Design of an Ecological Interface for Flow-Based Perturbation Management for Future Air Traffic Control

Interaction with an automated trajectory planning algorithm

D.S.A. ten Brink

Technische Universiteit Delft

TU Delft

Design of an Ecological Interface for Flow-Based Perturbation Management for Future Air Traffic Control

Interaction with an automated trajectory planning algorithm

by

D.S.A. ten Brink

to obtain the degree of Master of Science at the Delft University of Technology, to be defended publicly on Wednesday October 18, 2017 at 14:00.

Student number:	4106873	
Project duration:	December, 2016 – October, 2017	
Thesis committee:	Dr. ir. C. Borst,	TU Delft, supervisor
	Prof. dr. ir. M. Mulder,	TU Delft
	Dr. ir. M.M. van Paassen,	TU Delft

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

TU Delft

Delft University of Technology

Copyright © D.S.A. ten Brink
All rights reserved.

DELFT UNIVERSITY OF TECHNOLOGY
DEPARTMENT OF
CONTROL AND OPERATIONS

The undersigned hereby certify that they have read and recommend to the Faculty of Aerospace Engineering for acceptance a thesis entitled **"Design of an Ecological Interface for Flow-Based Perturbation Management for Future Air Traffic Control"** by **D.S.A. ten Brink** in partial fulfillment of the requirements for the degree of **Master of Science**.

Dated: October 18, 2017

Readers:

prof.dr.ir. M. Mulder

dr.ir. M. M. van Paassen

dr.ir. C. Borst

dr.ir. J. C. F. de Winter

Abbreviations

AH	Abstraction Hierarchy
ASA	Abstraction-Sophistication Analysis
ATC	Air Traffic Control
ATCo	Air Traffic Controller
ATM	Air Traffic Management
CSE	Cognitive Systems Engineering
C-SHARE	Joint ATM Cognition through Shared Representation
EID	Ecological Interface Design
JCS	Joint Cognitive System
KBB	Knowledge-based behaviour
LoA	Level of Automation
PRM	Probabilistic Roadmap
RA	Restricted Airspace
RRT	Rapidly-exploring Random Tree
RBB	Rule-based behaviour
RTA	Required Time of Arrival
SBB	Skill-based behaviour
SRK	Skills, Rules & Knowledge
SUPEROPT	Supervision of Route Optimization
TSR	Travel Space Representation
WDA	Work Domain Analysis

List of Figures

Contents

Part I

Master of Science Thesis Paper

Design of an Ecological Interface for Flow-Based Perturbation Management for Future Air Traffic Control

D.S.A. ten Brink
Supervisors: C. Borst, M.M. van Paassen and M. Mulder

***Abstract*—Air traffic demand is expected to increase, while the system is already working at its limits. New decision support tools with higher levels of automation that allow for four-dimensional (4D, i.e., space and time) trajectory-based operations are proposed to cope with the increase in traffic demand. This research presents a conceptual interface that allows for flow-based perturbation management in air traffic, based on the principles of Ecological Interface Design and by means of influencing a trajectory planning algorithm. A first evaluation of this interface, in which five participants were asked to structure a perturbed airspace as they saw fit, showed that the participants were able to influence the algorithm as they wanted in most cases and were supported by the ecological interface that visualized the workings of the algorithm. However, human influence predominantly did not improve the solution in terms of robustness and efficiency. Improvements to the interface are suggested and its use case needs further research.**

Keywords*—Air Traffic Control, Ecological Interface Design, flow-based perturbation management, trajectory planning algorithm, human-machine interaction

I. Introduction

Due to the expected increase in air traffic coming years, the demand on the Air Traffic Management (ATM) system will increase, while it is already working at its limits. Because of this increase in demand it is expected that the current way of sector-based tactical control by Air Traffic Controllers (ATCos) will change to a more strategic form of airspace control [2]. This shift to a strategic form of airspace control is made possible by technological advances in both the air and ground sides of the ATM-system. New decision support tools with higher levels of automation will allow the ATCos to perform four-dimensional (4D, i.e., space and time) Trajectory-Based Operations (TBO) and adopt a strategic control strategy. With TBO, the ATCos will be able to plan the aircraft more closely together, due to the increased accuracy and thus generate a higher throughput. Although planned 4D trajectories are per definition de-conflicted before take-off, perturbations and deviations of the planned trajectory are inevitable [3]. Additionally, there is a general consensus within the operational communities that the human controller will need to keep a supervisory role within the system, due to the high stakes and the many unforeseen situations within the ATM work domain [2].

Thus the need for ATCos will stay and the increase in air traffic means that the task will only become more difficult, while ATCos are already operating at full capacity. Therefore, increased automation is required in the future to lower the workload of ATCos. More automation, however, often creates new problems, as other socio-technical domains have shown. Examples of such problems are: coordination breakdowns, skill degradation, complacency, transient workload peaks and vigilance problems [4].

An approach to mitigate these risks is based on the Ecological Interface Design (EID) paradigm [5]. The focus of EID is to support the human operator to conduct the work, instead of replacing the human with automated systems. As such, the goal of EID is to develop a functional model of the work domain that represents the complete set of functional constraints to the human operator, rather than providing the explicit solutions. By means of a shared human-automation representation the automation is made understandable and transparent to the human controller, which mitigates problems that higher levels of automation create. For example, in case the automation fails, the human controller needs to take over. If the automation works as a black box, this will be a challenging task, however when the human controller knows how the automation works this task's difficulty is considerably lowered.

The future TBO approach to ATC, with higher levels of automation, has been researched in previous work. A promising solution is the Travel Space Representation (TSR) designed by the SESAR WP-E project Joint ATM Cognition through Shared Representation (C-SHARE) [6]. The TSR is a constraint-based decision support tool, which is based on principles of EID. The tool visualizes the boundaries of safe control of an individual aircraft, which can be used for the task of short-term local trajectory revisions. While the TSR is a promising solution, each aircraft needs to be attended to individually by an ATCo in case of a perturbation, and thus the TSR does not have the possibility in its current form to considerably lower workload.

It will be investigated to further reduce workload by means of an ecological interface that allows for management of airspace flows. In this manner the airspace is controlled at a larger time scale, as compared to control of individual aircraft missions. This, however, does mean that more automation is

The authors are with the Control and Simulation Section, Faculty of Aerospace Engineering, Delft University of Technology 2629 HS Delft, The Netherlands (email: douwetenbrink@gmail.com; c.borst@tudelft.nl; m.m.vanpaassen@tudelft.nl, m.mulder@tudelft.nl).

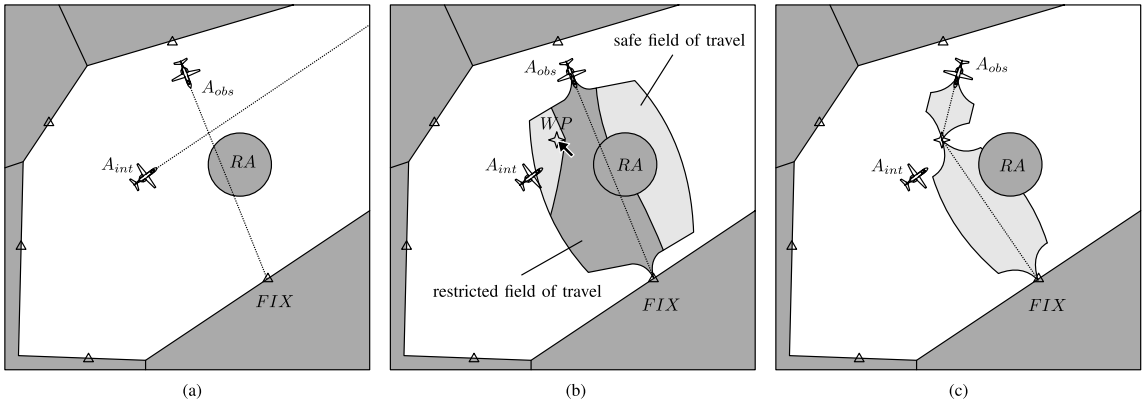


Fig. 1: TSR in an hypothetical traffic scenario. (a) Traffic scenario with two conflicting aircraft and a restricted airspace. (b) TSR and placement of an intermediate waypoint. (c) Resulting trajectory for the observed aircraft [1].

required. This will be in the form of a trajectory planning algorithm, which will cover the task of managing individual aircraft missions.

As argued by Borst et al. [7], with more automation it becomes more important to show more information about this automation. The goal of this is to ensure that both the human and the automation have a common ground that is a reflection of the control problem. The work domain analysis, which forms the basis of EID, is used to structure relationships and constraints within the work domain and as such find the common ground. The levels of control sophistication by Amelink [8], which concern levels of autonomy, will be used as an extension to the work domain analysis to incorporate the aspects of temporal scope.

Not only does this paper focus on making the workings of the automation visible, but also it focuses on how a human controller can influence the automation by means of constraining the automation in an ecological manner and as such working together as a team to control the airspace. Influencing automation is inspired by SUPEROPT [9], [10], an algorithm developed as part of SESAR WP-E. SUPEROPT allows the human controller to define how conflicts between aircraft are solved, for example one can tell the algorithm which aircraft will pass behind the other. The rationale behind influencing the automation is that it allows the human controller to structure the airspace in a manner that is logical and easy to understand to the human. Applying full automation could potentially lead to the human apparent chaos and black-box behavior. As such, advantages are that the airspace will be easier to monitor and to hand over the airspace to another ATCo.

This paper presents the design of a conceptual ecological interface for flow-based perturbation management by means of influencing a trajectory planning algorithm. First, the TSR is discussed, as this forms a basis and inspiration for this research. This is followed by an investigation of which trajectory planning algorithm is the most suitable for this particular interface and how this automation can be made visible and transparent to the human controller. Moreover, it will be investigated in which way the automation solutions can be influenced by the human controller, and if this would lead to a positive effect on the results of the automation. Next, a first concept design will be introduced, which is applicable during the pre-tactical management phase [2]. Results, discussion and conclusions will be presented from a first evaluation study.

II. Travel Space Representation

By means of the principles of EID the TSR is designed such that is visualizes the boundaries of safe control, by visualizing a set of constraints for safe and feasible control actions for a selected aircraft [6]. The general shape of the TSR is an ellipse and is determined by the aircraft's performance envelope, such as its speed and bank angle limits, while still realizing its required time of arrival (RTA) at the next waypoint [1]. Each ellipse within the TSR corresponds to a certain velocity of the aircraft, which realizes the RTA. On top of the shape the no-go areas are mapped, which result from additional constraints such as other traffic and restricted areas.

Fig. 1 shows the basic composition of the travel space representation in an hypothetical traffic scenario. In the scenario it is shown how a conflict resolution can be solved manually in three subsequent steps. The TSR can be directly manipulated with a mouse input device. In this situation the selected aircraft (A_{obs}) needs to be rerouted around the restricted airspace and intruding aircraft (A_{int}), while still maintaining the RTA at the waypoint *FIX*. This initial situation is shown in Fig. 1(a). By clicking on the aircraft (A_{obs}) the TSR is shown as illustrated in 1(b). The TSR shows a safe and restricted field of travel due to the intruding aircraft (A_{int}). With the mouse cursor a waypoint *WP* is placed within the safe field of travel. By pressing enter on the keyboard the flight path is selected and send to the aircraft. The modified flight path is shown in Fig. 1(c) and visualizes the new TSR for both segments. Waypoints could be placed outside the TSR, however these points will lead to not reaching the RTA and possibly lead to a conflict situation.

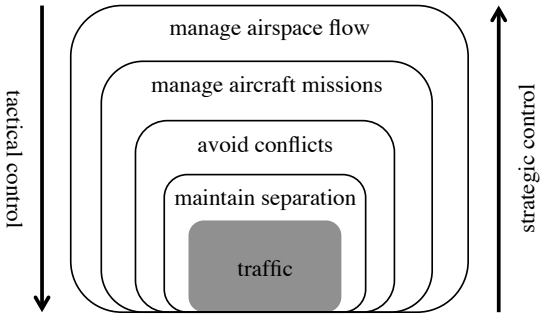


Fig. 2: Levels of control sophistication for ATC [11].

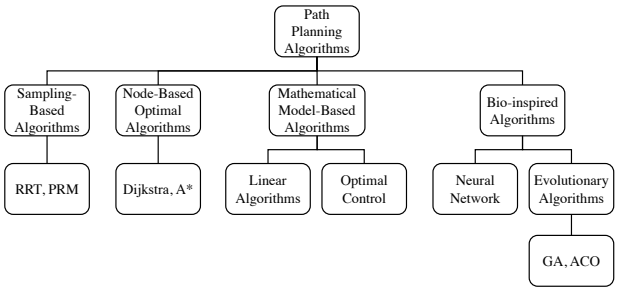


Fig. 3: Path planning algorithm taxonomy, adapted from [15].

III. Trajectory Planning Algorithm

The air traffic control problem can be represented as a system with several nested control loops, all acting at different time scales and with changing goals [11]. In Fig. 2 the control loops are shown, where the inner loops control the faster short-term dynamics and the outer the long-term applications. As discussed by Amelink [8] each loop represents a level of control sophistication, which relates to a level of autonomy. Thus, with increasing levels of control sophistication larger time spans are covered and automation is increased.

The TSR can be placed in the *manage aircraft missions* level, but to reduce the ATCo workload further and to control at another time scale, the next level of control sophistication, manage airspace flow, needs to be reached. To reach this level the human controller needs to be supported to close the *manage aircraft missions* nested control loop. A trajectory planning algorithm can be used to close this inner loop. This algorithm can take over the ATCo's role of rerouting and planning aircraft trajectories in case the airspace is perturbed.

Trajectory planning, or path planning when not associated with time [12], is the study of finding a sequence of actions that connect an initial state to a desired goal state. In path planning, the location the agents have are the states and the transition between states are the actions the agents can take, each with an associated cost. A path is optimal when it brings the agent from the initial state to the goal state with minimal cost. A path planning algorithm is considered *complete* when it always finds a path between the two states within finite time and notifies the user when no possible path is available [13].

For this research a trajectory planning algorithm is required which needs to be predictable and consistent in the solutions it provides [14]. Furthermore, the algorithm should be understandable to the human controller and computationally inexpensive.

Over the years many algorithms have been designed. These can be categorized into four main categories, all with their advantages and disadvantages: Sampling-Based Algorithms, Node-Based Optimal Algorithms, Mathematic Model-Based Algorithms, Bio-inspired Algorithms [15]. A fifth category is a fusion of combination of algorithms, but is not further considered in this research. An overview of the path planning taxonomy is found in Fig. 3.

Sampling-based algorithms are algorithms that have their basis in probability, in which they connect points sampled randomly through the space. The sampling-based algorithms are not considered complete, because of the randomness, but they provide probabilistic completeness guarantees. This is because the probability that the algorithm fails to provide a solution (if one exists) goes to zero as the number of samples approaches infinity [16].

Node-based optimal algorithms explore the space in a discretized manner. These kind of algorithms explore a set of nodes in the space and find the optimal path by calculating the cost while running through these placed nodes. The most well-known shortest path finding algorithm is Dijkstra's algorithm [17]. Many variants of Dijkstra's algorithm have been constructed, for example A* [18], [19]. Each of the node-based algorithms is heavily influenced by its settings. An increase in the step size or nodes affects the computation time and the optimality of the solution. In contrast to sampling-based algorithms, node-based algorithms provide the same solution each time the algorithm is run.

Mathematical model-based algorithms are algorithms that model the environment and the system. The environment is modelled as kinematic constraints and the system as dynamic constraints. To find an optimal solution to the path planning problem, these constraints are added to a cost function as inequalities or equations. The problem with these algorithms is that they have a complex formulation, because the whole environment and system needs to be modelled, they tend to be computationally expensive. The mathematical model-based algorithms can be mainly categorized into two categories: Linear Algorithms and Optimal Control [15].

Bio-inspired algorithms are based on the idea of mimicking nature to find an optimal solution. These algorithms have the ability to solve problems with many variables and non-linearity, something which his hard to do with mathematical model-based algorithms. Within the field of bio-inspired algorithms there are two categories: Evolutionary Algorithms and Neural Networks.

Because of this predictability constraint the sampling-based algorithms can directly be considered as infeasible. Sampling-based algorithms, as the name suggests, are highly random

and provide a different solution each time. Also bio-inspired algorithms can be considered as infeasible, due to their stochastic properties. Mathematical model-based algorithms can be considered infeasible for this research due to their complexity and computational expense. This leaves us with node-based algorithms.

As such, we decided to implement the Trajectory Flexibility Metric algorithm by Idris [20]–[22]. The approach relates to a node-based optimal algorithm and discretizes the space in position and time to account for dynamic objects. The idea behind this algorithm is that by implementing trajectory flexibility on an individual aircraft, the traffic complexity of the whole airspace can be maintained on acceptable levels. As the algorithm is node-based, the algorithm is predictable since it follows a fixed set of rules.

Furthermore, the algorithm is constructed such that it is not only possible to optimize for the shortest path, but also for the metrics robustness and adaptability, which are defined by Idris as follows:

Robustness: *"the ability of the aircraft to keep its planned trajectory unchanged in response to the occurrence of disturbances, for example, no matter which trajectory or conflict instances materialize." [21]*

Adaptability: *"the ability of the aircraft to change its planned trajectory in response to the occurrence of a disturbance that renders the current planned trajectory infeasible." [21]*

Preliminary research showed that a cost function including robustness or adaptability only became useful when rerouting trajectories starting with six to seven additional waypoints, whereas in ATC the ATCos usually create new trajectories with only one or maybe two additional waypoints. For the scope of this research the number of additional waypoints should stay limited, such that the trajectories planned by the algorithm are understandable to the human controller, therefore it was chosen to not consider robustness or adaptability in the cost function and only focus on the shortest path. Also, it was expected that by influencing the algorithm in combination with the shortest path as a cost, the sector robustness of the airspace could also be increased. Where sector robustness is defined as an average over the robustness of each aircraft [1]. To increase the sector robustness, for example, the human controller could tell the algorithm to find trajectories with a larger buffer around the perturbation. Another option the controller could opt for is to guide two traffic streams further away from each other and as such increase the sector robustness.

Due to the discretized nature of the algorithm, the positions where new waypoints can be placed, and thus the possible trajectories that can be found are constrained. With the TSR, however, the human controller is free where to place the waypoints. Fig. 4 shows the TSR, in which a trajectory has been chosen by the human controller with one additional waypoint around a perturbation. Also shown are all feasible points the algorithm can utilize to place an waypoint. Fig. 5 shows the same illustration for two additional waypoints.

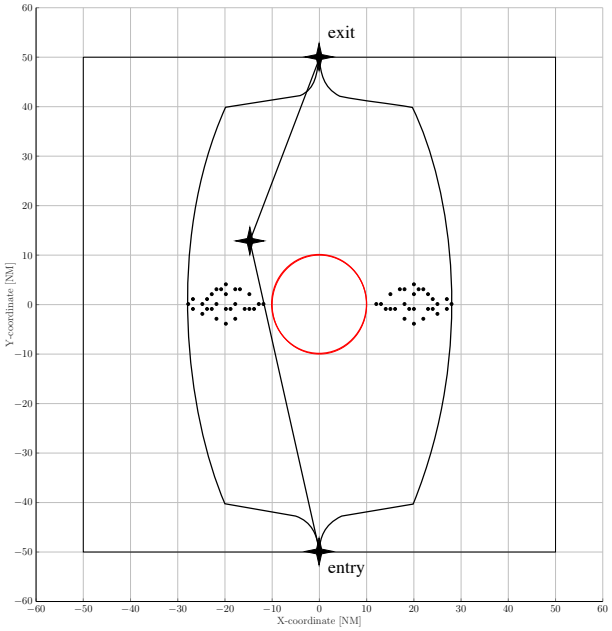


Fig. 4: Feasible points trajectory planning algorithm for one additional waypoints overlapped with a chosen trajectory by utilizing the TSR.

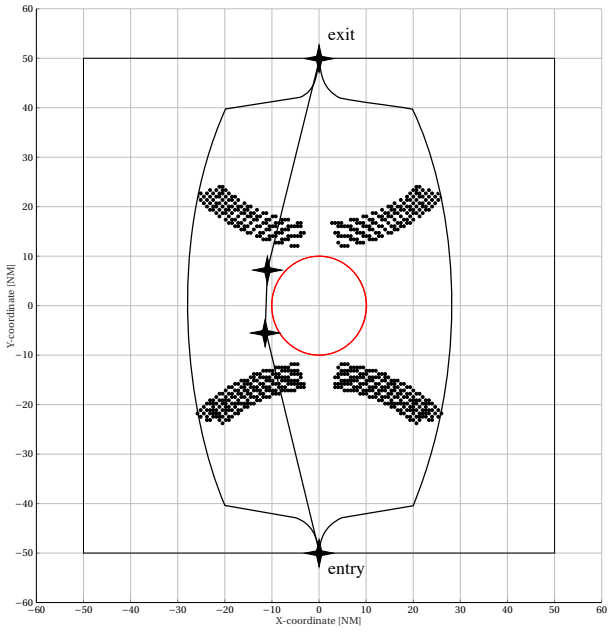


Fig. 5: Feasible points trajectory planning algorithm for two additional waypoints overlapped with a chosen trajectory by utilizing the TSR.

IV. Ecological Interface Design

A cognitive work analysis is the basis for an ecological interface. The process of a cognitive work analysis starts with a functional breakdown of the work domain, which reveals the functions and relations (means-ends links) on various levels of abstraction. This is followed by a control task analysis, which in this paper is combined with the cooperation analysis to analyze which tasks are performed by each actor. Next, a strategies analysis is performed, which shows several strategies to perform the control task. Finally, a worker competencies analysis is performed to determine how the human actors are supported in their task.

The system's boundaries in which the work domain analysis will be performed is limited to the perturbation management task for multiple aircraft at a fixed altitude. The task is performed in a strategic manner, prior to the aircraft entering the perturbed airspace, during the pre-tactical management phase [2]. This phase takes place several hours up to 30 minutes before current time. The functional purpose of the system is to do this in a productive, safe and efficient manner.

A. Work Domain Analysis

The work domain is analyzed by means of an abstraction hierarchy (AH). An AH was defined by Van Paassen et al. [6] for the TSR at a single aircraft level, however for this research the AH needs to be extended to the whole airspace in which the airspace flow management will take place. Also accounted for, is the different view on the work domain by the two stakeholders; the human controller and the agent, in this case the trajectory planning algorithm. Due to the fact that the algorithm is discrete and has its own limits and constraints, the view of the algorithm on the work domain is different than the human's view, although still partly overlapping. Therefore, as discussed by Naiker [23], the AH is split into two object worlds, which is defined as the stakeholder's view on the work domain. The object worlds are the physical domain and the agent domain, for the human controller and trajectory planning algorithm, respectively. The result of the complete AH is shown in Fig. 6.

a) Functional purpose: For both airspace and single aircraft control and the physical and agent domain the goals of this system are to produce 4D trajectory definitions in a safe and efficient manner.

b) Abstract function: For the physical domain locomotion, separation and economy are defined at the abstract function level to meet the system goals' functions, which are equal for both airspace and single aircraft levels of decomposition. Locomotion is realized within internal aircraft constraints and external constraints imposed by the airspace, such as airspace regulations. Separation is the main method for safety control in the system. Safety is reached when at all times enough separation is maintained between aircraft, terrain and hazardous weather. Economy is the main driver to be efficient, i.e., finding short and economical trajectories to the destination. In the agent domain the abstract function is defined by trajectory planning, which is a means to meet the system goals.

c) Generalized function: At this level of abstraction the two levels of control sophistication are represented in the physical domain: flow management and aircraft mission. Flow management is defined by the flows and streams in an airspace, the performance envelopes of aircraft in such a stream and the obstacles that can be present. Aircraft mission is defined by the flight plan of an aircraft, its performance and also the same obstacles that can be present in traffic streams. At the agent domain the two main tasks that are required for trajectory planning are represented: to minimize the cost function and to de-conflict traffic. These are required for trajectory planning as by minimizing the cost function the best trajectory is found, and with de-conflicting the traffic only the possible trajectories are determined.

d) Physical function: In the physical domain the physical function is only described at the single aircraft level of control sophistication, as multiple aircraft missions make up for the flow management level. The flight plan can be represented as a series of waypoints, the performance as a speed and heading envelope and the obstacles as traffic or weather. The physical functions in the agent domain are functions that are used to minimize the cost function and to de-conflict the traffic. Where track miles are used to determine the shortest path. All other functions, such as time constraints, conflict-free trajectories, air traffic and prohibited airspace contribute to de-conflicting the traffic.

e) Physical form: This level of abstraction describes the appearance of the functions described at the physical form level. At the physical domain for the series of waypoints these consist of locations and their RTA. The speed/heading envelope is represented by a minimum and maximum speed/heading, the traffic by other aircraft states and their intent and finally, weather by its location, size and shape. At the agent domain, the track miles are determined by the trajectory length and RTAs make up for the time constraints. The complete search grid dimension to determine conflict-free trajectories is defined by the grid size, number of additional waypoints and heading and speed strategy. Air traffic is defined by the 4D (i.e., space and time) locations of the aircraft, their planned trajectories and the protected zone (PZ) size of the aircraft. Finally, a prohibited airspace is defined by its location, size and shape.

