more insight (e.g., through ecological interfaces) and striving for compliance with single (machine) advice is an avenue worth exploring further.

7 Conclusion

This paper presented the empirical investigation of explicit means-ends relations, in an ecological interface, on fault diagnosis of automated advice in a supervisory air traffic control task. Although a significant improvement in fault detection and diagnosis was indeed observed in a high complexity scenario, the experiment also exposed unexpected results regarding the participants' interactions with

the advisory system and ecological interface. The explicit means-ends links appeared to have mainly affected participants' control strategy, which was geared toward taking over control from automation, regardless of the fault condition. A plausible explanation is that the explicit relations expanded the participants' view on the traffic situations and allowed them to see more solutions than just the one that was offered by the advisory. This suggests that offering more insight, versus striving for compliance with single (machine) advice, is a delegate balance worth exploring further.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

- Ali B, Majumdar A, Ochieng WY, Schuster W (2013) ADS-B: the case for London terminal manoeuvring area (LTMA). In: Tenth USA/Europe air traffic management research and development seminar, pp 1–10
- Bekier M, Molesworth BR, Williamson A (2012) Tipping point: The narrow path between automation acceptance and rejection in air traffic management. *Saf Sci* 50(2):259–265. doi:[10.1016/j.ssci.2011.08.059](https://doi.org/10.1016/j.ssci.2011.08.059)
- Bennett KB, Flach JM (2011) Display and interface design: subtle science, exact art. CRC Press, Boca Raton
- Borst C, Flach JM, Ellerbroek J (2015) Beyond ecological interface design: lessons from concerns and misconceptions. *IEEE Trans Hum Mach Syst* 45(2):164–175. doi:[10.1109/THMS.2014.2364984](https://doi.org/10.1109/THMS.2014.2364984)
- Borst C, Mulder M, Van Paassen MM (2010) Design and simulator evaluation of an ecological synthetic vision display. *J Guid Control Dyn* 33(5):1577–1591. doi:[10.2514/1.47832](https://doi.org/10.2514/1.47832)
- Burns CM (2000) Putting it all together: improving display integration in ecological displays. *Hum Factors* 42(2):226–241
- Burns CM, Hajdukiewicz J (2004) Ecological interface design. CRC Press, Boca Raton
- Carr N (2014) The glass cage: automation and us, 1st edn. W. W. Norton & Company, New York
- Christoffersen K, Hunter CN, Vicente KJ (1998) A longitudinal study of the effects of ecological interface design on deep knowledge. *Int J Hum Comput Stud* 48(6):729–762
- Consortium S (2012) European ATM master plan. The roadmap for sustainable air traffic management, pp 1–100
- Dekker SWA, Woods DD (1999) To intervene or not to intervene: the dilemma of management by exception. *Cognit Technol Work* 1:86–96
- Ellerbroek J, Brantegem KCR, van Paassen MM, de Gelder N, Mulder M (2013) Experimental evaluation of a coplanar airborne separation display. *IEEE Trans Hum Mach Syst* 43(3):290–301. doi:[10.1109/TSMC.2013.2238925](https://doi.org/10.1109/TSMC.2013.2238925)
- Ham DH, Yoon WC (2001) Design of information content and layout for process control based on goal-means domain analysis. *Cognit Technol Work* 3:205–223
- Jones SR (2003) ADS-B surveillance quality indicators: their relationship to system operational capability and aircraft separation standards. *Air Traffic Control Q* 11(3):225–250. doi:[10.2514/atcq.11.3.225](https://doi.org/10.2514/atcq.11.3.225)
- McIlroy RC, Stanton NA (2015) Ecological interface design two decades on: whatever happened to the SRK taxonomy? *IEEE Trans Hum Mach Syst* 45(2):145–163. doi:[10.1109/THMS.2014.2369372](https://doi.org/10.1109/THMS.2014.2369372)
- Mercado Velasco GA, Borst C, Ellerbroek J, van Paassen MM, Mulder M (2015) The use of intent information in conflict detection and resolution models based on dynamic velocity obstacles. *IEEE Trans Intell Transp Syst* 16(4):2297–2302. doi:[10.1109/TITS.2014.2376031](https://doi.org/10.1109/TITS.2014.2376031)
- Parasuraman R, Sheridan TB, Wickens CD (2000) A model for types and levels of human interaction with automation. *IEEE Trans Syst Man Cybern Part A Syst Hum* 30(3):286–97
- Reising DVC, Sanderson PM (2004) Minimal instrumentation may compromise failure diagnosis with an ecological interface. *Hum Factors J Hum Factors Ergon Soc* 46(2):316–33
- Smith A, Cassell R (2006) Methods to provide system-wide ADS-B back-up, validation and security. In: 25th Digital avionics conference, pp 1–7
- St-Cyr O, Jamieson GA, Vicente KJ (2013) Ecological interface design and sensor noise. *Int J Hum Comput Stud* 71(11):1056–1068. doi:[10.1016/j.ijhcs.2013.08.005](https://doi.org/10.1016/j.ijhcs.2013.08.005)
- Van Dam SBJ, Mulder M, Van Paassen MM (2008) Ecological interface design of a tactical airborne separation assistance tool. *IEEE Trans Syst Man Cybern* 38(6):1221–1233
- Vicente KJ (1999) Cognitive work analysis; toward safe, productive, and healthy computer-based work. Lawrence Erlbaum Associates, Mahwah
- Vicente KJ (2002) Ecological interface design: progress and challenges. *Hum Factors J Hum Factors Ergon Soc* 44(1):62–78
- Vicente KJ, Moray N, Lee JD, Rasmussen J, Jones BG, Brock R, Djemil T (1996) Evaluation of a Rankine cycle display for nuclear power plant monitoring and diagnosis. *Hum Factors J Hum Factors Ergon Soc* 38(3):506–521. doi:[10.1518/001872096778702033](https://doi.org/10.1518/001872096778702033)
- Vicente KJ, Rasmussen J (1992) Ecological interface design: theoretical foundations. *IEEE Trans Syst Man Cybern* 22(4):1–2
- Westin CAL, Borst C, Hilburn BH (2016) Strategic conformance: overcoming acceptance issues of decision aiding automation? *IEEE Trans Hum Mach Syst* 46(1):41–52. doi:[10.1109/THMS.2015.2482480](https://doi.org/10.1109/THMS.2015.2482480)
- Zhang J, Liu W, Zhu Y (2011) Study of ADS-B data evaluation. *Chin J Aeronaut* 24(4):461–466