B. Cooperation and Control Task Analysis

For the task of flow-based perturbation management in this interface there will be two actors: the human controller and the automation. Each actor must be allocated subtasks, that together make up the complete task. A decision ladder, for each actor, is used to represent the actions and subtasks and the relations between them, as shown in Fig. 8.

For this task, during the pre-tactical management phase the decision process activates when an event perturbs the airspace. The plan-view of the airspace, including the solution spaces for each airway, are used as observation to become aware of the traffic patterns, perturbations and performance envelopes. The solution spaces show the constraints of the algorithm, which are in return based on the performance envelopes of aircraft present in that specific airway. This can be used to

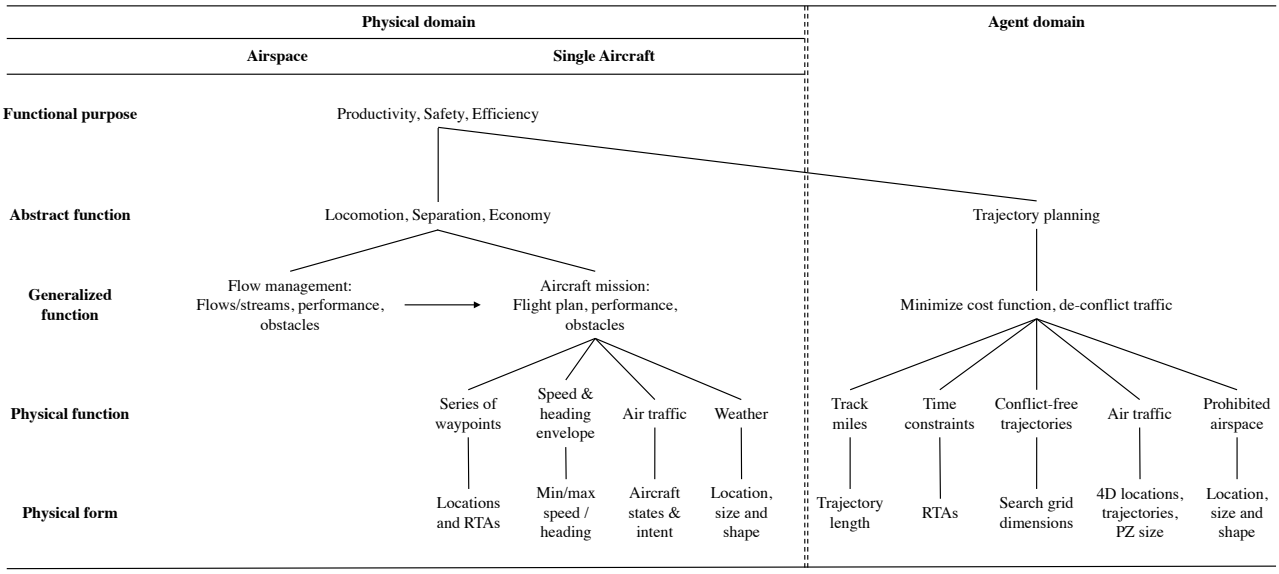


Fig. 6: Abstraction hierarchy flow-based perturbation management for two stakeholders.

determine the current state of the system. When the system state is known, the different solution methods of de-conflicting the air traffic around the perturbation can be evaluated. This is done by taking into account the mission goals of productivity, safety and efficiency.

Once a solution method has been selected, for example to guide traffic with a 5 NM buffer around the perturbation, the goal state of the airspace is known. This goal state is reached by defining the task. The traffic flows can be steered in a certain direction by flow obstacles or constraints, which influence the solution space of the algorithm. A procedural shortcut can be followed, in which the choice of influence on the algorithm is immediately defined in the task. Once the human controller is satisfied by the tasks, the scenario will be executed.

Another shortcut can be followed from the set of observations directly to the task definition. This shortcut can be utilized when a controller recognizes a certain scenario, which he or she solved before in a certain manner that gave a positive output. In this case it is thus not necessary to go through the whole knowledge-based domain again, and the task can be directly implemented in a rule-based manner.

The execution phase of the human controller activates the decision process of the automation. The automation will observe all traffic and constraints drawn in the airspace, from which it will determine all possible trajectories. From these trajectories the automation will filter the feasible ones and select those with the best cost function score; in this case the trajectories with the shortest path. The automation will update the trajectory, set the waypoints and as such execute the task. This process is repeated for each aircraft entering the airspace.

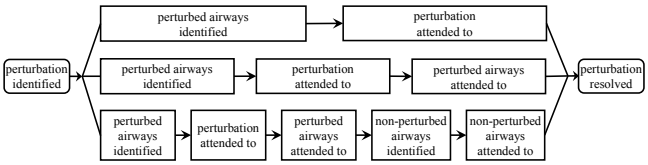


Fig. 7: Information flow-map of three control strategies.

C. Strategies Analysis

The control problem is to resolve perturbations in the airspace, once they are identified. Three different strategies to resolve the perturbation in a flow-based manner are considered, which are summarized in Fig. 7. With the first strategy the perturbed airways are identified, after which the decision is made to only attend to the perturbation itself. This is done by means of determining a prohibited airspace around the perturbation, with possibly a certain buffer, which acts as a constraint to the automated agent. A second strategy involves, in addition to the first strategy, attending to the perturbed airways. For example, the controller can decide to force the agent to find new trajectories only left of the perturbation or to make the trajectory longer one side around the perturbation which creates an extra buffer zone. With the final strategy, the airways that are not perturbed are also identified and attended to by the controller. Possibly, to reroute an airway, such that a perturbed airway has more space to be rerouted around the perturbation.

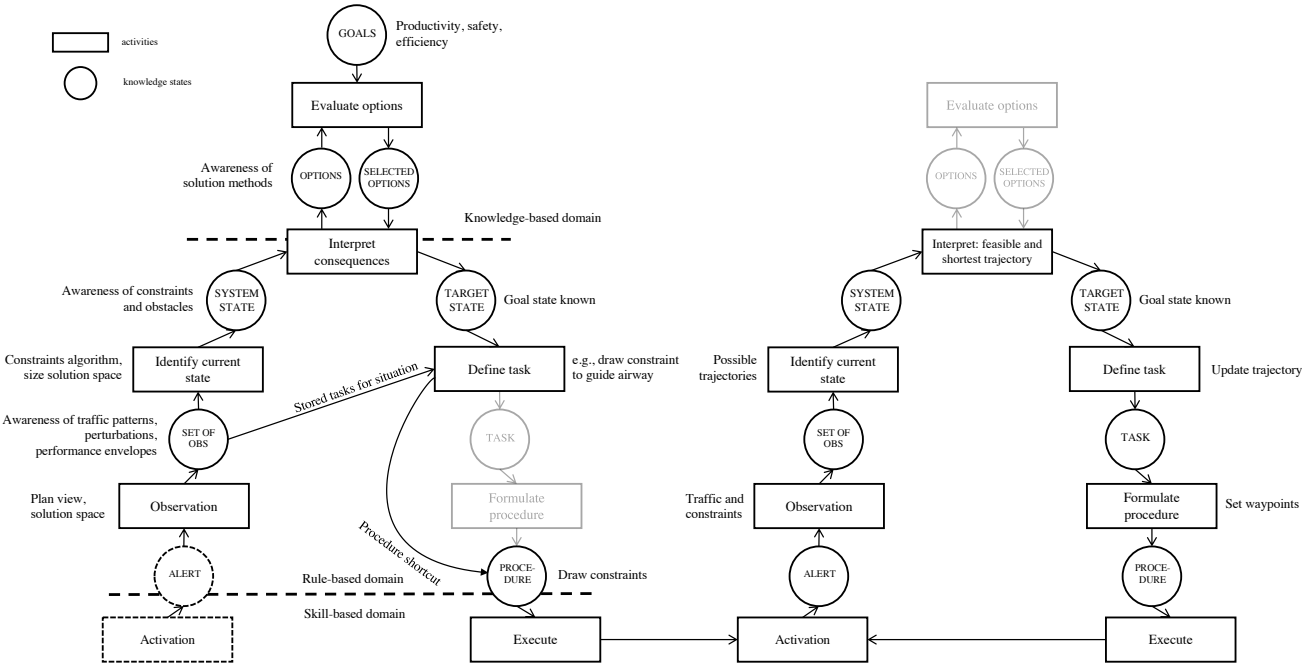


Fig. 8: Decision ladder of the flow-based perturbation management task, separated for the human controller (left) and automation (right).

D. Worker Competencies Analysis

The worker competencies analysis is used to assess the level of cognitive behaviour that is needed to perform the tasks that have been allocated to the controller. Rasmussen's skills, rule, knowledge taxonomy [24] is used as a qualitative framework to explain the hypothetical benefits of the interface. The behavior domains are indicated in Fig. 8.

a) Skill-based behavior: The activation and executing of procedures can be considered as skill-based behavior. These tasks do not require conscious control by the human operator.

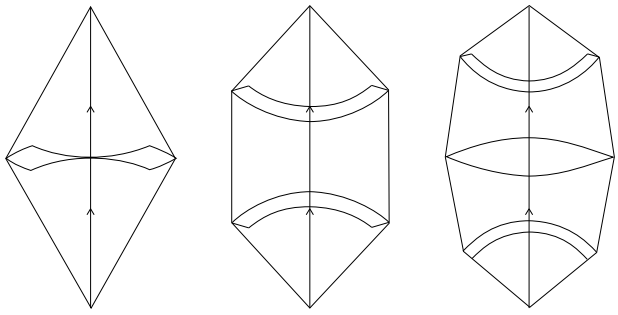
b) Rule-based behavior: Observation of the airspace and being aware of the traffic patterns and airspace situation can be considered as rule-based behavior. For example, as the controller observes that an airway crosses a perturbation, (s)he knows an action is required to reroute the airway around the perturbation. The information gained in the rule-based domain, supports decision making in the knowledge-based domain. The stored task shortcut can be utilized to skip the knowledge-based domain. The risk of utilizing this shortcut is that it might lead to poor control performance and less knowledge about what is going on in the system.

c) Knowledge-based behavior: Reasoning upon the possible solution methods in the knowledge-based domain is supported by the observations of the representation of the solution space and awareness of constraints. The controller can weigh and evaluate each option, while taking into account the mission's goals of productivity, safety and efficiency. Also, in case of unexpected events, such as an automation failure, the behavior can be seen as knowledge-based to solve the problem.

E. Interface Design

A prototype interface has been designed for the task of rerouting 3D (i.e., 2D space and time) trajectories at a fixed altitude in a flow-based manner during the pre-tactical monitoring phase. The trajectories will be rerouted by the automation, which the human controller can influence by constraining the automation. By constraining the automation one can point out several no-go areas for the automation. For instance, one could choose to force the automation to only find solutions at one side or to reroute around a perturbation, with possibly a certain buffer zone to increase the robustness of the airspace.

To support the human controller in the task of constraining the automation, the work domain constraints are made visible by means of the airway solution space. This allows the human controller to reason about the placement of the constraints and the manner in which the controller would like to guide aircraft in need of a trajectory revision. The airway solution space is based on the TSR, which visualizes the work domain constraints of the aircraft. However, the automation also has its own constraints due to the discretized nature of the algorithm, as was shown in the AH in Fig. 6. These constraints are visualized within the airway solution space. The airway solution space is dependent on a number of variables, which are: the number of waypoints, base speed, maximum speed, minimum and maximum heading and RTA at the exit waypoint.



(a) 1 (b) 2 (c) 3

Fig. 9: Solution space for one, two and three additional waypoints.

Fig. 9 shows three airway solution spaces for one, two and three additional waypoints. The outer boundaries of the shapes show the speed limits of the aircraft, within which it can still reach the exit point at the required time of arrival, as with the TSR. The curved areas show the possible locations in which a waypoint can be placed by the automation. When referring to Figs. 4 and 5 one can see the similarities with Figs. 9a and 9b, which inspired the design of the airway solution space. The direction of the airway is indicated by the two arrows on the center line. As multiple types of aircraft can fly in an airway, the airway solution space is based on the aircraft with the smallest performance envelope.

Fig. 10 provides an overview of the interface and the possible control actions a human controller can perform while working with the interface for a simple scenario. The numbers in the figure indicate parts of the work domain that are made visible in the interface. Fig. 10a is what will be presented to the human controller first when asked to structure the airspace to his or her preference. The scenario shown consists of two airways (1) with a perturbation (2) shown in the center.

To support the controller in this task, the airway solution space (3) (4) can be shown for a specific airway when hovering over this airway, as shown in Fig. 10b. This gives the controller an indication what the limits are of the algorithm and aircraft in that specific airway. For this research the number of additional waypoints is fixed to two, so if a rerouting is required, the automation will place two waypoints, one in each curved area.

To gain more information, the controller can select multiple airways and make the overlapping airway solution spaces visible (3) (4), as shown in Fig. 10c.

Once the controller knows how to structure the airspace, he or she can start placing constraints, which will guide the algorithm. In Fig. 10d it is shown how a circle constraint (5) is placed around the perturbation. The circle constraint acts as prohibited airspace for the algorithm. The constraint is drawn slightly larger than the perturbation to implement a buffer zone and as such increase the sector robustness.

In Fig. 10e an additional polygon constraint (5) is placed, which guides the traffic West of the perturbation, so as to steer traffic away from the other airway and to prevent a possible knock-on effect with this airway. This constraint needs to be placed, as without it the algorithm could choose either side of the perturbation to plan a trajectory as the trajectory length around both sides is equal. In Fig. 10g one aircraft is about to enter the airspace from the North and one just entered from the South (6). The aircraft from the South is highlighted (7), as the trajectory planning algorithm is activated to reroute the aircraft around the perturbation, taking into account the drawn constraints. The current trajectory of the aircraft is shown with the line from WP5 to WP6 (8). Finally, in Fig. 10f the result of the trajectory planning algorithm is shown (8) (9). The aircraft are safely separated from the perturbation (10) and without the airways affecting each other (11).

V. Conceptual Evaluation

In this early design phase a conceptual evaluation of the flow-based perturbation management interface has been performed. The evaluation is used to validate the underlying principles of the interface. Next to this, the data and feedback generated from the participants are used to find new valuable insights and improvements to the interface, which will guide the direction of future development.

A. Participants

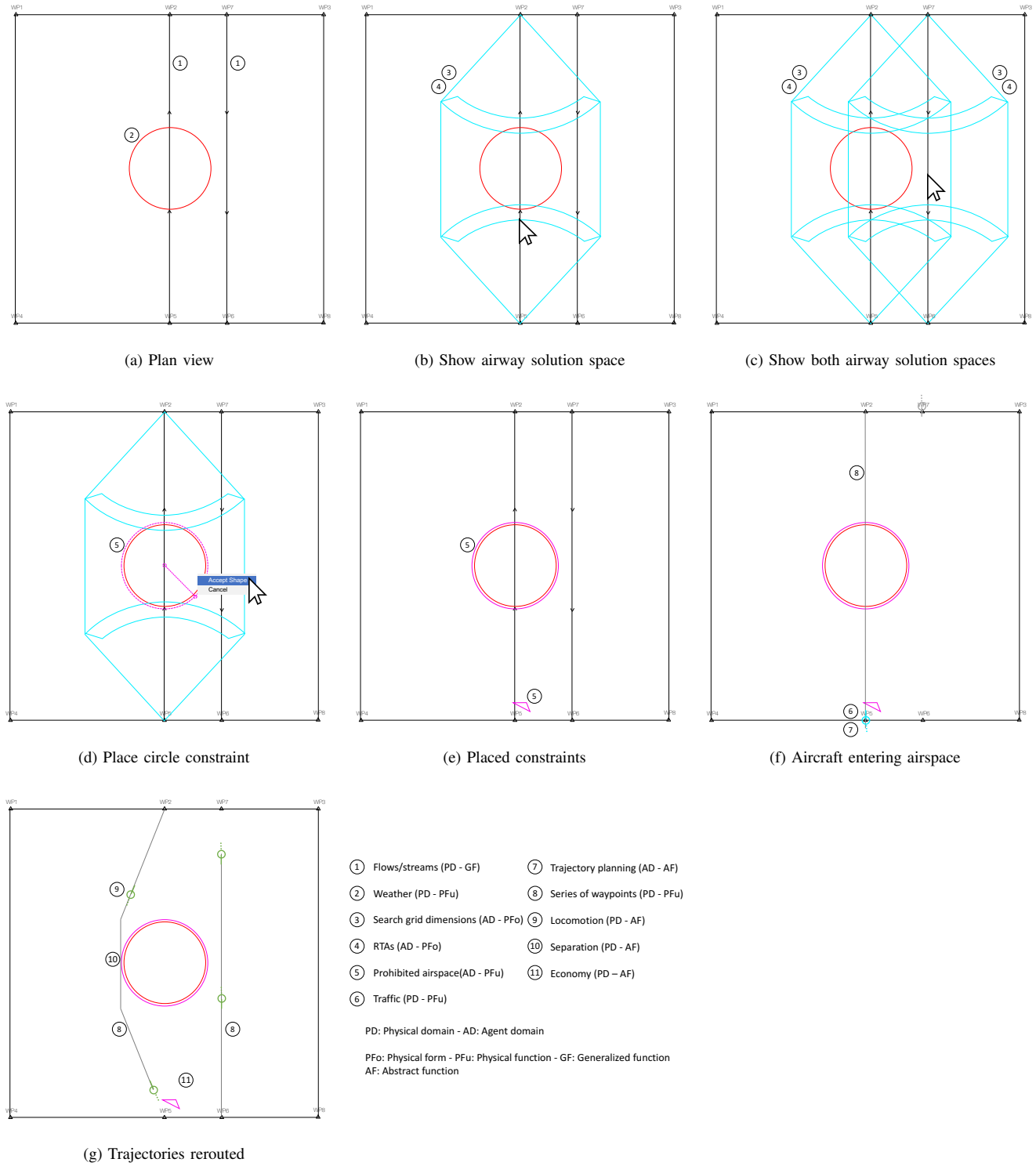
In total five participants engaged in the evaluation, who all followed a short ATC course, intended to familiarize researchers in ATC with hands on ATC practice. Each participant was male, ranging between 25 and 55 years of age. Four participants rated their ATM knowledge as intermediate, one as advanced. Regarding 4D operations, two participants considered their knowledge basic, two intermediate and one advanced. Finally, for human machine interface design one considered his knowledge as intermediate, the others as advanced.

B. Apparatus

The evaluation was performed on a dedicated software-based ATM platform in the ATM laboratory of the faculty of Aerospace Engineering at the Delft University of Technology. The interface was integrated in the TSR software, which implies a traditional plan-view display, providing a top-down view of the airspace and traffic. The interface was presented on a 30-inch screen (60-Hz LED, 2560 by 1600 pixels) placed in front of the participant. Input was given by a standard mouse input device and control options could be selected by on-screen buttons and drop-down menus.

C. Tasks and Instructions

Based on the TSR software platform [1], a prototype of the interface was programmed, which was used for the conceptual evaluation. For the evaluation, the participants were assigned



(a) Plan view

(b) Show airway solution space

(c) Show both airway solution spaces

(d) Place circle constraint

(e) Placed constraints

(f) Aircraft entering airspace

(g) Trajectories rerouted

Fig. 10: Multiple steps in the process of structuring a simple airspace with the interface, included with references to the Abstraction Hierarchy.

the task of an ATCo during the pre-tactical management phase, where they were asked to de-conflict the traffic that was disrupted by the perturbation. The scenarios were revealed to the participants in a plan-view display showing the sector borders, perturbation and airways. The participants had no information available over the upcoming traffic density. When hovering over or selecting an airway, the solution space of the algorithm for that specific airway was shown. With this information the participants were asked to guide the traffic around the perturbation and to structure the airspace in a fashion they deemed necessary by placing constraints, while considering and weighing the following interdependent criteria: safety buffers, additional trajectory length and airspace structure. Once the participant was satisfied with the applied structure, the simulation was started and fast forwarded to show the results of the trajectory planning algorithm that rerouted the flights upon entering the airspace, while taking into account the placed constraints. The reason why it was chosen not to allow control while aircraft were in the airspace, is because we wanted to prevent for this research that participants would apply a strategy of controlling each aircraft individually. This strategy is possible by a repeating process of placing and deleting constraints for each aircraft.

D. Baseline Performance

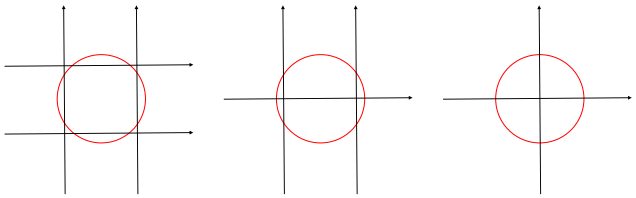
To assess the performance of the human controllers two baseline measures were used. One is the result of the full automation, thus without any human influence. The other baseline is set with the TSR interface by an expert controller. Comparison with the TSR should be taken carefully, as with the TSR the controller is rerouting trajectories throughout the scenario, whereas with the concept interface the controller sets his or her constraints at the start of the scenario without any knowledge of what will happen in the future and he does not have to ability to adjust trajectories throughout the scenario.

E. Independent Variables

For this concept evaluation two independent variables were considered, traffic density and perturbation location as within-subjects variables:

1) *Traffic density:* the traffic density with two levels - low (TDL) and high (TDH);
2) *Perturbation location:* the location of the perturbation - number of airways through center of perturbation: zero (PL0), one (PL1) and two (PL2).

The rationale for choosing two levels of traffic density is that it will be interesting to compare current traffic densities with expected future densities and to find out how the concept performs in these two situations. The perturbation location is chosen as an independent variable because preliminary research showed that the location highly influences the applied control strategies. For this research the locations are grouped into three categories: PL0, PL1 and PL2. The locations were chosen such that either 0, 1 or 2 airways crossed right through the center of the perturbation, see Fig. 11. The rationale behind this is that if an airway crosses the perturbation right through the center, the human controller can make a decision to steer the trajectory algorithm left or right of the perturbation. If an airway crosses the perturbation only slightly, as with PL0 and PL1, the trajectory algorithm will always choose the shortest path around the perturbation. It might only pass the other side of the perturbation if the algorithm cannot find a solution the shortest way around the perturbation due to air traffic. So although for PL0 four airways are affected, and for PL2 only two, the influence a human controller can have on the algorithm is much larger.



(a) PL0 (b) PL1 (c) PL2

Fig. 11: Three categories of perturbation locations as defined for this research.

F. Traffic Scenarios

To effectively assess the performance of the interface a conceptual airspace was designed, based on the multi-sector principle [25]. Three upper airspace sectors from the south-eastern part of France where combined into one large airspace of $92,000 km^2$. In this airspace eight hypothetical airways were placed in a structured grid-based manner. In addition, a high traffic density and a low traffic density scenario were defined.

The low traffic density scenario was based on the peak traffic that Eurocontrol currently sees in its airspace during summer: 5,500 flight movements during 24 hours in an airspace of $260,000 km^2$ starting from FL245 and up [26]. Mapping this onto the designed airspace and taking into account that only the horizontal dimension is used for the scope of this interface concept, the low density traffic scenario was set to 800 flights ($\sim$33 flights/hour). The high density scenario was scaled with a factor of 1.6, based on future traffic predictions [27]. This led to a high density traffic scenario of 1,280 flights ($\sim$53 flights/hours).

Both traffic scenarios were made conflict-free beforehand, based on a minimum separation distance of 5NM. All flights were set to be of the type Airbus A320 flying at FL245, which is the lower bound of the upper airspace. This is also near the flight level which leads to the maximum performance envelope of an A320 aircraft (FL240). The two traffic scenarios were designed so that the last aircraft would enter around 45 minutes in scenario time. Thus, for the low traffic density scenario 25 aircraft would enter the airspace and 40 for the high scenario.

In both traffic scenarios conflicts were introduced, by placing a large circle-shaped perturbation of 25NM at three different locations. A total of 21 different locations were evaluated

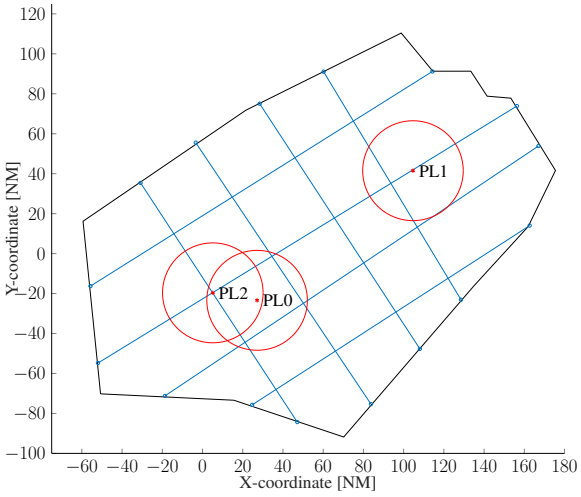


Fig. 12: Designed airspace including the selected perturbation locations.

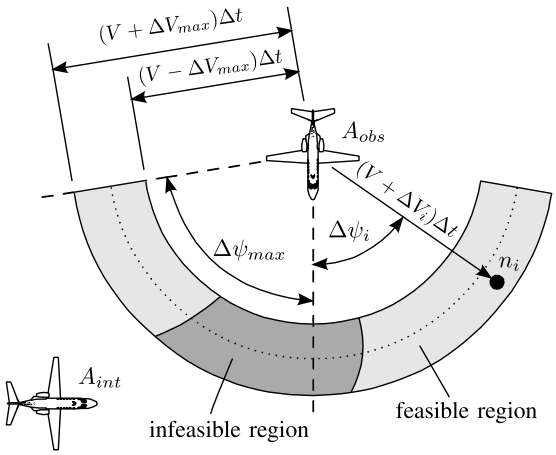


Fig. 13: Sketch of point-based robustness geometry and the resulting feasible and infeasible regions [1].

during preliminary research, split over three perturbation location categories as shown in Fig. 11. From every category one perturbation was selected that caused the highest disturbance in traffic. The airspace with the selected perturbation locations is shown in Fig. 12.

G. Dependent Measures

To compare the results of the participants to the automation and to investigate what effects control actions have on the algorithm, the following dependent measures are considered:

1) Constraints placement: the type, size and amount of constraints placed by the participants in the airspace.
2) Sector-based robustness: the minimum and average robustness off all aircraft trajectories [1]. In which the minimum is defined as the point of least robustness of all combined trajectories. Average sector robustness is defined as the average of all trajectories' robustness. The sector robustness is derived from the point-based robustness of a single aircraft at a certain point in time. At each point in time, the aircraft is assumed to be at a predicted state (t, x, y, V, ψ). To quantify the robustness in that point in time, RBT(t), the predicated state is taken as a starting point, at which the probability of feasibility of the aircraft to reach the set of predicted states at time $t + \Delta t$ is determined. Fig. 13 shows a sketch of an observed aircraft (A_{obs}) with the predicted positions the aircraft can be at at time $t+\Delta t$. The disc-shaped area is discretized into n_i probe segments, where for each probe segment the feasibility is determined, from which the robustness can be determined. Infeasible regions can occur by other aircraft predicted positions, as shown in Fig. 13, or by perturbation intrusions. The robustness will be calculated by taking both the aircraft and perturbation intrusions into account, but also by only taking into account the aircraft. This will allow to compare the strategies between the participants more fair, because it removes the bias of how large of a buffer the participant decided to take around the perturbation. As a larger buffer will directly mean an higher robustness due to a smaller infeasible region of aircraft flying past the perturbation. Both metrics are calculated by means of post-hoc calculations, where the maximum heading angle difference has been set to 80^o, discretized in steps of 5^o and the maximum speed difference to 20 kts IAS, in speed steps of 10 kts IAS. The look-a-head time was set to 120 s and the point-based robustness was sampled every 10 seconds.
3) Efficiency: additional track miles flown by the aircraft as compared to the de-conflicted non-perturbed airspace and the number of knock-on reroutes, which are defined as aircraft that need to deviate from their original trajectory due to conflicting with other aircraft and not due to the perturbation.
4) Situation awareness: after each run the participants are required to fill out a modified version of the SASHA questionnaire [28], measuring their situation awareness.
5) Trust between people and automation: after each run the participants are required to fill out a reduced version of the checklist for trust between people and automation [29], to measure their trust in the system.

H. Control Variables

To avoid confounds a set of variables has been kept constant throughout the evaluation. A summary of the variables, and their associated values, are found in Table I.

I. Procedure

Prior to the evaluation the participants were asked to fill out a questionnaire, rating their knowledge level of air traffic management, 4D operations and human machine interface

TABLE I: Concept evaluation control variables.

Control variables	**Value**
Airspace sector shape	-
Airspace sector size	$92,000[km^2]$
Airspace structure	-
Aircraft type	Airbus A320
Aircraft flight level	FL240
Aircraft instantaneous airspeed	400, 445 & 490 [kts IAS]
Aircraft heading limits	-50^o - 50^o, $\Delta 1^o$
Algorithm grid size	101 x 101 [-]
Algorithm state change strategy	relative
Algorithm number of intermediate waypoints	2 [-]

design. After the questionnaire and a short introduction to the concepts behind the interface, several training runs were performed by the participants to familiarize themselves with the interface and learn in which manners they were able to apply structure with the constraints. The participants were given a cheat sheet, which showed four different ways to steer the automation in a certain direction. In total there were six measured runs to be performed after the training. Three scenarios for each perturbation location, and three times the same scenarios rotated 90^o clockwise, to gather more data while preventing scenario recognition. The order in which the scenarios were presented to the participants was defined by the Latin square, as shown in Table II. The participants were only presented with an airspace without any knowledge about the traffic density, therefore the placed constraint set-up by the participants is applicable to both traffic density scenarios. Thus during the experiment the participants only saw the results of the high traffic density scenarios, but the experimenter was able to gather the low traffic scenario data by means of post-hoc runs. After each measured run the participants were asked to fill out a questionnaire. The questionnaire consisted out of three parts: Two open questions asking what they thought of the amount of structure they were able to add to the airspace and whether they would change something to the constraint set-up after seeing the result. The second part was an adapted version of the SASHA questionnaire [28] to measure their situation awareness. The final part was a shortened version of the Checklist for Trust between People and Automation [29]. A debrief followed after the evaluation, asking the participant to assess and provide feedback on the interface, automation and experiment. A complete session took around 1.5 hours.

TABLE II: Latin square experiment design.

ATCo	**Run 1**	**Run 2**	**Run 3**	**Run 4**	**Run 5**	**Run 6**
1	$PL0_{0^o}$	$PL1_{90^o}$	$PL2_{0^o}$	$PL0_{90^o}$	$PL1_{0^o}$	$PL2_{90^o}$
2	$PL1_{90^o}$	$PL2_{0^o}$	$PL0_{90^o}$	$PL1_{0^o}$	$PL2_{90^o}$	$PL0_{0^o}$
3	$PL2_{0^o}$	$PL0_{90^o}$	$PL1_{0^o}$	$PL2_{90^o}$	$PL0_{0^o}$	$PL1_{90^o}$
4	$PL0_{90^o}$	$PL1_{0^o}$	$PL2_{90^o}$	$PL0_{0^o}$	$PL1_{90^o}$	$PL2_{0^o}$
5	$PL1_{0^o}$	$PL2_{90^o}$	$PL0_{0^o}$	$PL1_{90^o}$	$PL2_{0^o}$	$PL0_{90^o}$

TABLE III: Number of polygon constraints placed per scenario.

	PL0		**PL1**		**PL2**		Σ
	0^o	90^o	0^o	90^o	0^o	90^o	
Participant 1	6	4	3	3	2	2	20
Participant 2	0	4	3	3	2	1	13
Participant 3	0	4	1	0	0	1	6
Participant 4	4	4	3	3	2	3	19
Participant 5	3	3	3	1	2	2	14
Sum	32		23		17		72

J. Hypotheses

It was hypothesized that with increasing perturbation location the number of placed constraints by the controller would increase (H.I). Next to this, it was hypothesized that the structure added to the airspace by the controllers, would lead to an increase in minimum and average robustness, compared to the automation baseline. In return, this would also mean an increase in added track miles and an increase in knock-on reroutes (H.II). It was also hypothesized that the perturbation location would have no significant influence on the controller's situation awareness and trust in the system (H.III).

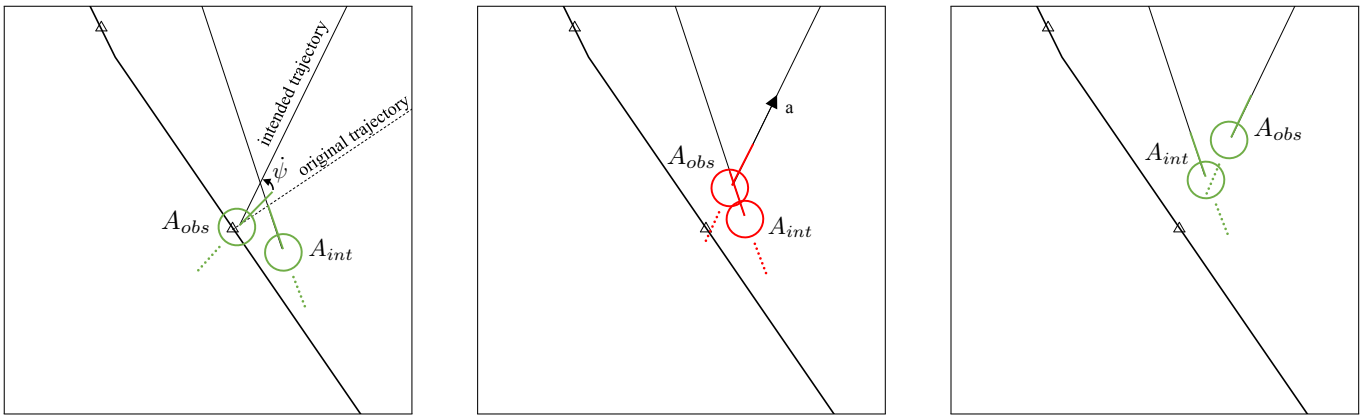
VI. Results

A. Constraints placement

Each participant was asked to draw a circle constraint around the perturbation, but was free to choose the size of the constraint, as long as it fully covered the perturbation. Analysis of the scenarios showed that each participant kept the same strategy throughout the scenarios. Participants 1, 2 and 5 drew a small buffer of around 2.5 NM around the perturbations, thus giving a larger weight to the safety buffer criteria. Participants 3 and 4 chose to draw the circle constraints tightly around the perturbation, shifting the weight towards minimizing additional trajectory length. When reflecting back on the decision ladder in Fig. 8, the circle constraints were drawn by the participants by following the shortcut from the set of observations to the task definition.

Next to the circle constraint required to guide traffic around the perturbation, the participants only drew polygon constraints with three corners. The number of constraints drawn is shown in Table III. The rotation in scenarios was not found to be significant according to a Friedman test, therefore the data between the 0^o and 90^o scenarios are averaged for further statistical tests. A Friedman test revealed that the perturbation location significantly affected the number of placed constraints ($\chi^2(2) = 7.11, p = 0.03$). Pairwise comparison, after adopting a Bonferroni correction, confirmed only a significant difference between PL0 and PL2.

It was hypothesized (H.I) that for increasing perturbation location, the number of drawn constraints by the participants would be larger, however the statistical tests reject this hypothesis. For PL0 most participants would still place a polygon constraints to be sure the automation would find a solution for the shortest way around the perturbation, even though by just



(a) $t = 0s$ (b) $t = 40s$ (c) $t = 90s$

Fig. 14: Transient effects leading to a loss of separation. (a) A_{obs} turning to the intended trajectory determined by the algorithm to avoid a conflict with A_{int}. (b) loss of separation occurred as algorithm did not take rotation and forward acceleration into account of A_{obs}. (c) past loss of separation, A_{obs} reached intended velocity.

placing a circle constraint around the perturbation would lead to nearly the same result. This also explains why most of the times the participants chose to add four polygon constraints, as there were four airways intersecting the perturbation. The same effect was shown for PL1 and PL2, where in PL1 most participants chose to use three constraints, as there were three airways crossing the perturbation and in PL2 only two.

Most participants placed at least one or two constraints, except for Participant 3, who added no constraints in three out of six runs and only one constraint in two runs. Looking at the Latin square design in Table II, it suggests that a training effect may have influenced the control strategy of Participant 3, which gave the participant more trust that the automation would figure out the best solution. The same training effect influenced Participant 2 in the PL0 scenario, who did not place any additional constraints the second time. Participants 2 and 3 thus partly support the hypothesis that at PL0 less constraints would be drawn.

B. Sector-Based Robustness

Figs. 18a and 18b show all minimum sector-based robustness data points for each scenario and traffic density calculated with and without perturbation, respectively. It shows the lowest achieved value of minimum robustness over all aircraft in a run. Figs. 18c and 18d show the average sector-based robustness, which is an indication of the robustness of the complete airspace. The vertical lines in all four plots show the spread between participants for a certain perturbation location.

First thing to notice is that the minimum robustness in a few cases reached a value of 0.0, which means that a loss of separation occurred. Out of the 1950 flights in total, six losses of separation occurred. One of which that was close to the perturbation and thus does not show up as a minimum robustness value of 0.0 in Fig. 18b. The losses of separation were unexpected as the algorithm is programmed such that it will prevent these. Close inspection of the data showed that these losses of separation can be be accounted to the transient effects that take place. The algorithm assumes instantaneous state changes, whereas the aircraft flying in the airspace need time to accelerate or rotate to the set speed or heading. It was expected that the assumption of instantaneous state changes would not affect the results due to the large time scale of planning the algorithm was used for. However, due to the fact that the algorithm would plan aircraft that close to each other, the transient effects actually do have an effect. Fig. 14 illustrates how a loss of separation occurs due to the transient effect not taken into account. It shows that the observed aircraft (A_{obs}) is still rotating and accelerating, whereas the algorithm assumed the aircraft was already at the set heading and up to speed.

The rotation in scenarios was not found to be significant according to a Friedman test for both the minimum and average robustness, therefore the data between the 0^o and 90^o scenarios are averaged for further statistical tests.

Friedman tests showed that the experimental condition significantly affected the minimum robustness for both with perturbation ($\chi^2(5) = 22.05, p < 0.01$) and without perturbation ($\chi^2(5) = 11.63, p < 0.04$) in the calculation. Further pairwise comparison with Bonferroni showed no significant differences, except that the minimum robustness was higher for PL0 compared to PL2 in the low traffic density scenario with perturbation taken into account.

Also for the average robustness, Friedman tests revealed that it is significantly affected by the experiment conditions, both for with perturbation ($\chi^2(5) = 19.97, p < 0.01$) and without perturbation ($\chi^2(5) = 23.63, p < 0.01$) in the calculation. For the average robustness with perturbation a Bonferroni corrected pairwise comparison revealed only two significant differences between the high and low traffic density scenarios, which was to be expected. Between the perturbation locations

within both traffic densities, no significant difference was found. For the average robustness without the perturbation taken into account pairwise comparison found that within the high traffic density the average robustness was significantly lower for PL2, compared to PL0 and PL1, after adopting a Bonferroni correction.

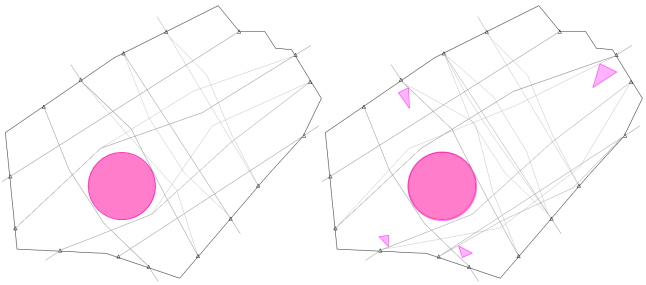
A Friedman test showed that the minimum robustness without perturbation taken into the calculation is significantly higher than with perturbation ($\chi^2(1) = 5.00, p < 0.03$). Exactly the same result was found for the average robustness ($\chi^2(1) = 5.00, p < 0.03$). For a fair comparison of the participants to the baselines the two metrics that do not take into account the perturbation are considered, as this takes away the bias and effect of how close participants chose to let the aircraft fly by the perturbation.

Also, to compare the data in detail, the data between the 0^o and 90^o degrees scenarios are not averaged, in comparison to the statistical tests, due to the large difference that sometimes occurs within the data of a participant. For example, in Fig. 18a Participant 3 managed to reach in PL0 a minimum robustness of 0.0 and one of 0.29. The control actions belonging to these scores are shown in Fig. 15. The two control actions differ much, where the participant in one scenario added quite some structure to the airspace and in the other scenario he just left it to the automation by drawing a tight circle constraint around the perturbation. For this perturbation location that was a logical choice when looking at the data in Fig. 18, as the automation scores better than nearly all participants in both metrics. Also, when looking at Fig. 19, which shows the minimum robustness over time for both scenarios, together with an comparison to the two baselines, it shows clearly that in the scenario where the participant did add structure to the airspace the difference in minimum robustness over time is quite considerable. As mentioned in Section VI-A it is expected that the large difference in control strategy for this participant is due to a training effect in the experiment, where the participant gained more trust over time in the automation and thus did not feel the need to structure the airspace.

Furthermore, for PL0 in the high traffic density condition most of the participants perform slightly worse than both baselines. In the low traffic density condition the results are on par with the automation baseline.

For PL1, the participants also predominantly scored worse than the automation baseline, this effect is now also shown for the low traffic density condition. Interesting to note is the positive performance of Participant 1 in the 90^o scenario. Closer inspection showed that the participant chose a control strategy in which he steered the aircraft from the airway intersecting the perturbation North to South, the long side around. The algorithm, however, could not find solutions this side around with only two additional waypoints and therefore automatically increased to three additional waypoints. With the increase in waypoints the automation was able to find solutions the short side around the perturbation and the drawn constraint, as shown in Fig. 16. This increase in waypoints allowed the algorithm to find more optimal solutions, and thus the robustness increased.

In PL2 the participants also predominantly did not out perform the automation baseline. Especially in the high traffic density scenario they perform worse in terms of average robustness, the only exception is Participant 4. In the 0^o run the participant managed to increase the minimum robustness by 0.1, while in the 90^o run he succeeded to increase the average robustness slightly, but at the cost of a loss of separation. Both strategies are shown in Fig. 17, and Fig. 20 shows the minimum robustness over time for both strategies. The data show minimum robustness of 0.0 was only reached once in the 90^o scenario. For the rest of the time it performed on par with the automation baseline. The participant was still outperformed by the TSR baseline.



(a) 0 (b) 0_{90^o}

Fig. 15: Two control strategies by participant 3 for PL0.

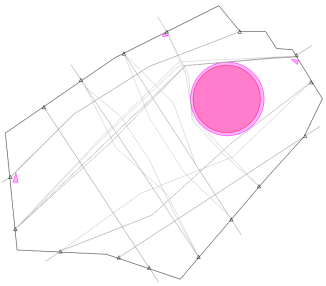
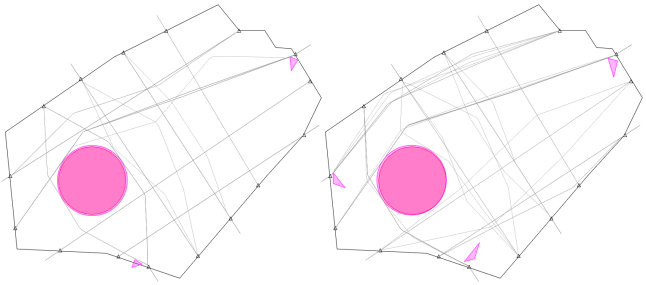
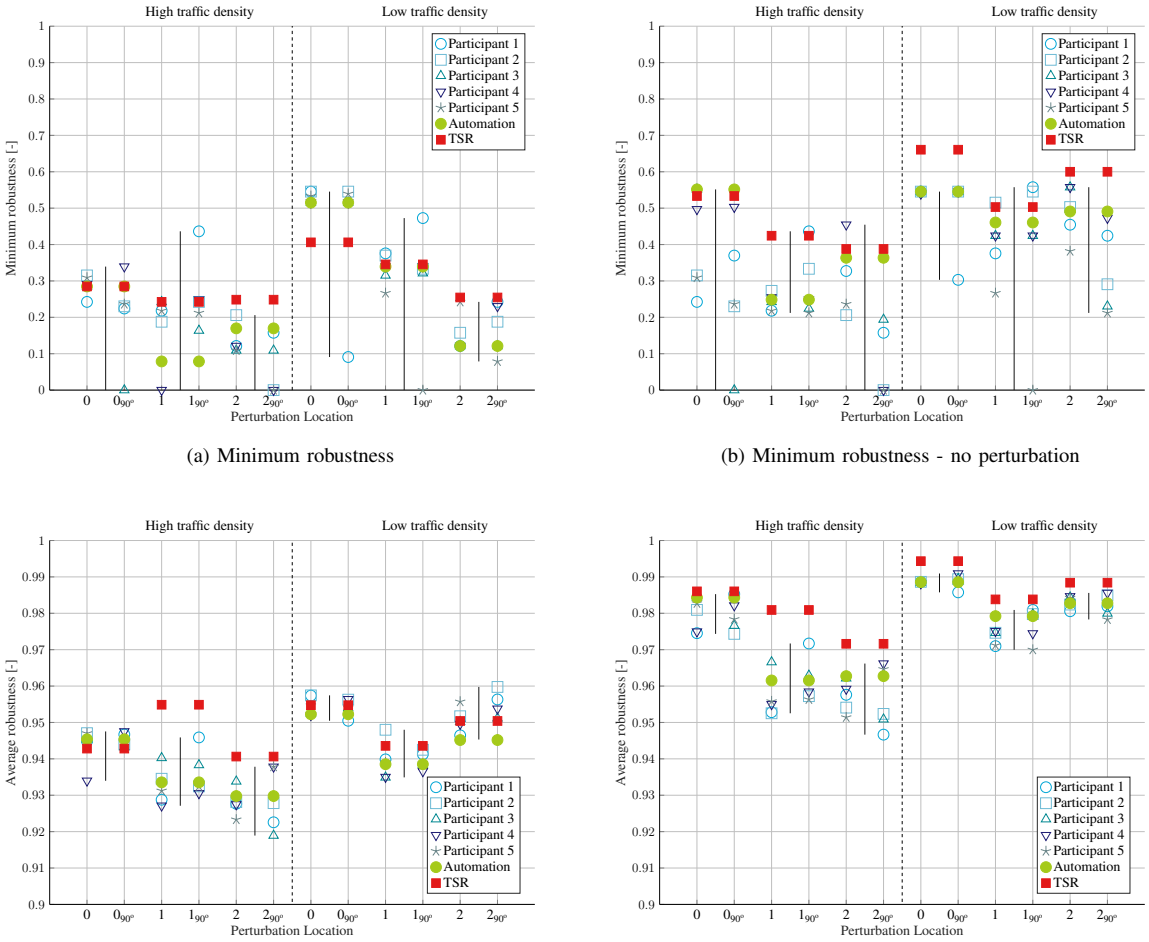


Fig. 16: Control strategy Participant 1 for $PL1_{90^o}$.



(a) 0 (b) 0_{90^o}

Fig. 17: Two control strategies by Participant 4 for PL2.



(a) Minimum robustness

(b) Minimum robustness - no perturbation

(c) Average robustness

(d) Average robustness - no perturbation

Fig. 18: Minimum and average sector robustness per perturbation location and traffic density.

When looking at the data with the perturbation taken into the robustness calculations, one can see that for the minimum robustness in PL1 nearly all participants outperformed the automation baseline. Closer inspection showed that all participants chose to guide the airway from East to West through the center of the perturbation along the North side of the perturbation, apparently having a positive influence on the minimum robustness. In other scenarios the robustness performance has the same trend as with taking the perturbation not into account; occasionally outperforming the automation, but most of the time having similar or worse performance.

C. *Efficiency*

Fig. 21a shows the added track miles per participant for each scenario. The track miles are shown relative to the trajectories before the perturbation is introduced. The knock-on reroutes are shown in Fig. 21b, which are defined as aircraft that need to deviate from their original trajectory due to conflicting with other aircraft and not due to the perturbation.

The rotation in scenarios was not found to be significant according to a Friedman test for both the added track miles and knock-on reroutes, therefore the data between the 0^o and 90^o scenarios are averaged for further statistical tests.

A Friedman test showed that the experimental condition significantly affected the added track miles ($\chi^2(5) = 23.51, p < 0.01$). Pairwise comparison, using a Bonferroni correction, revealed that in the high traffic density scenarios the added track miles were significantly higher for PL2 compared to PL0 and PL1.

The knock-on reroutes was also found to significantly affect the experimental condition (Friedman: $\chi^2(5) = 22.70, p < 0.01$). However, additional pairwise comparison found no significant differences within the traffic density groups, after adopting a Bonferroni correction.

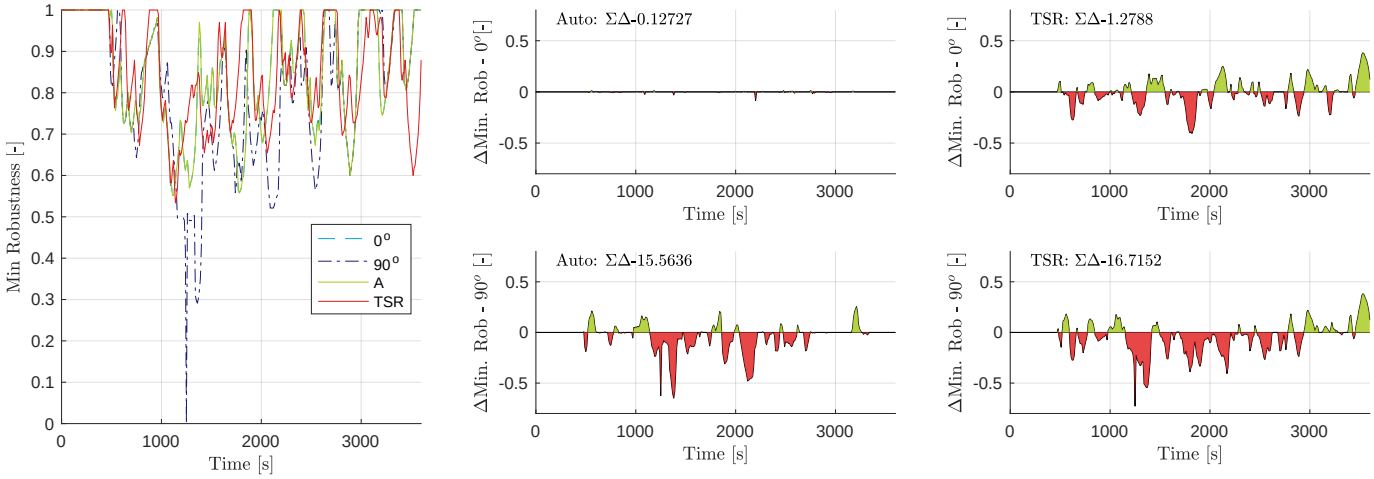


Fig. 19: Minimum robustness over time by Participant 3 for PL0 in the high traffic density scenario. Difference in minimum robustness as compared to the automation and TSR baselines is shown on the right.

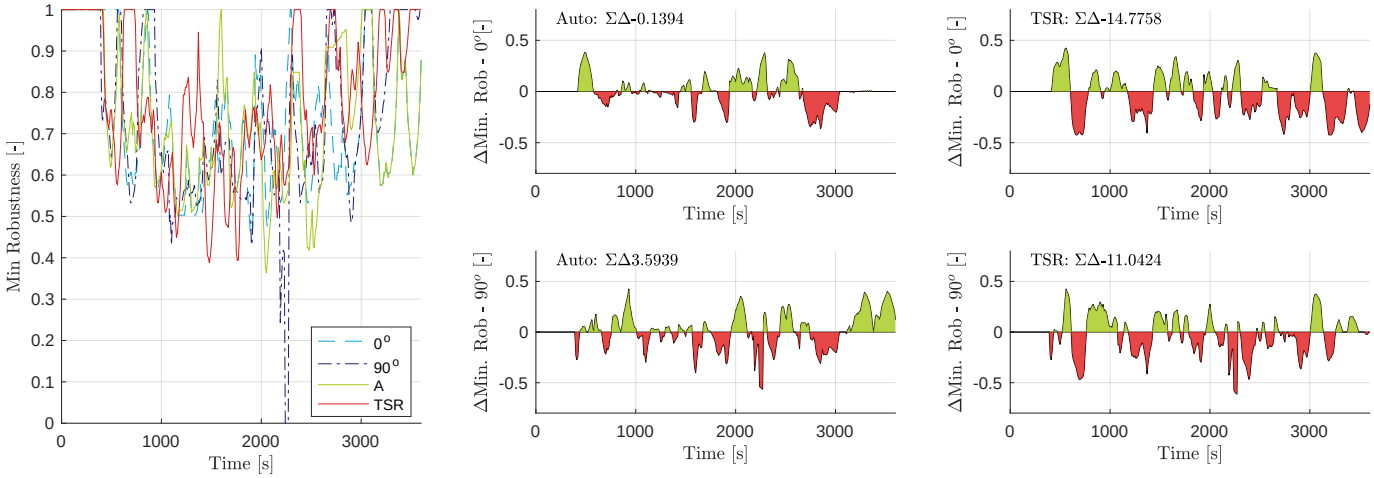


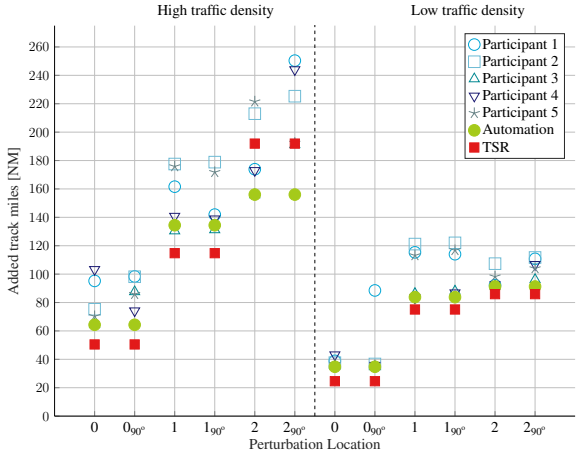
Fig. 20: Minimum robustness over time by Participant 4 for PL2 in the high traffic density scenario. Difference in minimum robustness as compared to the automation and TSR baselines is shown on the right.

Compared to the automation baseline, the added track miles of the participants predominantly increases. This is an expected result, as the participants add constraints to the airspace around which the aircraft need to fly, thus causing longer trajectories. This same result was found for the knock-on reroutes.

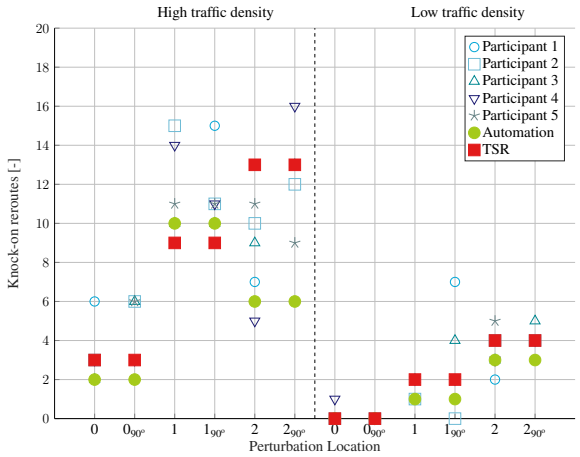
The knock-on reroutes are on average larger for the PL1 scenario than PL2 with high density traffic. It was expected to be larger for the PL2 case, however in PL1 the traffic needed to be rerouted close to a busier airway, which explains the difference in knock-on reroutes.

D. Situation awareness

After each run, participants were asked to fill out the SASHA questionnaire regarding their situation awareness. A Friedman test showed no significant effect in SASHA score between the experiment conditions ($\chi^2(2) = 1.78, p = 0.41$). Therefore, only the overall situation awareness of the interface is discussed. Fig. 22 shows per question the frequency of ratings over all participants and scenarios. Generally speaking, the outcome of the SASHA questionnaire is positive. From Figs. 22a and 22b can be concluded that in most runs the participants were able to structure the airspace as they wanted, with the automation being steered in the direction that was

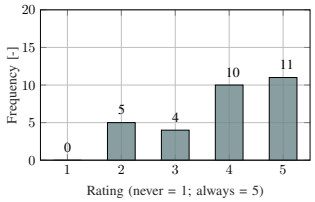


(a) Added track miles

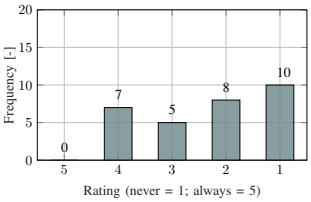


(b) Knock-on reroutes

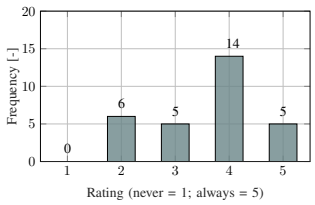
Fig. 21: Added track miles and knock-on reroutes per perturbation location and traffic density.



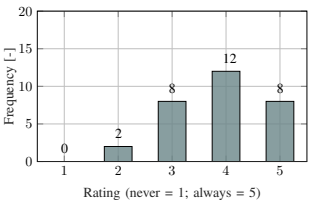
(a) Did you have the feeling you were able to plan and organize your work as you wanted?



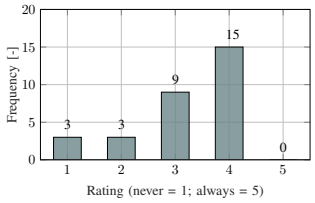
(b) Have you been surprised by a trajectory rerouting that you were not expecting?



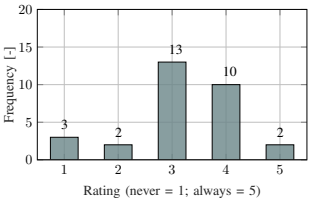
(c) Were you able to predict the result of the automation, based on your constraints?



(d) Did you have the feeling you were able to steer the automation?



(e) Do you think the solution space provided you with useful information?



(f) Did the solution space help you have a better understanding of the situation?

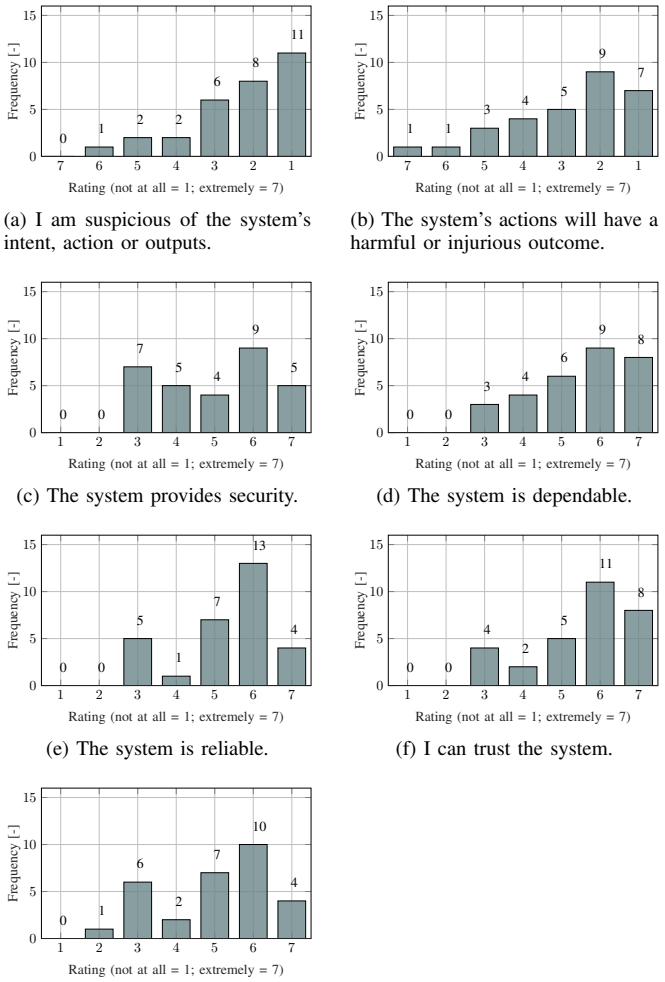
(g) How would you rate your overall situation awareness during this exercise?

Fig. 22: SASHA questionnaire.

required. Although in a number of runs the participants were surprised by the actions of the automation, as seen in Fig. 22b. From Figs. 22e and 22f it follows that the airway solution space in most cases did support the participants in their task, albeit moderately. Overall, the situation awareness during most runs was considered to be good, following from question Fig. 22g.

E. Checklist for trust between people and automation

Next to the SASHA questionnaire, participants were asked to complete a modified version of the checklist for trust between people and automation after each run. Again, a Friedman test showed no significant effect in trust score between the experiment conditions ($\chi^2(2) = 4.11, p = 0.13$), thus only the overall trust in the interface is discussed. The frequency of ratings per question are shown in Fig. 23. Interesting to see is that overall the participants seem to trust the system quite well. Figs. 23c, 23e and 23f, however, do show a few lower ratings. These lower ratings were mainly due to a bug in the algorithm, which occurred once during 1 or 2 runs for each participants. The algorithm reroutes a trajectory of an aircraft that is in conflict with another aircraft, but after rerouting the aircraft were still in conflict. The conflict was solved by simply running the algorithm again. Closer inspection of this bug shows that if one of the waypoints is moved by a few pixels the aircraft is already out of conflict, thus it is expected that the bug can be



(a) I am suspicious of the system's intent, action or outputs.

(b) The system's actions will have a harmful or injurious outcome.

(c) The system provides security.

(d) The system is dependable.

(e) The system is reliable.

(f) I can trust the system.

(g) I am familiar with the system.

Fig. 23: Checklist for trust between people and automation questionnaire.

assigned to rounding errors in the discretization of the airspace.

F. Debrief

During the debriefing the participants were asked to assess and provide feedback on the interface and experiment. Regarding the airway solution space the participants were generally positive, as it gave them insights in how traffic streams would flow and where the freedom of movement is. It also supported them in drawing the constraints. A problem indicated with the airway solution space was that it was fixed to only two waypoints, thus if no solution was found by the automation with two additional waypoints, the automation chose paths with additional waypoints the participants were not expecting. One participant also indicated that the placement of the waypoints feels unnatural, thus he would like to influence the position of the curved areas.

Drawing with the provided constraints (polygon and circle) was rated fairly good and intuitive by most participants, although one of them described working with the constraints as cumbersome and that it feels like a hack to trick the automation. A problem raised with the constraints is the influence of the constraints on other traffic streams, thus placement of constraints needed to be considered carefully. One of the participants thought the baseline constraints around the perturbation should be drawn automatically. Participants also indicated that they felt they could not place the constraints with certainty, due to the possible changeability of the airspace and since too little information about the traffic was available.

The participants said that they were generally able to influence the automation, but with some surprises. Apart from the bug discussed earlier, also when the automation started to reroute trajectories with three additional waypoints, they felt out of control. One participant indicated that he wanted online control of the airspace and thus the possibility to adjust the constraints when the scenario was started. One participant indicated that influencing the automation felt useless. He indicated that the reason for this is because he had the same amount of information available as the automation about the airspace, and thus could not make better decisions than the automation. If he would, for example, have information available of traffic densities in the airways, he would be able to make better decisions as he then would be able to reason upon how the traffic scenario would unfold itself in the future. Something the automation cannot do, as the automation only reroutes trajectories once an aircraft enters the airspace.

In general, multiple participants indicated that providing more information regarding traffic densities in certain airways would be beneficial. Another idea was to create stream-specific constraints to prevent unwanted influence of constraints on other airways. One participant did not like the complete blocking of the algorithm with the constraints and suggested to penalize certain routes (e.g., left is "mud", high cost, and right is asphalt, low cost), as such this could avoid introducing artificially longer paths in an unwanted direction. A suggestion regarding the experiment is to evaluate more realistic air traffic scenarios, which are known to the controller and thus have more information regarding the airspace.

VII. Discussion

The goal of this research was to design a conceptual ecological interface for flow-based perturbation management by means of influencing a trajectory planning algorithm. Looking at individual planned trajectories, the selected node-based trajectory flexibility metric algorithm by Idris performed well in this research, the algorithm was in principle predictable and consistent.

An improvement to the algorithm can be made as to where the waypoints will be placed in the longitudinal direction. In the current implementation it was chosen to discretize this linearly, thus with one waypoint at one third of the RTA and one at two thirds. This led to sub-optimal trajectories, which was clearly seen when perturbations were placed close to the entering or exit waypoint. To improve this, the human

controller could be given control over as to where to place the curved areas in which the waypoints will be placed. Additionally, the algorithm should be improved by taking into account the transient effect of aircraft reaching a new state. As was shown losses of separation would occur due to the assumption of instantaneous state changes.

The intentions of the algorithm were made visible to the human controller by means of the airway solution space. The evaluation revealed that the participants were generally positive in the support it provided, it gave them a good image of how traffic streams would flow. However, in cases were the algorithm could not find a solution with only the two additional waypoints as set in the airway solution space, and thus needed to increase to three, or more, additional waypoints, use of the airway solution space was lost. In those cases the algorithm planned trajectories participants were not expecting, which lowered their situation awareness. Improvements should be made in what to do in those cases. One could opt to ask the controller in those cases to plan the trajectory manually with the TSR. Another option is to constrain the algorithm even more, as analysis of the strategies showed that participants only placed the constraints so that the aircraft would be planned one way around. Drawing a small constraint is enough to force the algorithm to one side with only two additional waypoints, but as soon this increases to three waypoints the algorithm can find a solution around the drawn constraint. Therefore a logical solution seems to constrain the algorithm, stream-specifically, by allowing the controller to adjust the airway solution space. For example, by making one side of the airway solution space smaller by dragging this side to the center line of the airway. This might also solve problems where constraints were not drawn accurately enough and thus still would make choices the controller does not expect. Also, it could reduce the sensitivity of the control strategies as the human controller is more constrained regarding control actions.

Some participants indicated that placing the constraints felt contrived and that their actions would not lead to a better solution than as when it would be fully left to the automation. To overcome this, multiple actions should be taken. One is to provide the human controller with additional information of the airspace, such as traffic densities in certain airways. Another option would be to allow the controller to 'scroll' back and forth through time, to see how traffic streams would flow, before the perturbation. Next to this, allowing the controllers to adjust their set constraints, dependent on the results they have been shown, would increase acceptation of the automation and interface.

The interface was evaluated for a structured airspace with eight different traffic streams. However, the way airspaces will be structured in the future is still unclear. It is expected that effectiveness of the interface depends on the airspace, which was also seen between the three perturbation locations in the same airspace. Although the data disapproved hypothesis H.I, it is still expected that with more training and knowledge of the algorithm the influence a controller will and can exert on the algorithm is higher with PL2 as compared to PL0.

The data showed that the influence of a human controller predominantly lowered the robustness, when considering the case in which the perturbation was not taken into the robustness calculation. However, with the perturbation taken into account slightly better results were acquired. This was expected, as participants were able to draw a buffer zone around the perturbation, which lead to increments in robustness as compared to the automation baseline. The efficiency decreased in nearly all cases, due to an increase in added track miles and knock-on reroutes. Due to the fact that the robustness was not increased, when not taking the perturbation in the robustness calculation, hypothesis H.II is not fully accepted. Possibly with more training and knowledge of the airspace better results can be achieved. However, it should be reiterated that the automation worked sub-optimal, with only two additional waypoints and utilizing the shortest path cost function. Perhaps with more waypoints and the robustness or adaptability cost function better results can be achieved, although care should be taken of keeping the automation understandable to the human controller. In addition, if the number of waypoints are set too high and the human needs to take over, he or she will have a difficult time adjusting trajectories with many waypoints.

Hypothesis H.III can be accepted, as no significant effect was found on the situation awareness and trust in the systems of the participants between the three different perturbation locations. Overall the participants considered their situation awareness during the evaluation as good. Also, most of the time they did trust the system. One of the reasons to allow the participants to influence the trajectory planning algorithm, was that in this manner they would be to prevent chaos and black-box behaviour generated by the automation. As the situation awareness questionnaire suggests, this indeed seems to be prevented, however, more research is required in this specific area as there was no data to compare the results to. Also included in this future research should be whether other ATCos will be able to take over the airspace when this is required, for example at the end of the work shift of an ATCo.

In this research, the participants were only allowed to structure the airspace before aircraft entering the airspace. Possibly, with the improvements stated above, the interface can be used while aircraft are entering the airspace. As the results of the TSR baseline showed, which did have the possibility to adjust aircraft while in the airspace, a considerable higher sector robustness can be reached. However, it is expected in this case that with this interface the human controllers will attend to each aircraft individually and thus essentially shifting back to the manage aircraft missions level. However, it is recommended to further investigate this in future research. Special focus in this research should be on how participants react when a perturbation occurs, while they are monitoring the airspace.

Overall, it is questioned whether this type of flow-based control is the best solution to the problem of increased air traffic. Although the goal was influencing the algorithm and making its choices clear to the controller was certainly achieved, there are still quite some complications to consider it as an effective tool for flow-based perturbation management. A possible solution is to use the interface in combination with the TSR, where this designed interface is used to reroute all trajectories beforehand and the TSR to improve the specific

trajectories which lead to a lower sector robustness.

Future research should implement the improvements discussed, investigate how the tool can be made effective for multiple scenarios, research situation awareness with and without being able to influence the automation and experiment how controllers react when automation failures are induced while monitoring the airspace.

VIII. Conclusion

This paper presented the conceptual design of an ecological interface for 4D flow-based perturbation management in air traffic control. A first evaluation showed that the interface supported the controller's understanding and situation awareness of the airspace, and allowed for influencing the trajectory planning algorithm. In the evaluation the perturbation location and traffic density were varied and the results were compared in terms of robustness and efficiency. Results showed that in most cases influence of the human does not have a positive result on the outcome in terms of robustness and efficiency. Control strategies applied to the different perturbation locations were not as expected, but it suggested that this happened due to a lack of training and knowledge. To improve the interface multiple ideas have been suggested, but the effectiveness of the interface for flow-based perturbation management is questioned. A possible solution would be to use the interface in combination with the TSR, where this interface acts as a first planning tool, followed up by optimizing low robustness trajectories with the TSR.

References

[1] R. Klomp, C. Borst, M. M. van Paassen, and M. Mulder, "Expertise Level, Control Strategies, and Robustness in Future Air Traffic Control Decision Aiding," *IEEE Transactions on Human-Machine Systems*, vol. 46, no. 2, pp. 255–266, 2016.

[2] Eurocontrol, "SESAR Concept of Operations," 2007.

[3] M. Paskota and O. Babic, "Correspondence Analysis in Examination of Reasons for Flight Schedule Perturbations," *Journal of the Brazilian Air Transportation Research Society*, vol. 2, no. 2, pp. 9–23, 2006.

[4] L. Bainbridge, "Ironies of automation," *Automatica*, vol. 19, no. 6, pp. 775–779, 1983.

[5] K. J. Vicente and J. Rasmussen, "Ecological interface design: Theoretical foundations," *IEEE transactions on systems, man, and cybernetics.*, vol. 22, no. 4, pp. 589–606, 1992.

[6] M. M. van Paassen, C. Borst, R. Klomp, M. Mulder, P. van Leeuwen, and M. Mooij, "Designing for shared cognition in air traffic management," *Journal of Aerospace Operations*, vol. 2, no. 1-2, pp. 39–51, 2013.

[7] C. Borst, J. M. Flach, and J. Ellerbroek, "Beyond Ecological Interface Design: Lessons From Concerns and Misconceptions," *Ieee Transactions on Human-Machine Systems*, vol. 45, no. 2, pp. 164–175, 2015.

[8] M. H. J. Amelink, "Ecological Automation Design, Extending Work Domain Analysis," Ph.D. dissertation, Delft University of Technology, 2010.

[9] A. Richards, "Constraining the Sense of Conflict Resolution: Supervision of Route Optimization," *First SESAR Innovation Days*, no. December, pp. 1–7, 2011.

[10] O. Turnbull and A. Richards, "Examples of Supervisory Interaction with Route Optimizers," *Second SESAR Innovation Days*, no. November, pp. 1–8, 2012.

[11] M. M. Van Paassen, C. Borst, J. Ellerbroek, M. Mulder, and J. M. Flach, "Ecological Interface Design for Vehicle Locomotion Control," *IEEE Transactions on human-machine systems (submitted, under review)*, 2018.

[12] L. Yang, J. Qi, J. Xiao, and X. Yong, "A literature review of UAV 3D path planning," *Proceedings of the World Congress on Intelligent Control and Automation (WCICA)*, no. March, pp. 2376–2381, 2015.

[13] D. Ferguson, M. Likhachev, and A. Stentz, "A guide to heuristic-based path planning," *Proceedings of the International Workshop on Planning under Uncertainty for Autonomous Systems, International Conference on Automated Planning and Scheduling (ICAPS)*, pp. 1 – 10, 2005. [Online]. Available: http://cs.cmu.edu/afs/cs.cmu.edu/Web/People/maxim/files/hsplanguide{_}icaps05ws.pdf

[14] C. E. Billings, *Aviation Automation: The Search for A Human-Centered Approach*. CRC Press, 1997.

[15] L. Yang, J. Qi, D. Song, J. Xiao, J. Han, and Y. Xia, "Survey of Robot 3D Path Planning Algorithms," *Journal of Control Science and Engineering*, 2016.

[16] S. Karaman and E. Frazzoli, "Sampling-based Algorithms for Optimal Motion Planning," *International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011. [Online]. Available: papers3://publication/uuid/E8F0E216-BDAB-425B-9305-B43797C655C3

[17] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[18] P. E. Hart and J. Nils, "Formal Basis for the Heuristic Determination of Minimum Cost Paths," *IEEE Transactions of Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.

[19] N. J. Nilsson, *Principles of Artificial Intelligence*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1980.

[20] H. R. Idris, T. El-wakil, and D. J. Wing, "Trajectory Planning by Preserving Flexibility: Metrics and Analysis," *Proceedings of the AIAA Guidance Navigation and Control (GNC) Conference*, no. August, pp. 1–14, 2008.

[21] H. R. Idris, N. Shen, T. El-Wakil, and D. J. Wing, "Analysis of Trajectory Flexibility Preservation Impact on Traffic Complexity," *AIAA Guidance, Navigation, and Control Conference*, no. August, pp. 1–14, 2009. [Online]. Available: http://arc.aiaa.org/doi/abs/10.2514/6.2009-6168

[22] H. R. Idris, N. Shen, and D. J. Wing, "Improving Separation Assurance Stability through Trajectory Flexibility Preservation," *Aviation*, no. September, pp. 1–11, 2010.

[23] N. Naikar, R. Hopcroft, and A. Moylan, "Work Domain Analysis: Theoretical Concepts and Methodology," Air Operations Division, DSTO Defence Science and Technology Organisation, Edinburgh, Tech. Rep., 2005.

[24] J. Rasmussen, "Skills Rules and Knowledge, Other Distinctions in Human Performance Models," *IEEE transactions on systems, man, and cybernetics.*, vol. 13, no. 3, pp. 257–266, 1983.

[25] "Sesar joint undertaking — multi-sector planning," date last accessed 14-09-2017. [Online]. Available: https://www.sesarju.eu/sesar-solutions/conflict-management-and-automation/multi-sector-planning

[26] "Eurocontrol - coping with soaring air traffic demand," date last accessed 14-09-2017. [Online]. Available: https://www.eurocontrol.int/press-releases/coping-soaring-air-traffic-demand-new-sectors-improve-traffic-handling-airspace

[27] "Eurocontrol - statistics and forecasts," date last accessed 15-09-2017. [Online]. Available: https://www.eurocontrol.int/statfor

[28] Eurocontrol, "The Development of Situation Awareness Measures in ATM Systems," European ATM programme, Tech. Rep., 2003. [Online]. Available: https://www.eurocontrol.int/sites/default/files/content/documents/nm/safety/safety-the-development-of-situation-awareness-measures-in-atm-systems-2003.pdf

[29] J.-Y. Jian, A. M. Bisantz, and C. G. Drury, "Foundations for an empirically determined scale of trust in automated systems," *International Journal of Cognitive Ergonomics*, vol. 4, no. 1, pp. 53–71, 2000.

Part II

Book of Appendices

A

Literature Study

Note: this part has already been examined under the course AE4020

This chapter discusses the literature study performed for this thesis work. Section A.1 will elaborate on the Ecological Interface Design (EID) framework. This research continues on the development of the Travel Space Representation (TSR), thus in Section A.2 the design of the TSR will be discussed and looked into the results of the interface. The next tool that will be looked into is a taxonomy to categorize the interface in terms of automation, this will be discussed in Section A.3. For the interfaces an automated path or trajectory planning algorithm is needed. The general path planning algorithms will be discussed in Section A.4 and algorithms specifically designed for Air Traffic Management (ATM) are discussed in Section A.5. Section A.6 discusses why the trajectory algorithm by Idris was chosen. In Section A.7 elaborates on the workings of the trajectory planning algorithm and how it is implemented in the TSR software.

A.1. Ecological Interface Design

In this section the interface design framework EID theory will be summarized. EID was first introduced by Kim J. Vicente and Jens Rasmussen to increase safety in process control work domains [1]. To increase safety the goal of EID is to make constraints and relationships visible to the operator in the complex cognitive work domain, such that the operator can limit him or her self to higher order problem solving and knowledge-based decision making. This deeper understanding will lead to a critical advantage, namely that it greatly benefits in the acceptation of automation [2, 3]. This is because it is easier for the operator to observe and understand the automated agents choices.

The EID framework is based on the concepts of Abstraction Hierarchy (AH) and Skills, Rules & Knowledge (SRK) taxonomy, both adapted from Cognitive Systems Engineering (CSE) [4, 5]. In the following two subsections these concepts will be explained.

A.1.1. Abstraction Hierarchy

The AH is used to perform a Work Domain Analysis (WDA) to describe the complex work domain. The AH, as proposed by Jens Rasmussen, consists of five different levels of abstraction, where the top levels describe an higher order of abstraction [4]. The system is described at each of these levels in the AH using how and why relationships. Moving from the top of the AH downwards answers how the elements in the system are accomplished, whereas moving upwards answers why the elements are in the system. The relations are shown by using means-ends links [6].

The five levels of abstraction used in the AH are defined as follows:

- **Functional purpose** System objectives, a desired state of the environment.
- **Abstract function** High level functions, e.g. mass transport, storage, energy addition physics of the world.

- **Generalised function** System design choices, processes, principles.
- **Physical function** Functional processes of components.
- **Physical form** Appearance, anatomy, shape, material, location, etc.

A second dimension can be added to the AH, which leads to the Abstraction-Decomposition Space [7, 8]. The second dimension shows the whole-part decomposition of the system, thus the result is a two-dimensional matrix with the levels of abstraction at the vertical axis and the part-whole decomposition along the horizontal axis.

A.1.2. Skills, Rules & Knowledge Taxonomy

The SRK taxonomy was developed by Jens Rasmussen in 1983 to help interface designers combine human aspects of cognition and information requirements for a system [4]. The taxonomy is defined by three distinguishing levels of human behaviour that emerge when looking at representing constraints in a deterministic environment or system. The levels that emerge are skill-, rule- and knowledge-based behaviour of which each level represents a different level of cognitive control [6]. Each level of behaviour is related to a manner in which information is interpreted, which are Signals, Signs and Symbols, respectively. The levels and their relationship are shown in Figure A.1.

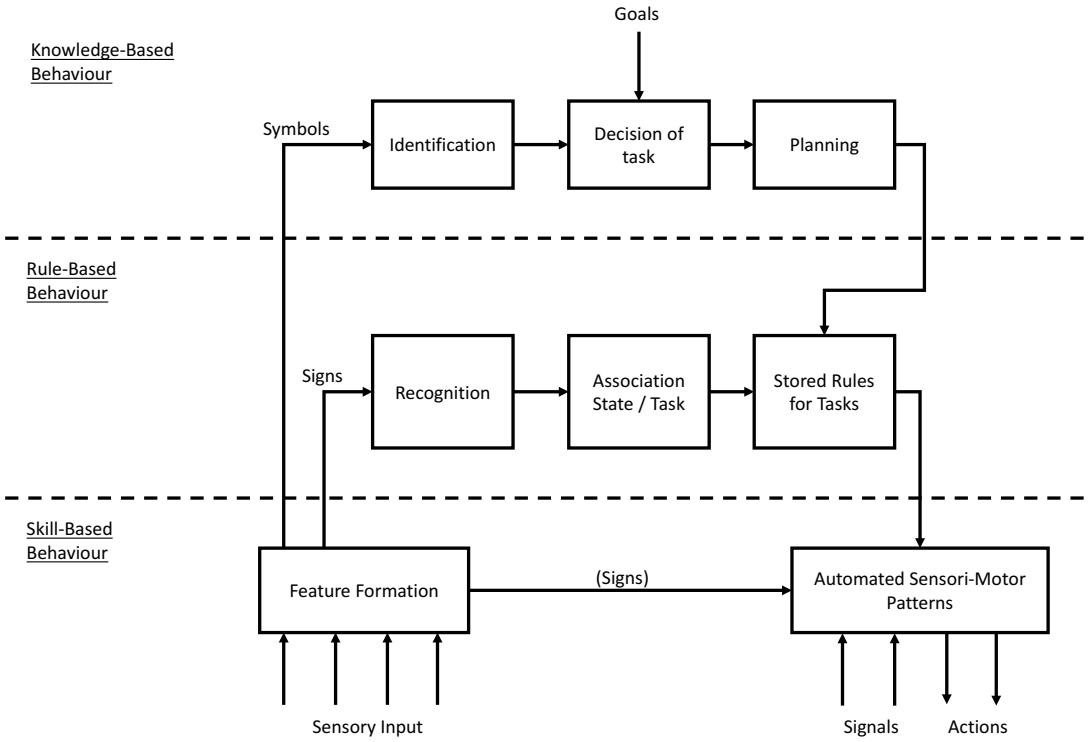


Figure A.1: Diagram that shows the three levels of behaviour in the SRK taxonomy [4].

Skill-based behaviour

Skill-based behaviour (SBB) is based on sensory-motor performance, which once started, requires no conscious control by the human. SBB takes place as smooth and highly integrated patterns of behaviour. For control at the SBB level the information is perceived as time-space *signals*, which are continuous quantitative indicators of the time-space behaviour of the environment processed by the human.

Rule-based behaviour

At the level of Rule-based behaviour (RBB) the behaviour is typically controlled by execution of a stored rule. The rule may have been communicated by others, self-developed or prepared for the occasion. Based on past

experiences, the most appropriated rule is selected. RBB information is perceived as *signs*, which serves to activate predetermined actions and cannot be processed directly.

Knowledge-based behaviour
During situations that are unfamiliar to the user and thus has no rules for control available, the control is moved to the higher level Knowledge-based behaviour (KBB). At this level plans for control are developed and selected and reflected to the goal. This form of control is slow, complex and error-prone. KBB is trigged by *symbols*, which can be formally processed and represents other information, variables, relations and properties.

A.2. Travel Space Representation

A promising ecological concept for 4D trajectory management is the Travel Space Representation (TSR), developed by the Joint ATM Cognition through Shared Representation (C-SHARE) project. The travel space forms the basis for shared human-automation cognition in the C-SHARE Joint Cognitive System (JCS). The first prototype of the TSR was developed by Van Paassen et al. as a constraint-based decision support tool for short-term trajectory based ATC [9]. The prototype was further developed and evaluated as an Air Traffic Control (ATC) tool by Klomp et al.[10–12]

Since the designers of the TSR followed the EID framework a WDA was performed by means of constructing of an AH. The AH for the TSR at a individual flight control level is found in Figure A.2. The shaded rectangular boxes indicate that they are directly visible in the plan view of the air traffic controller's interface, as shown in Figure A.3(a). The non-shaded rectangular boxes are made visible by the travel space once an aircraft is selected, which can be seen in Figure A.3(b).

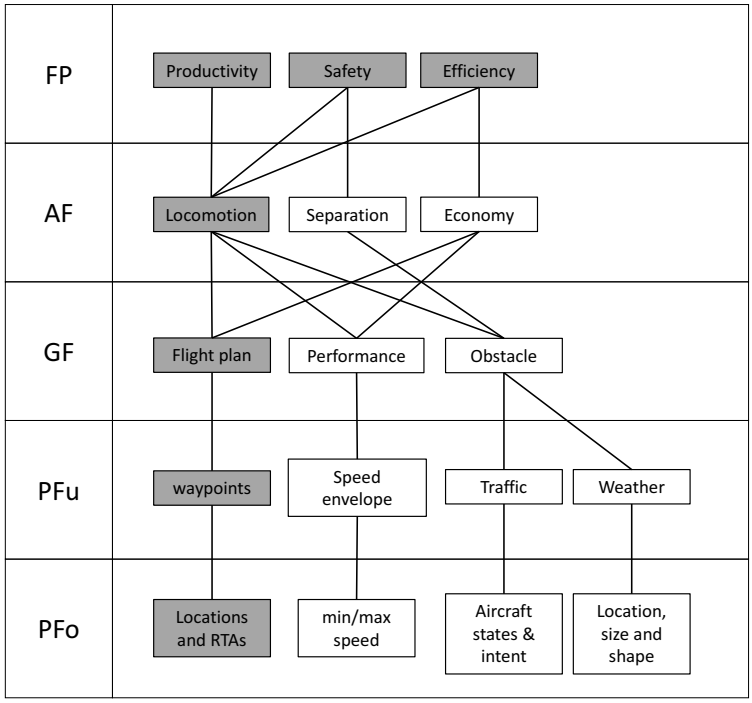


Figure A.2: Abstraction Hierarchy of the TSR .

By means of the principles of EID the TSR is designed such that is visualizes the boundaries of safe control, by visualizing a set of constraints for safe and feasible control actions for a selected aircraft [9]. The general shape of the TSR is an ellipse and is determined by the aircraft's performance envelope, such as its speed and bank angle limits, while still realizing its Required Time of Arrival (RTA) at the next waypoint [12]. Each ellipse within the TSR corresponds to a certain velocity of the aircraft, which realizes the RTA. On top of the shape the no-go areas are mapped, which result from additional constraints such as other traffic and restricted areas.

Figure A.3 shows the basic composition of the travel space representation in an hypothetical traffic scenario. In the scenario it is shown how a conflict resolution can be solved manually in three subsequent steps. The TSR can be directly manipulated with a mouse input device. In this situation the selected aircraft (A_{obs}) needs to be rerouted around the Restricted Airspace (RA) and intruding aircraft (A_{int}), while still maintaining the RTA at the waypoint *FIX*. This initial situation is shown in Figure A.3(a). By clicking on the aircraft (A_{obs}) the TSR is shown as illustrated in A.3(b). The TSR shows a safe and restricted field of travel due to the intruding aircraft (A_{int}). With the mouse cursor a waypoint *WP* is placed within the safe field of travel. By hitting enter on the keyboard the select flight path is selected and send to the aircraft. The modified flight path is shown in Figure A.3(c) and visualizes the new TSR for both segments. Please note that waypoints could be placed outside the TSR, however these points will likely lead to a conflict situation or not reaching the RTA.

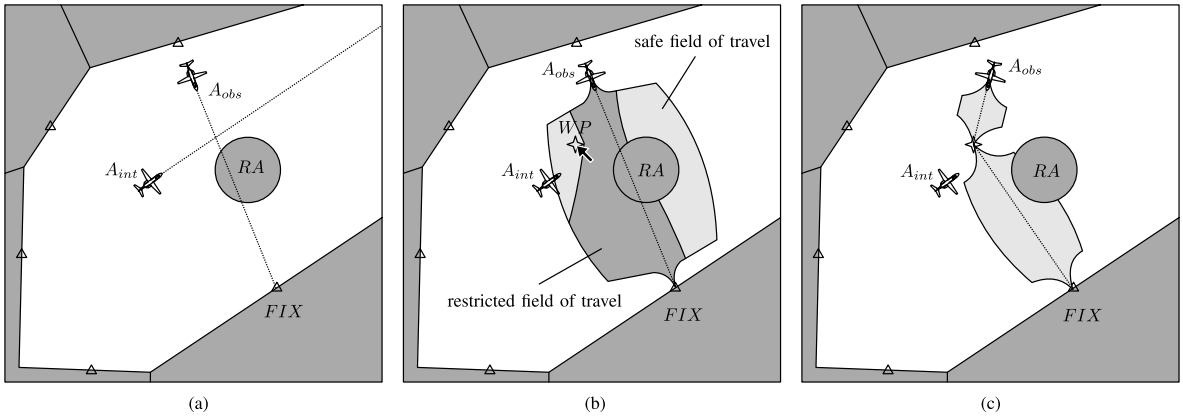


Figure A.3: TSR in an hypothetical traffic scenario. (a) Traffic scenario with two conflicting aircraft and a restricted airspace. (b) TSR and placement of an intermediate waypoint. (c) Resulting trajectory for the observed aircraft [12].

The TSR has been evaluated in two experiments by Klomp et al. [11, 12]. In the first experiment the overall objective was to see whether the TSR would support the task of en-route ATM in various traffic and perturbation scenarios. The result of this was that the tool never suffered from a breakdown and only four safety-critical events occurred out of 2232 total flights. Next to this the subjects accepted the tool and found it supportive. The workload did increase though, but was still experienced manageable, considering the novel system with relatively advanced settings [11].

In the second experiment the focus was on how effective the interface was to support the preservation of airspace robustness. Three groups of participants with various levels of operational ATC experience. The results showed that the level of expertise of the participants mainly influenced the robustness of the airspace. Which indicated that the interface would be most effective for experienced domain experts, because with the interface they can make smarter, knowledge-based, decisions with regard to the trade-off between safety. Although the control with this interface did not negatively influence the robustness, it might not be desirable in more dense traffic situations or situations with larger perturbations. In this situation an interface with a higher level of flow-based control might be more efficient [12].

Previous work by Pinto et al. was also focussed on creating a flow-based interface for 4D trajectory management in ATM [13]. Flow-based in this research means that multiple aircraft are controlled at the same time by the air traffic controller. By selecting multiple aircraft a joint TSR would show the possible field of safe travel for all selected aircraft. A human-in-the-loop experiment with this interface was performed by Nagaraj et al.[14] Results from the experiment suggested that multiple aircraft control was beneficial for maintaining airspace structure and robustness at low conflict angles. However, at higher conflict angles using multiple aircraft control would lead to head-on trajectories because the aircraft merge to the same selected waypoint, something air traffic controllers try to avoid.

A.3. Control Sophistication

In the past many Level of Automation (LoA) taxonomies have been thought off to rank systems on a scale of automation. Examples of such taxonomies are the levels of automation for ATC by Charles E. Billings [15] , the 10 levels by Wickens et al. [16] or the stages and levels by Parasuraman et al. [17] The problem with these taxonomies are however that they have been thought of many years ago and the discrete taxonomies do not necessarily have a place in these discrete taxonomies to place our systems.

Another interesting tool is the use of levels of control sophistication, as introduced by Amelink [18]. It is a way of showing at what level the system is controlling and what needs to automated to reach that level.

The levels of control sophistication can be used as an extension to the work domain analysis. Instead of adding the whole-part decomposition of a system to the AH as a second dimension, the levels of control sophistication are added as a second dimension. This is called the Abstraction-Sophistication Analysis (ASA) [18, 19]. The result of an ASA is a two dimensional matrix where the description at each level of sophistication all the levels below that level but not the whole system.

Each level of sophistication in the ASA is characterized by the following: [18]

- Each level of sophistication is a layer of control based on the levels below, achieving more sophisticated control of the system.
- Each level of sophistication holds functions and concepts for control specific to that level.
- Lower levels describe the inner control loops and higher levels describe the outer control loops of the total system.
- At each level of sophistication five levels of abstraction are used to find the means–ends structure of the control problem described at that level of sophistication. These levels correspond to the levels Rasmussen found for process control.

A setup of levels of sophistication for ATC was introduced by Borst et al. (in press) and is depicted in Figure A.4. The outer levels imply a more strategic form of control, whereas the inner nested loops are more tactical forms of control. For this research the aim is to let the Air Traffic Controller (ATCo) control the highest level of sophistication, the flow management level. All levels below it need to be automated. In the current TSR the ATCo is controlling in the aircraft missions level, with the levels below it again automated. It is also possible to shift from level of control sophistication during usage of the system. For example, when the automation fails at the flow management level, it is possible to shift one level lower to control aircraft on an individual mission level.

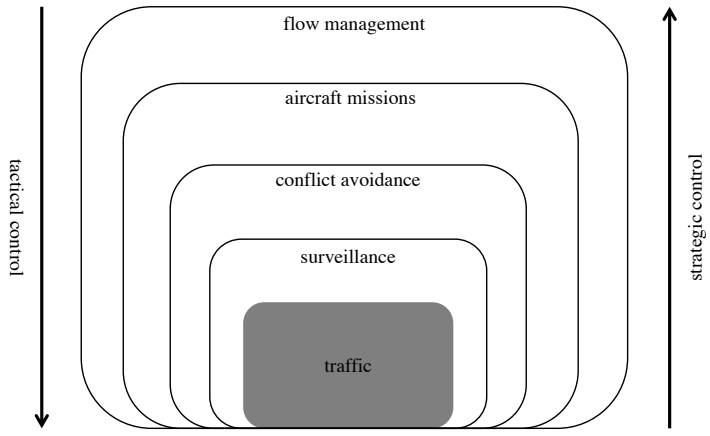


Figure A.4: Levels of control sophistication in the ATC work domain. [Van Paassen et al. 2017 (in press)]

A.4. Path Planning Algorithms

Path planning, or trajectory planning when associated with time [20], is the study of finding a sequence of actions that connect an initial state to a desired goal state. In path planning the location the agent has are the states and the transition between states are the actions the agent can take, each with an associated cost. An path is optimal when it brings the agent from the initial state to the goal state with minimal cost. A path planning algorithm is considered *complete* when it always finds a path between the two states within finite time and notifies the user when no possible path is available [21].

Over the years many path planning algorithms have been thought off, which can be categorized into four main categories: Sampling Based Algorithms, Node Based Optimal Algorithms, Mathematic Model Based Algorithms, Bio-inspired Algorithms [20]. A fifth category is a fusion of combination of algorithms, but is not further considered in this research. An overview of the path planning taxonomy is found in Figure A.5.

The rest of this section is structured such that each category, including its most important algorithms, will be discussed.

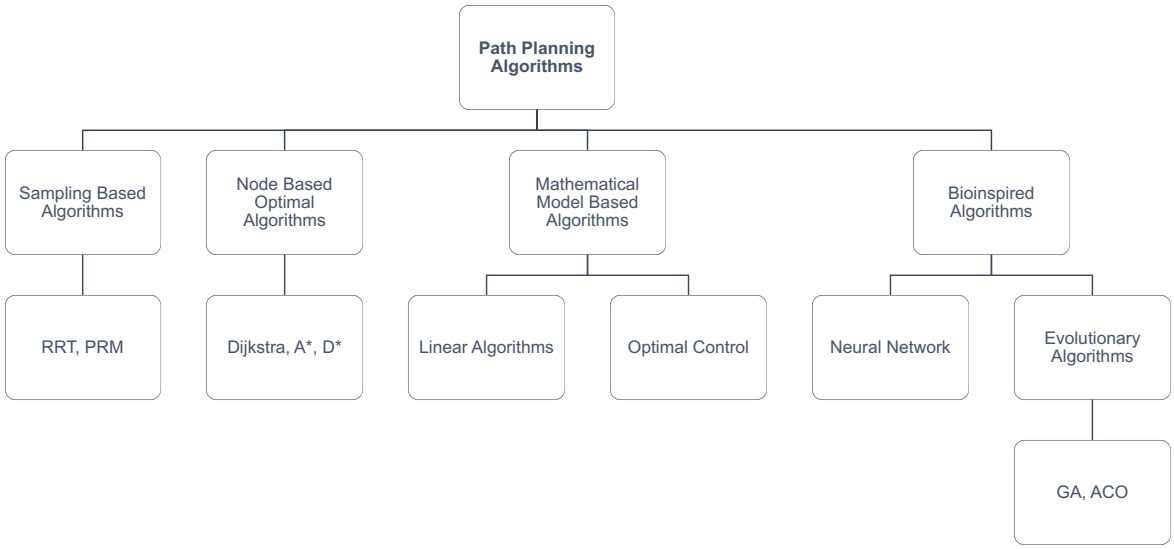


Figure A.5: Path planning algorithm taxonomy (Adapted from [22]).

A.4.1. Sampling Based Algorithms

Sampling based algorithms are algorithms that have a basis in probability, in which they connect points sampled randomly through the space. The sampling based algorithms are not considered complete, because of the randomness, but they provide probabilistic completeness guarantees. This is because the probability that the algorithm fails to provide a solution (if one exists) goes to zero as the number of samples approaches infinity [23].

The two most important sampling based path finding algorithms are the Probabilistic Roadmap (PRM) [24] and the Rapidly-exploring Random Tree (RRT) [25]. Both algorithms are based on the sampling of random points through the state space and connecting these to find an optimal path. However, the ways these two algorithms construct the graph to connect the points are different. For both methods means that each time the algorithm is performed on the same problem a different solution is found.

Probabilistic Roadmap

The PRM algorithm is a multi-query method, that consists out of two phases. In the first phase, the construction phase, a graph (roadmap) is constructed which represents a set of obstacle avoiding trajectories. In the second phase, the query phase, the optimal path from the initial state to the final state is computed through

the graph. An example of how the PRM works, with a simple graph construction algorithm, is found in Figure A.6.

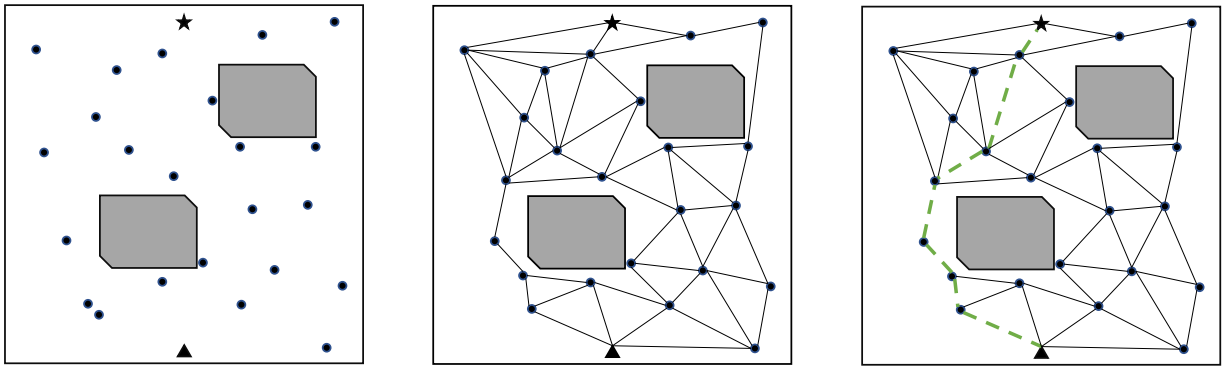


Figure A.6: Three steps in the PRM algorithm, from constructing a roadmap to finding the optimal path. The triangle indicates the starting point and the star the goal.

Rapidly-exploring Random Trees

The RRT algorithm rapidly searches the space by incrementally constructing a tree from the starting state, using random samples from the space. At each step a new node is sampled, and if a path can be drawn from the sampled node to the nearest node, a new node will be added. In Figure A.7 a simplified example of how the RRT algorithm works is found.

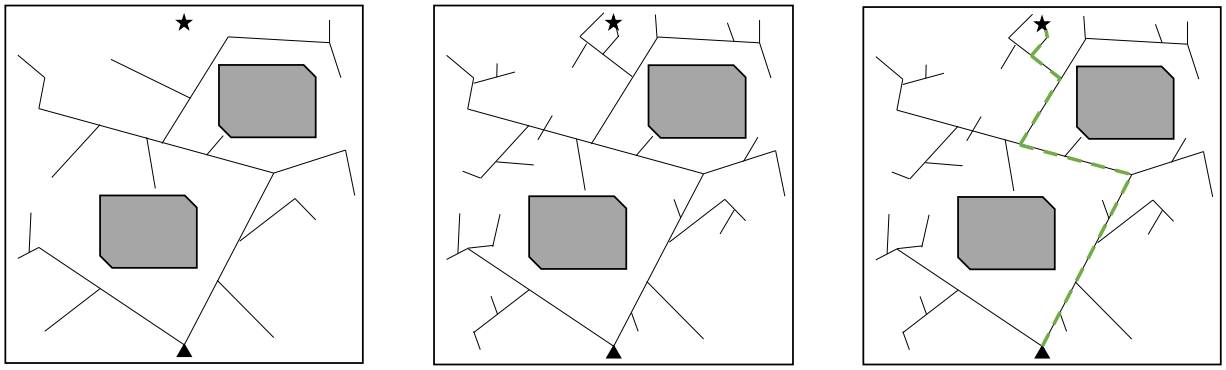


Figure A.7: Construction of a simplified RRT. The triangle indicates the starting point and the star the goal.

A.4.2. Node Based Optimal Algorithms

Node based optimal algorithms explore the space in a discretized manner. These kind of algorithms explore a set of nodes in the space and find the optimal path by calculating the cost while running through these placed nodes. The most well-known shortest path finding algorithm is Dijkstra's algorithm, invented by Edsger W. Dijkstra [26]. Based on Dijkstra's algorithm many other variants have been constructed, such as A*[27, 28] and D*[29].

Each of the node based algorithms is heavily influenced by it settings. An increase in the step size or nodes has an influence on the computation time and the optimality of the solution. In contrast to sampling based algorithms, node based algorithms provide the same solution each time the algorithm is run.

Dijkstra's Algorithm

Dijkstra's algorithm is an exploring algorithm that expands from the starting node outwards [26]. The algorithm depends on dynamic programming and looks at the local path cost to each following node. The

algorithm has no information of where to go and will thus explore everything in the space, until the goal has been reached with creating new nodes and paths. At each node the shortest local path to it is determined, so once the goal is reached the shortest path from destination to the goal is known. The Dijkstra's algorithm will always find the shortest path due to the fact that the whole space is being explored. However, a drawback of this is that the algorithm is slow due to the large amount of calculations that need to be performed. An qualitative example of the Dijkstra's algorithm at work is shown in Figure A.8.

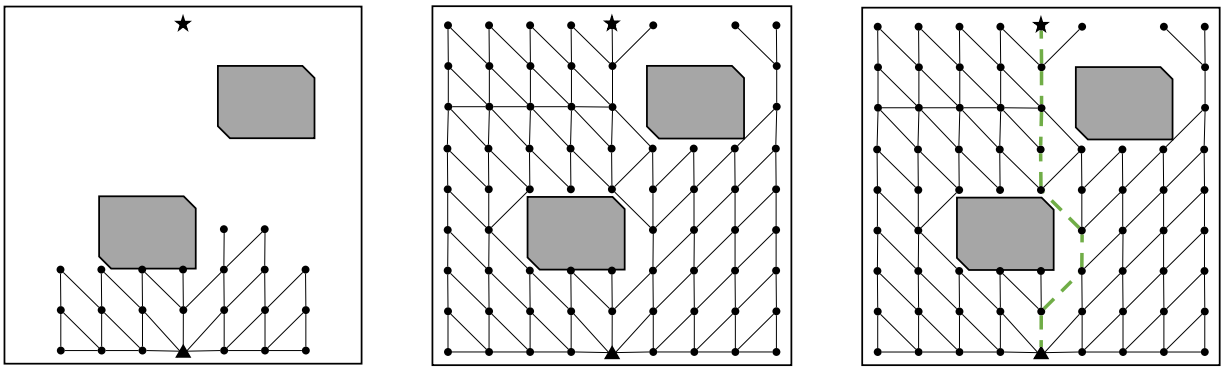


Figure A.8: Three consecutive steps in the Dijkstra algorithm. The triangle indicates the starting point and the star the goal.

A*

The A* algorithm is an extension of the work of Dijkstra's algorithm, with the goal of achieving a higher performance speed performance by means of heuristics [27, 28]. The algorithm uses heuristics to guide the algorithm towards the goal. An example of how the A* algorithm works is found in Figure A.9. As can be seen from the figure, compared to the Dijkstra's algorithm in Figure A.8 it needs much less nodes to find the shortest path and thus will run faster and more efficient than the Dijkstra algorithm.

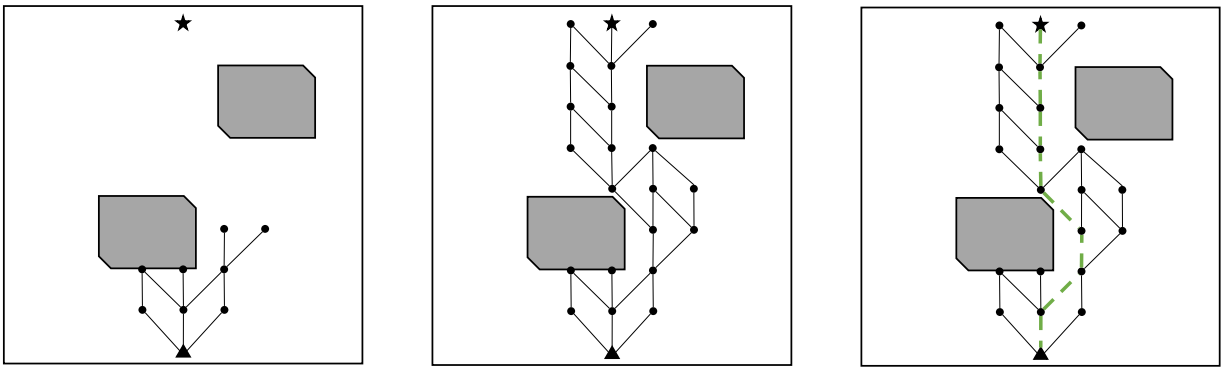


Figure A.9: Three consecutive steps in the A* algorithm. The triangle indicates the starting point and the star the goal.

D*

The D* algorithm is a variant based on the A* algorithm. Its name is derived from Dynamic A* [29]. The algorithm works just as the A* star algorithm, except that it has a better performance when something in the space changes, such as dynamic obstacles changing in real time. Whereas A* will need to do all calculations again, D* has implemented an efficient way to enable searching in dynamic environments.

A.4.3. Mathematical Model Based Algorithms

Mathematical model based algorithms are algorithms that model the environment and the system. The environment is modelled as kinematic constraints and the system as dynamic constraints. To find an optimal solution to the path planning problem, these constraints are added to the cost function as inequalities or

equations. The problem with these algorithms is that they have a complex formulation, because the whole environment and system needs to be modelled, they tend to be computational expensive. The mathematical model based algorithms can be mainly categorized in to two categories: Linear Algorithms and Optimal Control [22].

Linear Algorithms
Linear algorithms are used to model the complete environment, including kinematic and dynamic constraints. These linear algorithms are able to handle the control disturbances and model uncertainty. An example of a linear algorithm, implemented with Mixed-Integer Linear Programming (MILP) is found in Reference [30].

Optimal Control
Optimal control can be seen as an extension to the linear algorithms. In optimal control it is a lot easier to model the uncertainty as linear chance constraints, because of a infinite number of variable conditions. Optimal control can be used to path planning, by posing the problem in the optimal control form. In this form the state and control oriented path can by found on the basis of a set of differential equations [31]. An variant of optimal control has been implemented in Reference [32].

A.4.4. Bio-inspired Algorithms
Bio-inspired algorithms are based on the idea of mimicking nature to find an optimal solution. These algorithms have the ability to solve problems with many variables and non-linearity, something which his hard to do with mathematical model based algorithms. Within the field of bio-inspired algorithms there are two categories: Evolutionary Algorithms and Neural Networks.

Evolutionary Algorithms
Evolutionary algorithms are based, as the name suggests, on the evolution of a species. Evolutionary algorithms are based on stochastic processes. In general they start with selecting randomly feasible solutions as a first generation. From this first generation the fitness of each solution is determined by taking into account the constraints. This leads to the next generation, after which mutation and crossover is applied. This process repeats it self until a solution is found. The two most well known evolutionary algorithms are the Genetic Algorithm [33] and the Ant Colony Optimization Algorithm [34].

Neural Networks
Neural Networks is used as an algorithm to mimic the way that a human brain solves problems. By using a large collection of neural units the algorithm can be self-learning and trained to find the optimal solution. Glasius et al. first introduced neural networks to the path planning and obstacle avoidance problem. [35]

A.5. Trajectory Planning Algorithms in ATM
In Section A.4 the main categories and most important path planning were discussed. These discussed algorithms mainly form the basis of all path planning variants that have been designed throughout the years. In this section the focus will shift to specific trajectory planning algorithms that can be used for ATM. Note that the terminology changed to trajectory planning instead of path planning algorithms, this is because in 4D air traffic management the time factor is key and thus should be called trajectory planning according to the definition in Reference [20].

For the design of a shared human-automation representation for flow-based perturbation management an algorithm is necessary that a human can understand and which choices for certain trajectories can be explained, as discussed in Section A.1. From this it is quite clear that an algorithm should be chosen which is based on Node Based Optimal Algorithms, since the other algorithms are either based on stochastic principles or are simply too complex to grasp by a human. Node based algorithms can be programmed such that the human will be able to understand its choices. Another reasons to pick an node based optimal algorithm is that the computational power necessary can be kept relatively low.

An interesting algorithm however, which is not a node based algorithm but an linear mathematical model based algorithm, is Supervision of Route Optimization (SUPEROPT) [36, 37]. This algorithm has been devel-

oped as part of SESAR workpackage E and uses the Mixed-Integer Linear Programming approach. SUPEROPT allows the ATCo to influence the trajectory planning algorithm by defining the sense of the conflict resolution or by specifying which aircraft passes ahead of another.

Another approach has been developed by Idris et al. and is known as the Trajectory Flexibility Metric [38–40]. With this metric Idris aims to preserve the airspace flexibility by trajectory planning. The idea behind the concept is that by implementing trajectory flexibility on individual aircraft, the traffic complexity of the whole airspace can be maintained on acceptable levels. The approach is based on a node based optimal algorithm and discretises the space in position and time to account for dynamic objects. The algorithm is constructed such that it is not only possible to optimize for shortest path, but also for the metrics robustness and adaptability, which are defined as follows:

> *"**Robustness** is defined as the ability of the aircraft to keep its planned trajectory unchanged in response to the occurrence of disturbances, for example, no matter which trajectory or conflict instances materialize."*[39]

> *"**Adaptability** is defined as the ability of the aircraft to change its planned trajectory in response to the occurrence of a disturbance that renders the current planned trajectory infeasible."[39]*

Using robustness and adaptability should lead to a more human-like way of controlling the airspace, as professional ATCos also apply them in their control strategies.

A.6. Discussion Best Trajectory Planning Algorithm

As could be seen in Sections A.4 and A.5, there are many forms of path planning algorithms, which in theory could all be used for this research. The goal is this research is however to design an interface in which the controller can easily understand the algorithm's choices and influence the choices of the algorithm, as explained in Section A.1. Also, as stated by Billings, the automation needs to be predictable and thus be consistent in the solution it provides [15].

Because of this predictability constraint the sampling based algorithms can directly be considered as infeasible. Sampling based algorithms, as the name suggests, are highly random and provide a different solution each time. Also bio-inspired algorithms can be considered as infeasible, due to their stochastic properties.

Mathematical model based algorithms can be considered infeasible for this research due to their complexity and computational expense. This leaves us with node based algorithms. Each of the base algorithms discussed in Subsection A.4.2 could be feasible. However, each of these algorithms needs quite some modification to fit into the ATM environment. For example, time and moving objects need to be incorporated into the algorithm. For the algorithms in Section A.5 this is already the case. From the two algorithms considered in this section, the Trajectory Flexibility Metric by Idris et al. was considered the best for this research for the following reasons:

- The algorithm is transparent in its workings and has the ability to make the same kind of control choices as a human controller.
- The algorithm's settings are easily controllable (such as number of waypoints and multi-dimensional cost function) and adjustable to the aircraft performance envelopes.
- Built-in implementation of robustness and adaptability.
- Overlap with the TSR in its constraints.

A.7. Implementation Trajectory Flexiblity Metric Idris

This section will elaborate on the how the trajectory flexibility metric exactly works and how it is implemented in the TSR software. The algorithm is a node based optimal algorithm and works in 2D space, with the addition of time. The algorithm's main workings can be divided into four steps:

1. Define metric grid, based on grid strategy, number of segments, state change strategy and obstacle list.

2. Perform forward propagation to determine all reachable cells.

3. Perform backward propagation to determine all feasible cells and branches.

4. Find best trajectory based on the cost function.

If no solution is found in step 4, the process starts again but with an increase in the number of segments by one. In the next four subsections each step will be explained in detail.

A.7.1. Define Metric Grid

In the first step the metric is defined, which consists of four different parts. The first part is the grid strategy, in which is defined how the space is discretized in 2D. An illustration of this is found in Figure A.10, where the start point is the current position of the aircraft and the end point is where the aircraft leaves the airspace. For this illustration a linear grid space of 11 by 11 cells is used, for the implementation however a much finer grid of 101 by 101 cells is used. Next to this it would also be possible to implement a non-linear grid, to get for example a finer grid in the middle of the grid and more coarse on the outer edges. This is however left for further research.

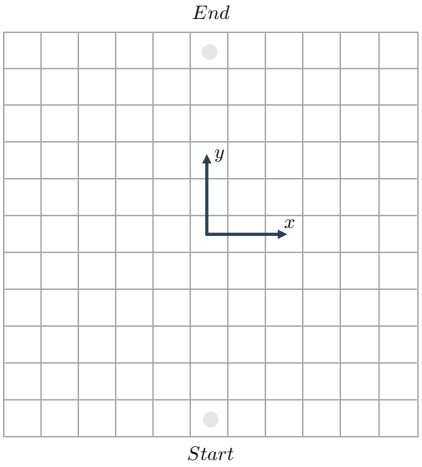


Figure A.10: Grid strategy: discretize the space in 2D.

In the second part the 2D grid needs to be converted to 3D, where the third dimension is time. This is done by discretizing the time from $T = 0$ to $T = RTA$ in to a number of segments, an illustration is found in Figure A.11. As can be seen the time is discretized in to equal time steps ϵ, but this could also be changed to a non-linear variant. The points in between segments can be considered as the waypoints the aircraft will fly through. By discretizing the time in this manner, it must be assumed that the speed and heading changes at the waypoints are instantaneous and that the speed and heading will stay constant in between waypoints.

The third part of the definition of the metric grid is the choice of a state change strategy. The state change strategy is based on a heading and a speed strategy. In the heading strategy the minimum heading angle Ψ_{min} and maximum heading angle Ψ_{max} of the aircraft are defined. To complete the strategy a choice need to be made of in how large steps the heading can be changed, $\Delta\Psi$. The speed strategy is build up in the same way, with a minimum V_{min} and a maximum speed V_{max} and the possible changes in speed ΔV.

Finally, the fourth part is to add the obstacles in the airspace to the metric grid. Obstacles can be stationary, such as no-fly zones or weather cells, or dynamic, such as other aircraft. Stationary obstacles are blocked by marking them in the metric grid on the same cells in every time step. Dynamic obstacles move through time, and therefore the blocked cells differ in each time step. An example of a dynamic obstacle moving through the grids is found in Figure A.13.

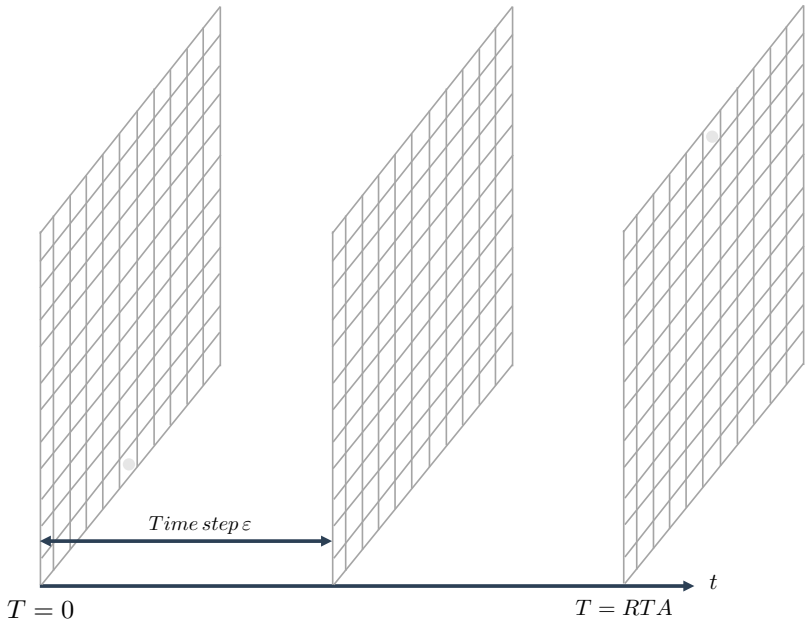


Figure A.11: Convert grid space to 3D to include the time variable.

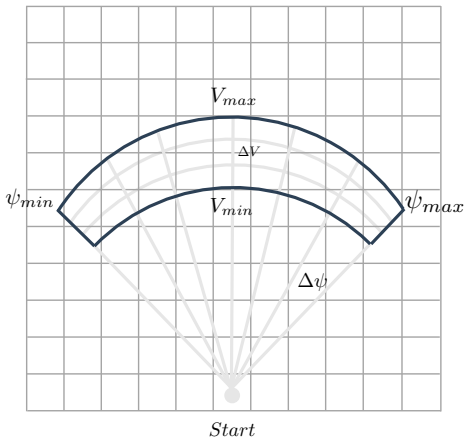


Figure A.12: State change strategy

A.7.2. Forward Propagation

With the metric grid fully defined, it is now possible to determine all reachable cells, i.e. all cells which the aircraft can reach, taking into account the heading and speed constraints. From the center of the start cell all reachable branches to the next 2D grid are drawn. From the center of the cells reached in the second 2D grid again all reachable branches are drawn. This process continues until the last 2D grid. Once this process is complete, a set of all reachable trajectories from the starting cell is known. An example of this process can be found in Figure A.14.

A.7.3. Backward Propagation

From the set with all reachable trajectories it is now necessary to determine all feasible trajectories. A trajectory is considered feasible when it reaches the end cell and does not cross any obstacles. To determine the feasible trajectories a backward propagation is performed, starting from the end cell. From here all branches to the next 2D grid, that do not lead to an object, are marked as feasible. From the center of these feasible cells

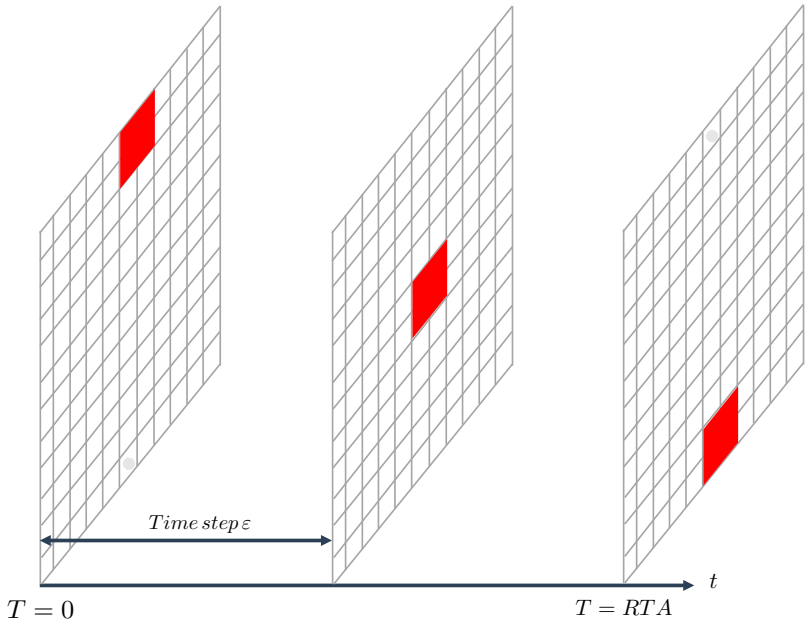


Figure A.13: Dynamic obstacle moving through the 3D grid.

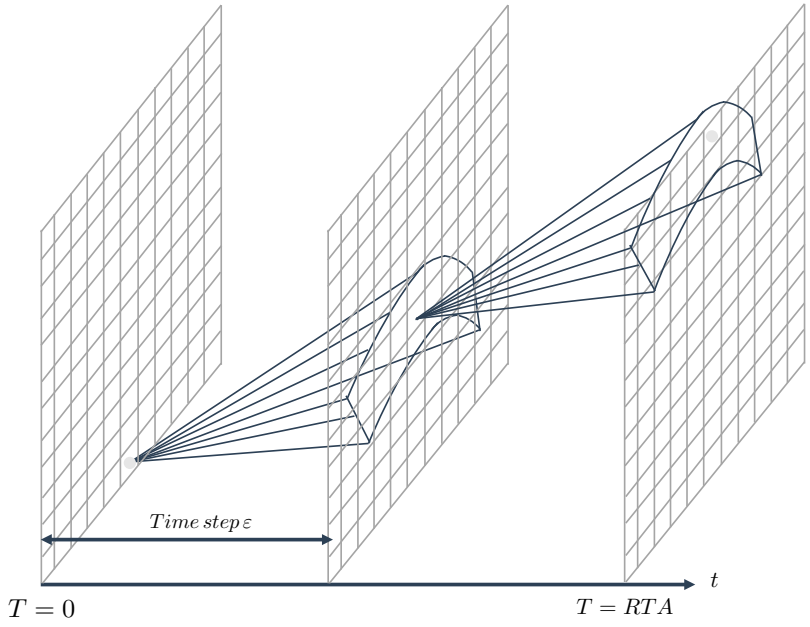


Figure A.14: Forward propagation to determine all reachable cells.

the process is again repeated until the start cell is met. The result is a complete set of feasible trajectories.

A.7.4. Find Optimal Trajectory

From the set of feasible trajectories the most optimal trajectory now needs to be selected. For this research four different cost functions have been defined, based on the work of Idris et al [40]. These cost functions are used to optimize for shortest path, adaptability, robustness, or a combination of the three. The cost functions

are defined as follows:

$$Shortest\ path = SP = Minimize(Distance) \tag{A.1}$$

$$Adaptability = ADP = Maximize(Reachable\ trajectories) \tag{A.2}$$

$$Robustness = RBT = Maximize(\frac{Feasible\ trajectories}{Reachable\ trajectories}) \tag{A.3}$$

$$Combination = Maximize(-C_{SP} * SP + C_{ADP} * ADP + C_{RBT} * RBT) \tag{A.4}$$

Where in Equation A.4 C_{SP}, C_{ADP} & C_{RBT} are weights that trade off shortest path, adaptability and robustness.

B

Experiment Briefing

Thank you for participating in this experiment. The goal of this experiment is to investigate the influence of human control on a trajectory planning algorithm in an 4D (space and time) trajectory based ATC perturbation management task.

In future air traffic control, all flight trajectories are planned way ahead of the actual flight with 4D trajectory-based operations (TBO). With the addition of time, this means aircraft not only need to follow certain waypoints, but also reach these waypoints at a certain time. All flight plans are de-conflicted beforehand, however a perturbation such as a weather cell, no-fly zone or an aircraft in distress could occur in an airspace, which means all trajectories in that airspace need to be rerouted. This research investigates a method to re-plan these trajectories automatically by means of a trajectory planning algorithm, but allow a human controller to influence the algorithm. Figure B.1 shows an example of a perturbation (no-fly zone) in an airspace, in which it is required to reroute the airways.

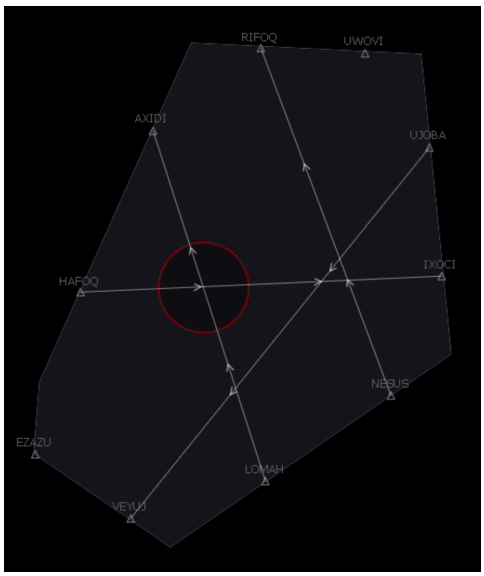


Figure B.1: Example of a perturbation (no-fly zone) in an airspace.

B.1. The Experiment

In this experiment, it is your task to steer the trajectory planning algorithm by means of constraining the solution space of the algorithm to cope with the perturbation. You will be presented several scenarios in which the de-conflicted airspace is perturbed by a no-fly zone. You will be asked to apply a structure to the airspace, which you think will lead to the most optimal solution. Only after you indicate that your happy with the structure, the actual flight scenario will start to show you the results of your choices.

Consider the following criteria during the execution of the task:

- Safety buffers
- Additional trajectory length
- Airspace structure

You are free to give a weighting to these interdependent criteria yourself.

The experiment will start with several training scenarios to familiarize yourself with the control task and interface. Please make sure that you ask all questions that you have in relation to the task and interface during the training scenarios. Once the actual experiment begins I cannot answer any questions you have.

B.2. Timeline

Table B.1 shows the timeline of the experiment.

Table B.1: Timeline of the experiment

Activity	**Estimated duration**
Introduction to the experiment	5 min
Training	30 - 40 min
Measured runs	6 x 8 min
Debriefing	5 - 10 min

In total the experiment will take around 1.5 hours.

B.3. Experiment Set-up

The experiment will be conducted in the Air Traffic Management Laboratory (ATMLab), on the second floor of the SIMONA-building. This lab uses a 30-inch LCD screen on which the interface is shown, which provides a top-down view of the airspace and traffic. Input can be given by a standard mouse and keyboard input devices.

The following mouse and keyboard inputs are available to interact with the interface:

- *LMB click:* select constraint / airway / draw constraint
- *RMB click:* accept, cancel, delete constraint
- *Backspace:* deselect airways
- *A:* re-plan selected aircraft trajectory

The following colors are used in the interface:

- *Red:* no fly-zone, air traffic should be guided around this. Also used when an airway is fully blocked by a constraint
- *Cyan:* color used to indicate the airway's solution space
- *Magenta:* constraints drawn by human controller, automation will not cross these areas
- *Grey:* Airways

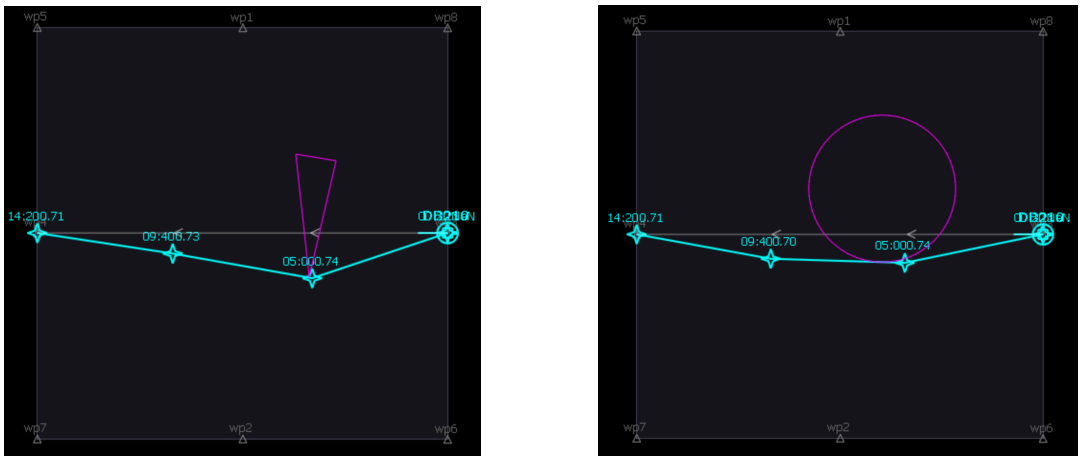
B.4. Training

Before the actual experiment will start, you will go through a number of training scenarios to familiarize yourself with the interface. As stated before, please ask all the questions you have regarding the interface during the training, make sure you feel fully comfortable in using the interface.

Training 1 - Interface familiarization part 1

The interface shows all airways on which air traffic will enter the airspace. You will perform the task of constraining the automation before any traffic enters the airspace.

1. Shown is an airspace with one airway, where the two arrows indicate the direction of the airway. On this airway aircraft will enter once the scenario is started.
2. In the top bar a magenta polygon and circle are shown, these are constraints you can draw to steer the trajectory planning algorithm.
3. Draw a polygon constraint, accept with RMB. See Figure B.2a.
4. Now the experimenter will let the aircraft enter the airspace, assess the result.
5. Draw a circle constraint, accept with RMB. See Figure B.2b.
6. Delete both constraints.



(a) (b)

Figure B.2: Two examples of steering the trajectory planning algorithm around self-drawn constraints

Training 2 - Interface familiarization part 2

1. Shown is an airspace with one airway, where the two arrows indicate the direction of the airway. On this airway aircraft will enter once the scenario is started.
2. Hovering over or selecting an airway shows the solution space, also seen in Figure B.3, in which the automation is able to find solutions for possible new trajectories of aircraft entering the airspace, while still maintaining the required time of arrival at the exit waypoint. This experiment is restricted to new trajectories with two additional waypoints, which is indicated by the two curved areas, which are the possible areas in which the waypoints will be placed by the automation.
3. Draw constraints to guide the algorithm through the top half of the airspace.
4. The solution space can guide you in drawing the constraints.

These were the basic functionalities of the interface. In the next training will be shown what these will be used for.

Training 3 - Perturbations

1. Shown is an airspace with one airway and a no-fly zone indicated in red.
2. The automation does not see the no-fly zone, so it will be your task to draw a circle constraint around the no-fly zone.
3. Assess the result of the automation.

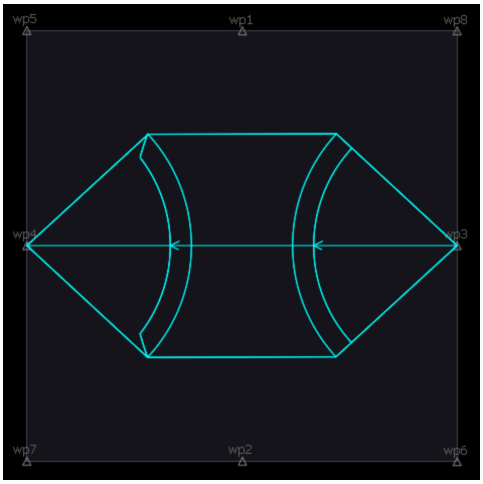


Figure B.3: Solution space trajectory algorithm for two waypoints.

Training 4 - Multiple airways

1. Shown is an airspace with two airways and a no-fly zone.
2. Apply the circle constraint as in the previous scenario.
3. The experimenter will start the scenario to show you the result of the automation.
4. For the horizontal airway, now draw a constraint to force the algorithm to find a solution the other side of the no-fly zone. Do this by drawing the constraint as big as indicated in Figure B.4. As you can see the vertical airway now highlights in red as it is fully blocked by the constraint and is thus won't able to find a solution.
5. Delete the constraint
6. Now draw a constraint that will guide the trajectory the other side, without blocking the other airway.

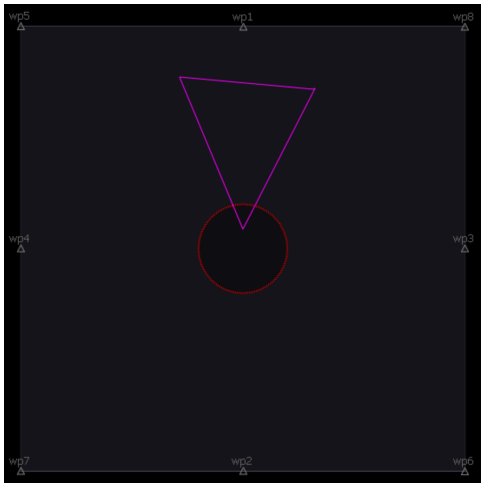


Figure B.4: Assignment training 4

Training 5 - Optimize goals

Please apply a structure to this airspace, consider the goals as stated in Chapter 1.

Why did you make these choices?

Training 6 - Multiple strategies, same goal

You are asked to apply a structure to this airspace with different control strategies, which all lead to the same result.

Try to come up with three different strategies yourself to guide the horizontal east-west airway below. After this the experimenter will show you his options.

Training 7 - Full scenario

Now you are familiarized with the interface and know how you can influence the algorithm it is time to do a full scenario.

Apply a structure to the scenario as you see fit. Notify the experimenter once you are satisfied with your applied structure.

Please note that all traffic is deconflicted beforehand and that conflicts that will be induced by the rerouting of the trajectories are automatically solved.

Training 8 - Full scenario

Apply a structure to the scenario as you see fit. Notify the experimenter once you are satisfied with your applied structure.

You are now ready to participate in the actual experiment. If you still have any questions, this is your last chance to ask them.

Good luck!

C

Analysis Placement Perturbation

This chapter investigates the effect of the placement of a perturbation in an airspace. For this analysis a circular perturbation with a radius of 25 NM is chosen. The focus of this research is on a structured type of airspace.

The airspace and the locations of the perturbations that are analysed are shown in Figure C.1.

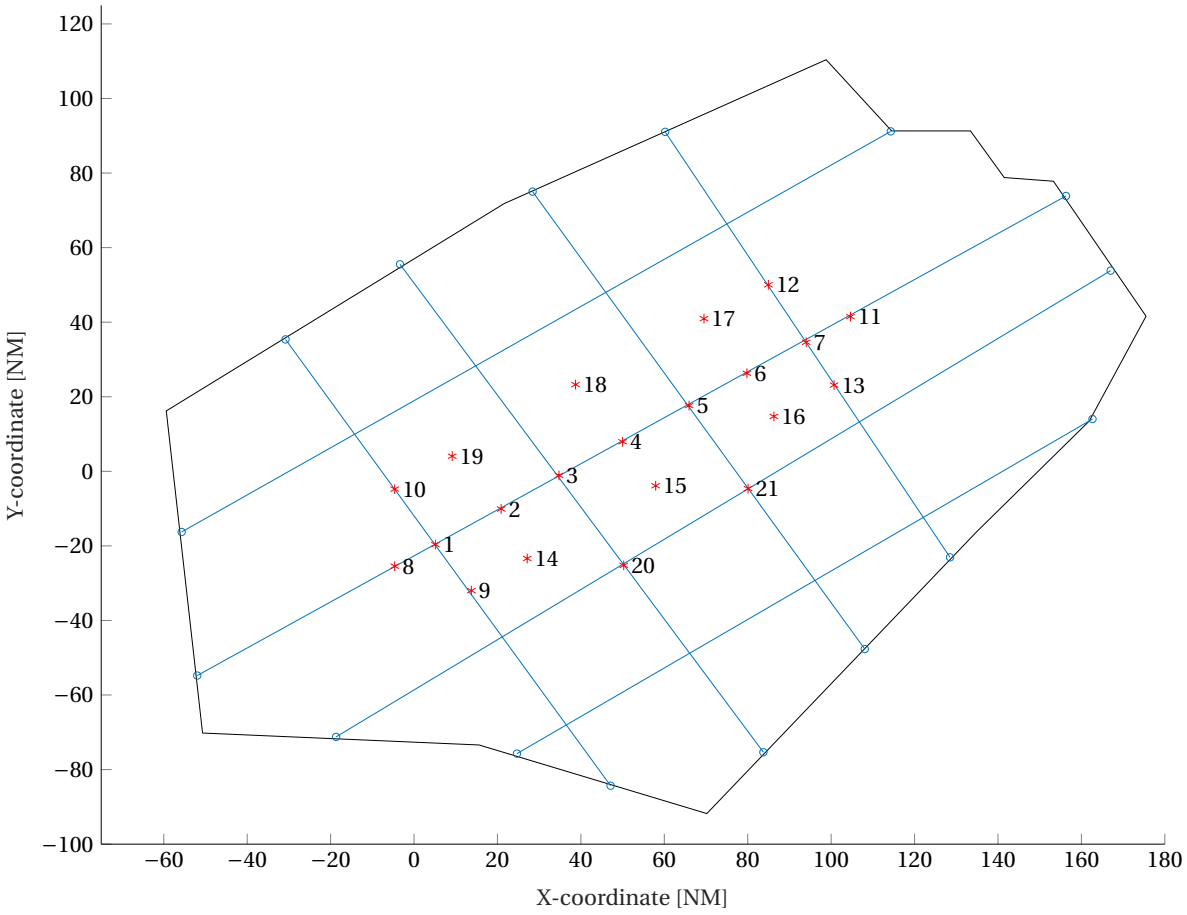
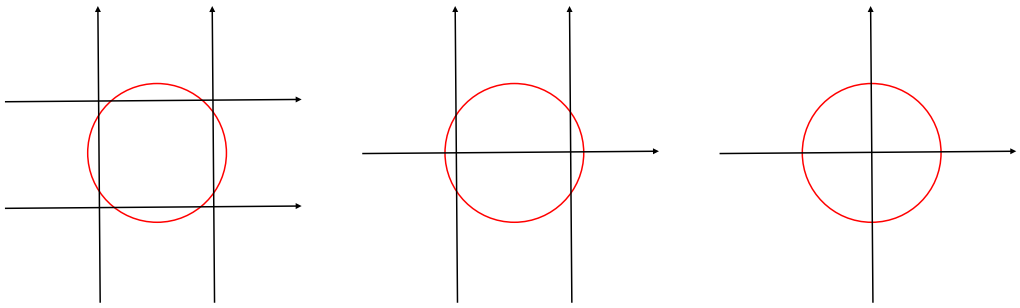


Figure C.1: Locations of all analysed perturbations.

The perturbations are placed in to three categories, as shown in Figure C.2. Category 0 has no airways intersecting the perturbation through the center, category 1 has one airway and category 2 has two airways intersecting through the center.

In Table C.1 it is shown which perturbation falls in to which category.



(a) PL0 (b) PL1 (c) PL2

Figure C.2: Three categories in to which the perturbations are placed.

Table C.1: Perturbations per category

Category	**Perturbation**
PL0	14, 15, 16, 17, 18, 19
PL1	2, 4, 6, 8, 9, 10, 11, 12, 13
PL2	1, 3, 5, 7, 20, 21

The reason why these categories where chosen is because it is hypothesized that for each category the influence a human operator can have differs. As one can imagine for category 0 the human will not make much different choices than the automation as it is planning around this. For category 2 however, the human can make the choice to guide the airway around either side of the circular perturbation.

In Section C.1 an overview of all data is given that was found by running the scenario in full automation mode for the high traffic density scenario. In the sections following the data is focussed per category and the most interesting perturbation location selected. This selected perturbation is then used in the human-in-the-loop experiment to investigate whether the human is able to find a more optimal solution.

The metrics investigated to find the most interesting locations are the following:

1. Added track miles.
2. Knock-on reroutes
3. Average sector robustness.
4. Minimum sector robustness.

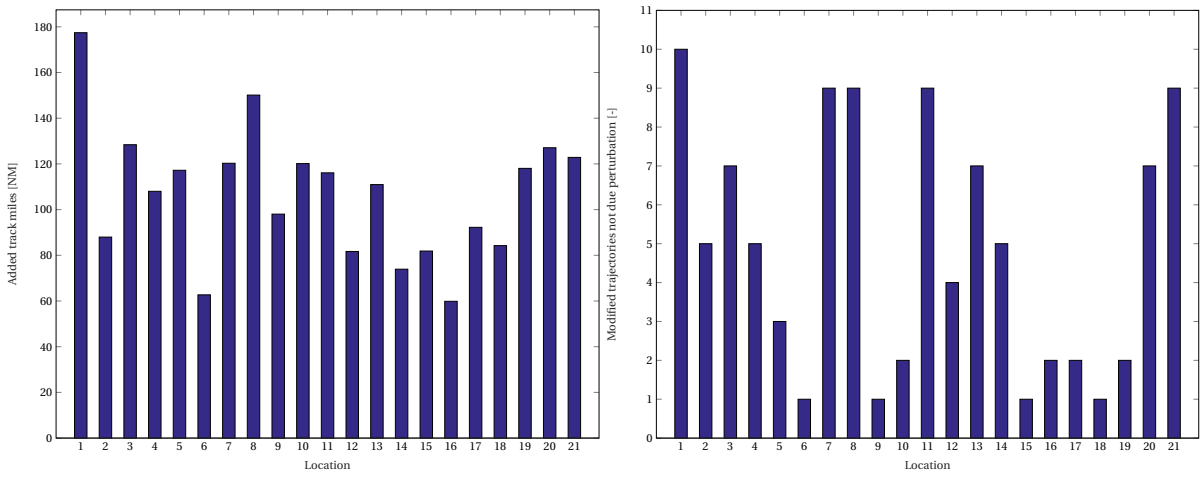
Where the knock-on reroutes are defined as aircraft that need to deviate from their original trajectory due to conflicting with other aircraft and not due to the perturbation.

To find the most interesting location it is tried to find one with a relatively high added track miles and rerouted trajectories. For robustness a relatively low value is most interesting.

C.1. Overview data

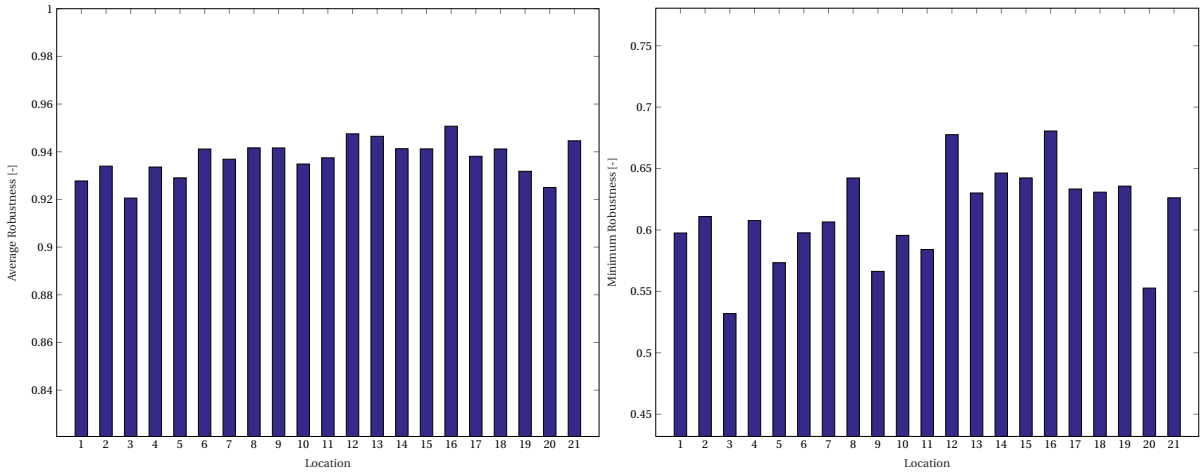
This section summarizes all data collected.

Interesting to see are the averages for each metric per category in Table C.2. For category 0 the track miles and modified trajectories are relatively low and the robustness high. For category 2 however, the track miles and modified trajectories are relatively high and the robustness low. Which could suggest that for category 2 the human can have a larger positive influence on the outcome of the trajectory algorithm.



(a) (b)

Figure C.3: (a) Added track miles per perturbation location. (b) Knock-on reroutes per perturbation location.



(a) (b)

Figure C.4: (a) Average sector robustness per perturbation location. (b) Minimum sector robustness per perturbation location.

Table C.2: Averages of each metric per category

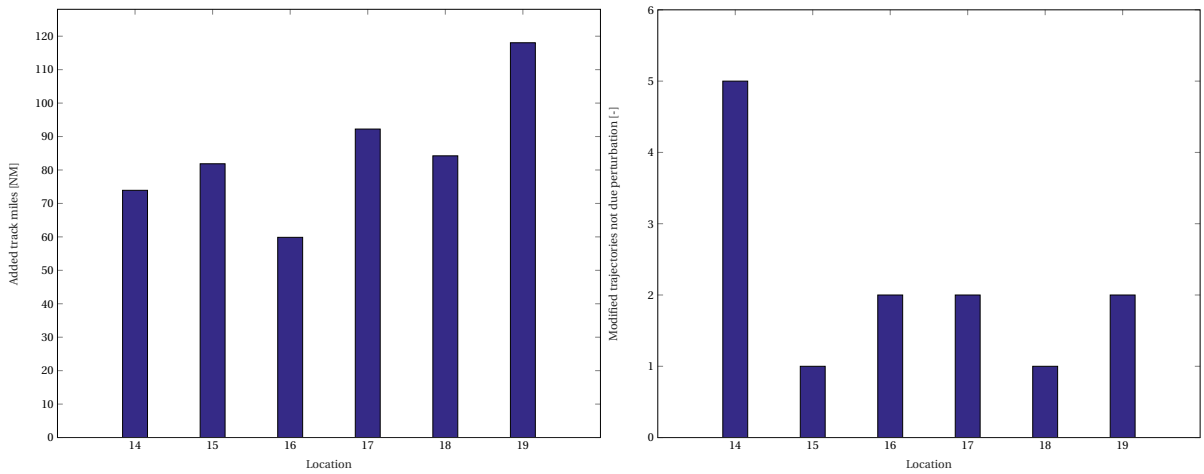
	Cat 0	**Cat 1**	**Cat 2**
Added track miles [NM]	85.0	103.9	132.2
Modified trajectories [-]	2.1667	4.7778	7.5
Avg sector robustness [-]	0.9407	0.9398	0.9306
Min sector robustness [-]	0.6448	0.6125	0.5814

C.2. Category 0 - no center intersections

This section shows the data for category 0. By analysing the data and scenarios perturbation number 14 was chosen for the experiment because of the following reasons:

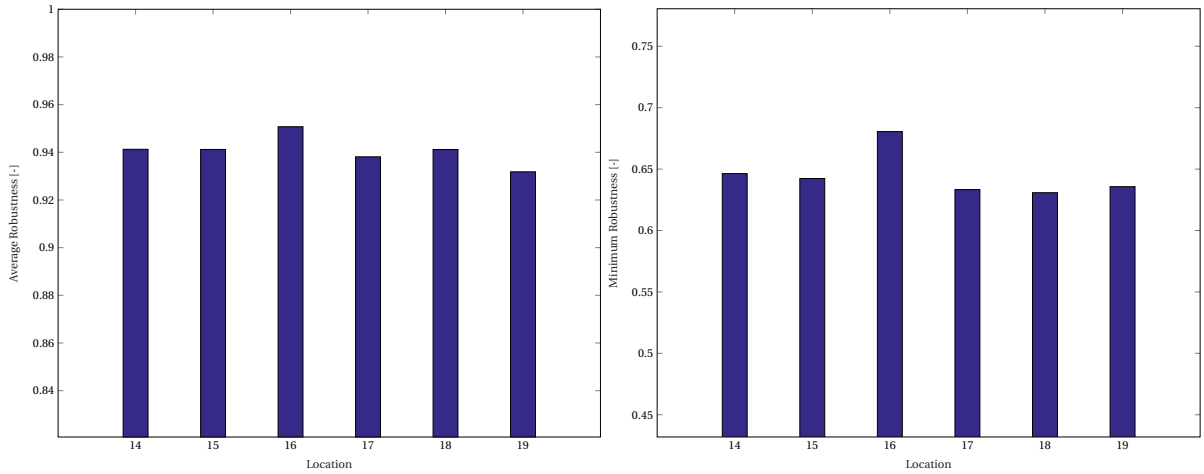
- Looking at the track miles the most obvious choice would be location 19. However, the placement of this scenario lead to some traffic situations which bias the data and thus can be seen as an outlier.
- Following the knock-on reroutes trajectories graph, location 14 is the most interesting.
- From the average sector robustness data no clear conclusion can be drawn due to the small differences between the locations.
- As for the average sector robustness, also for the minimum robustness there is no location that stands out. Difference are small, except for location 16.

Following that location 19 is a bit flawed and the difference for sector robustness are small perturbation location 14 is chosen.



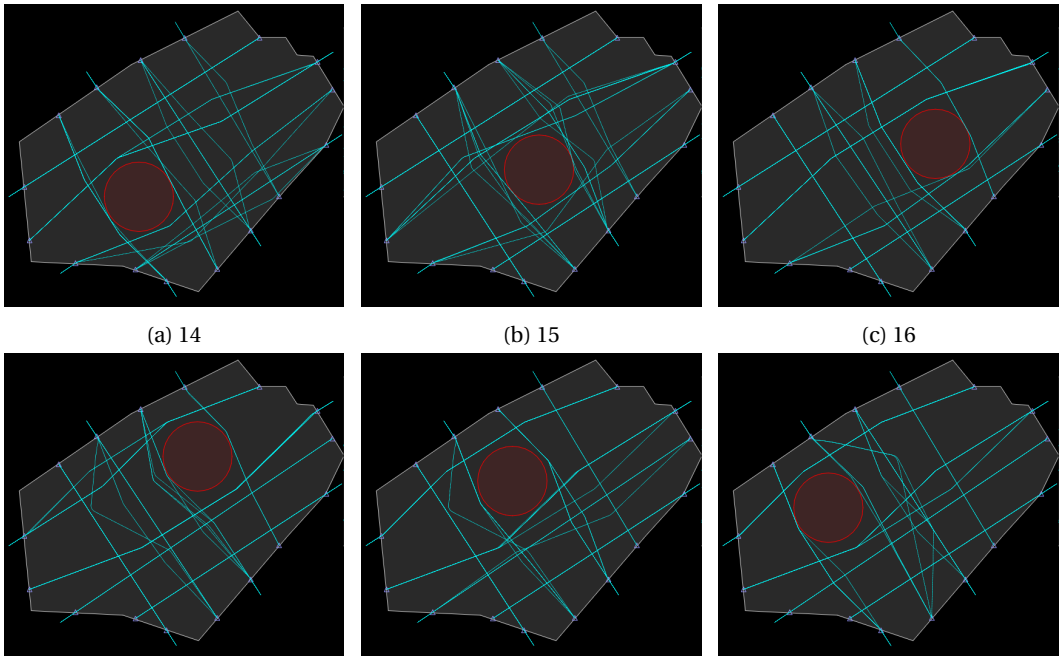
(a) (b)

Figure C.5: (a) Added track miles per perturbation location. (b) Knock-on reroutes per perturbation location.



(a) (b)

Figure C.6: (a) Flexibility metric average robustness per perturbation location. (b) Flexibility metric minimum robustness per perturbation location.



(a) 14 (b) 15 (c) 16

(d) 17 (e) 18 (f) 19

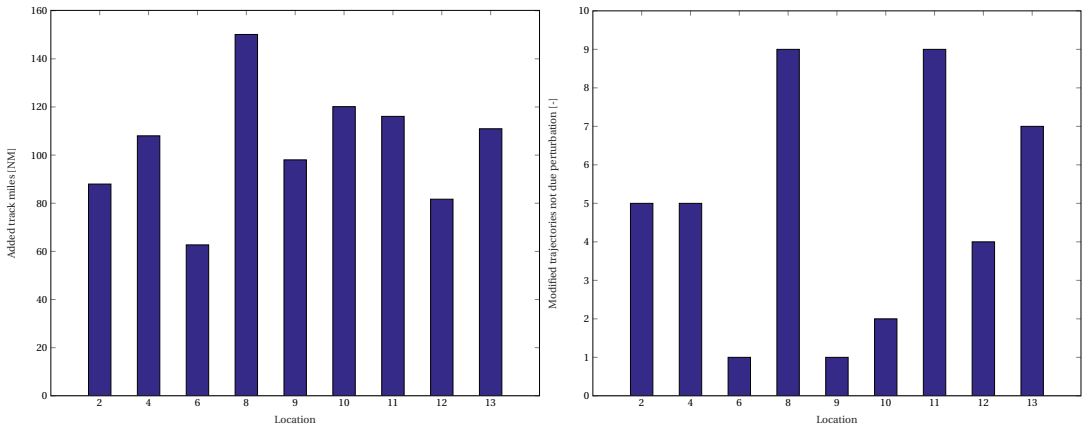
Figure C.7: Results trajectories full automation

C.3. Category 1 - one center intersection

This section shows the data for category 1. By analysing the data and scenarios perturbation number 11 was chosen for the experiment because of the following reasons:

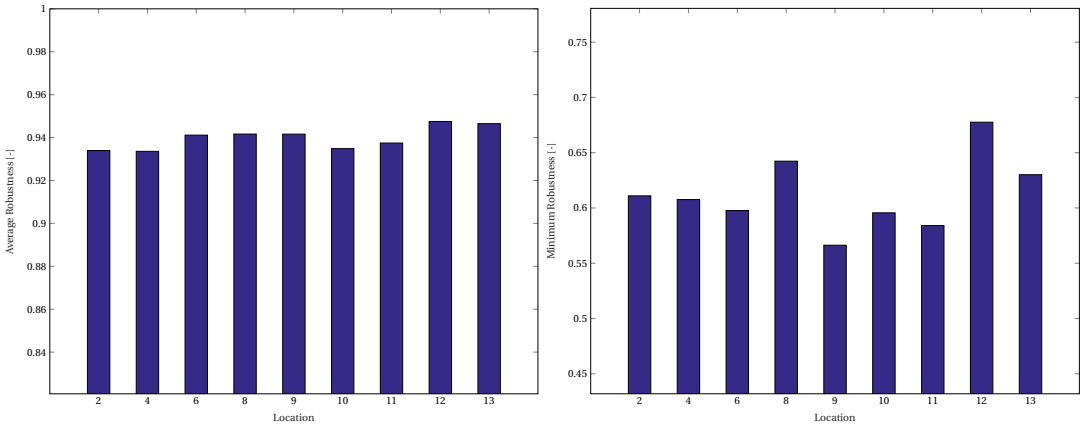
- Track miles indicates location 8 as the most interesting and 6 and 12 as the least interesting.
- Knock-on reroutes shows location as 8 and 11 as most interesting.
- Average sector robustness only shows small differences between the locations
- Minimum sector robustness indicates location 9 and 11 as most interesting.

Location 11 is chosen because it has the most interesting combination of values, especially with the low minimum robustness and a high value of knock-on reroutes.



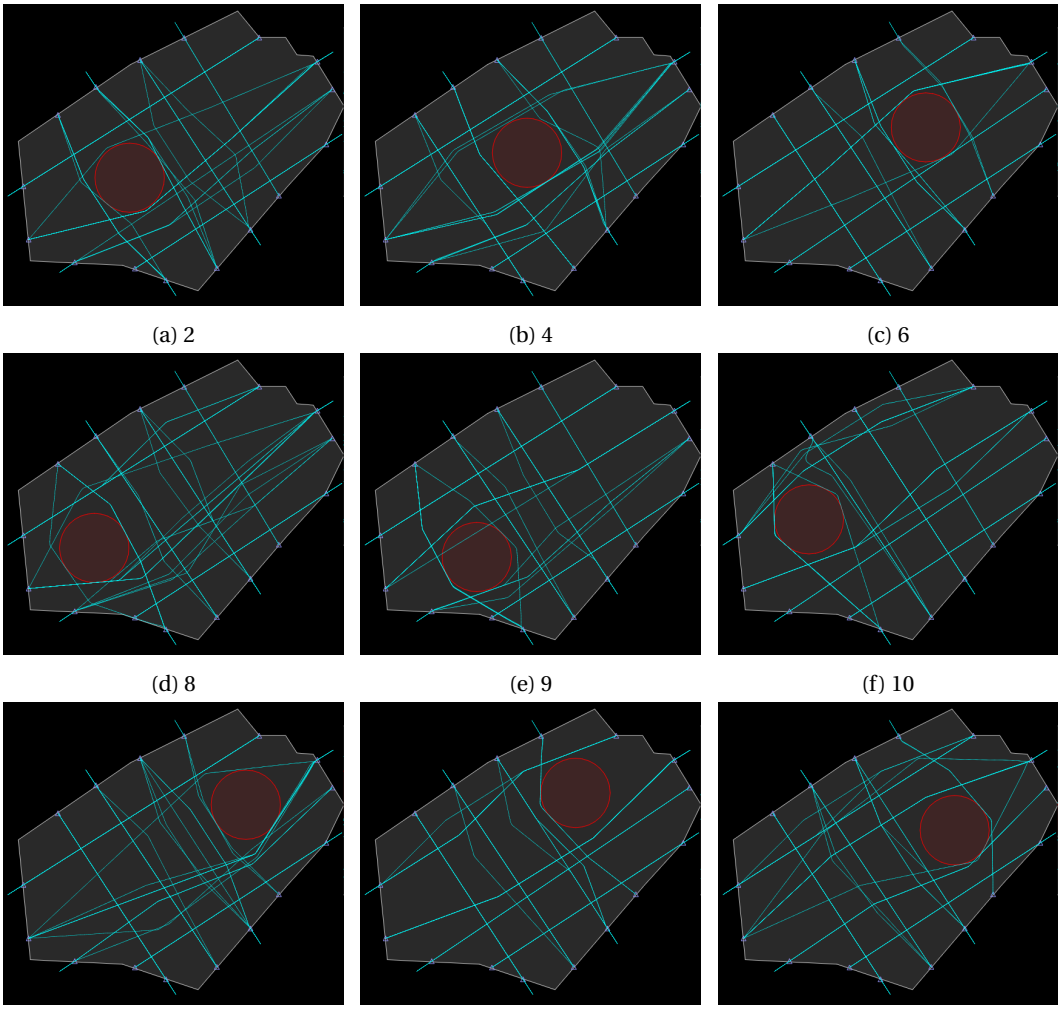
(a) (b)

Figure C.8: (a) Added track miles per perturbation location. (b) Knock-on reroutes per perturbation location.



(a) (b)

Figure C.9: (a) Flexibility metric average robustness per perturbation location. (b) Flexibility metric minimum robustness per perturbation location.



(a) 2 (b) 4 (c) 6

(d) 8 (e) 9 (f) 10

(g) 11 (h) 12 (i) 13

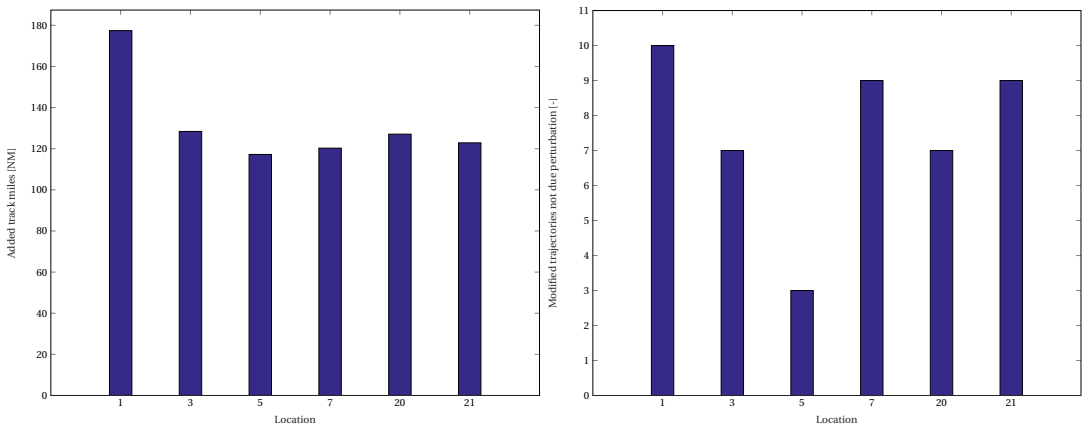
Figure C.10: Results trajectories full automation

C.4. Category 2 - two center intersections

This section shows the data for category 2. By analysing the data and scenarios perturbation number 1 was chosen for the experiment because of the following reasons:

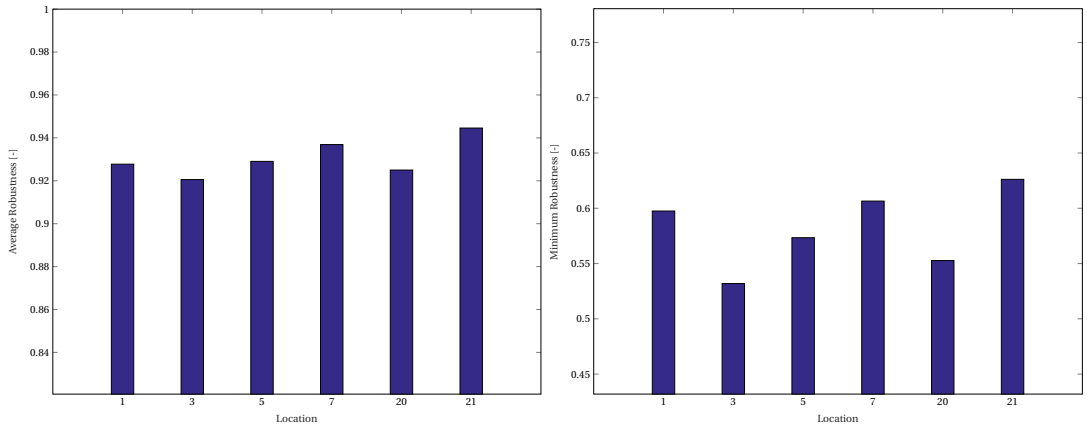
- Location 1 is by far the most interesting in terms of track miles.
- Location 1, closely followed by 7 and 21, are most interesting in terms of knock-on reroutes.
- Average sector robustness does not show large differences, but 3 and 20 seem slightly more interesting.
- Location 3 and 20 are more interesting for the minimum sector robustness.

Although quite some locations are interesting in this category, it is chosen for 1. Mainly, because it is expected that the track miles and knock-on reroutes can be improved substantially.



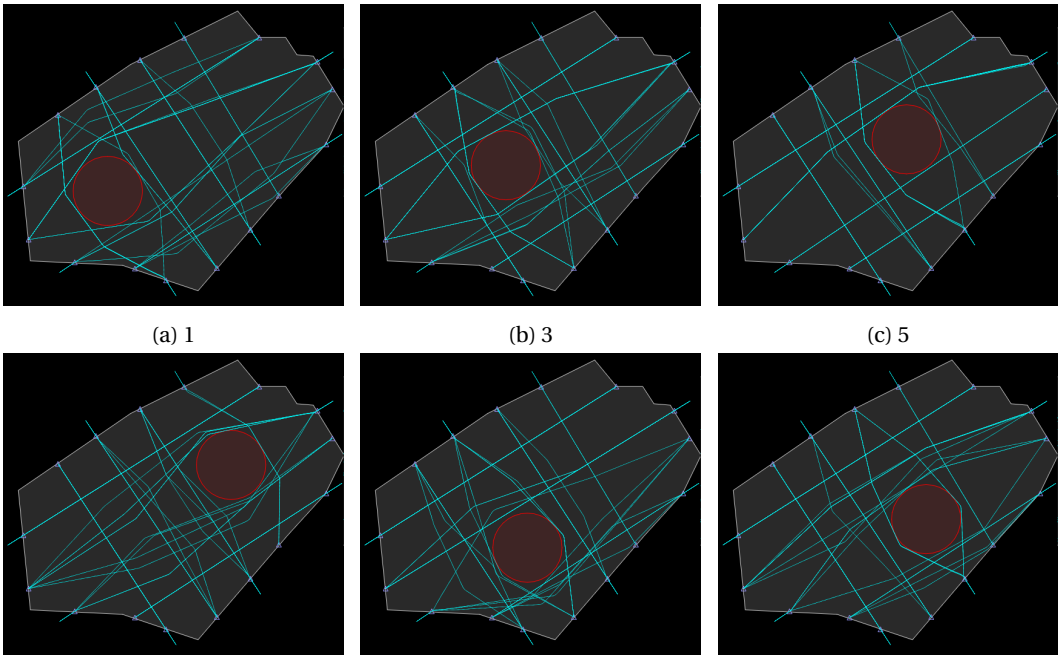
(a) (b)

Figure C.11: (a) Added track miles per perturbation location. (b) Knock-on reroutes per perturbation location.



(a) (b)

Figure C.12: (a) Flexibility metric average robustness per perturbation location. (b) Flexibility metric minimum robustness per perturbation location.



(a) 1 (b) 3 (c) 5

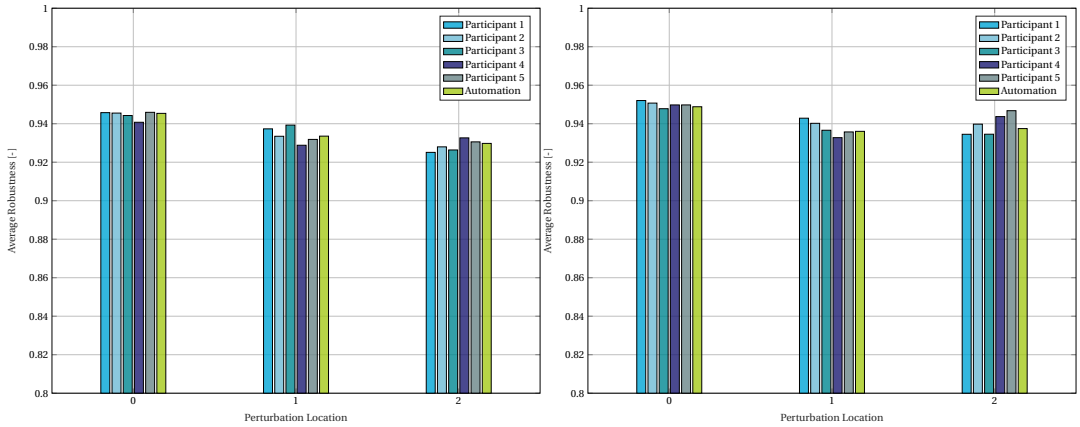
(d) 7 (e) 20 (f) 21

Figure C.13: Results trajectories full automation

D

Additional data

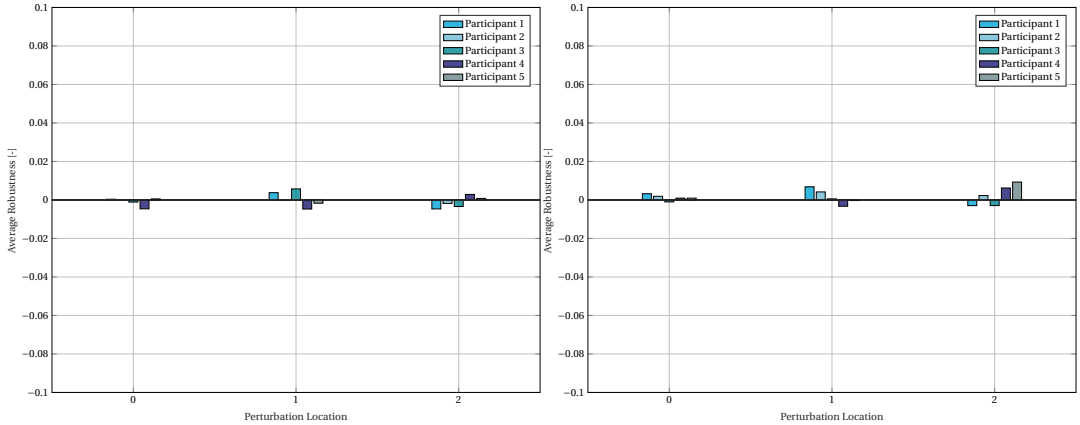
D.1. Metrics averaged over perturbation locations



(a) High traffic density

(b) Low traffic density

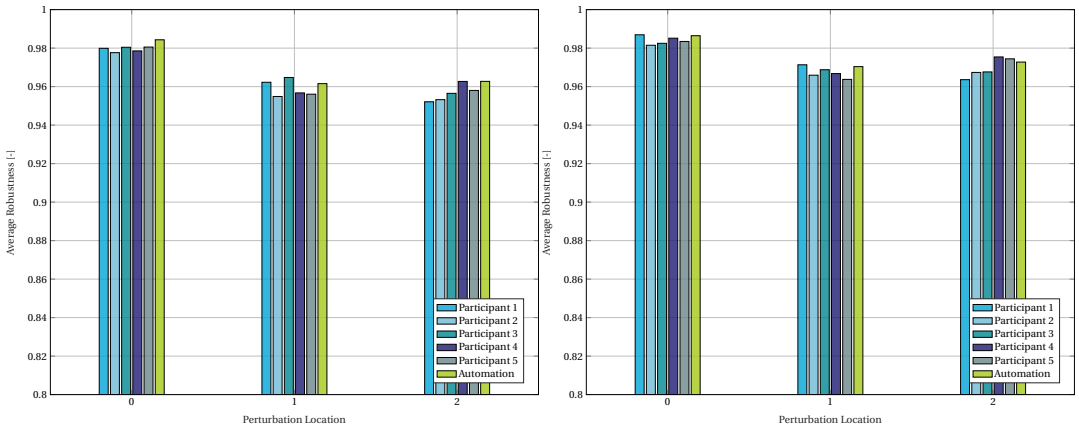
Figure D.1: Per traffic density the average sector robustness shown for perturbation location and participant. Including perturbation in robustness calculation.



(a) High traffic density

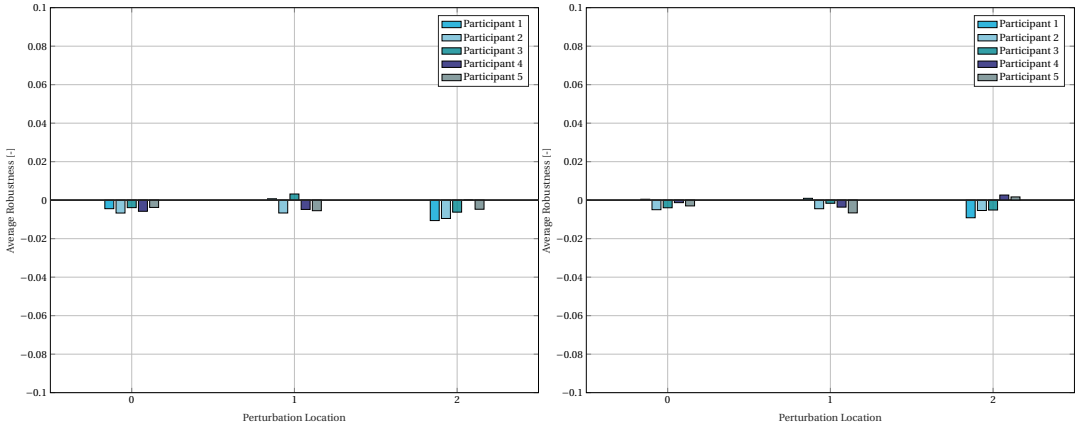
(b) Low traffic density

Figure D.2: Per traffic density the difference in average sector robustness with automation shown for perturbation location and participant. Including perturbation in robustness calculation.



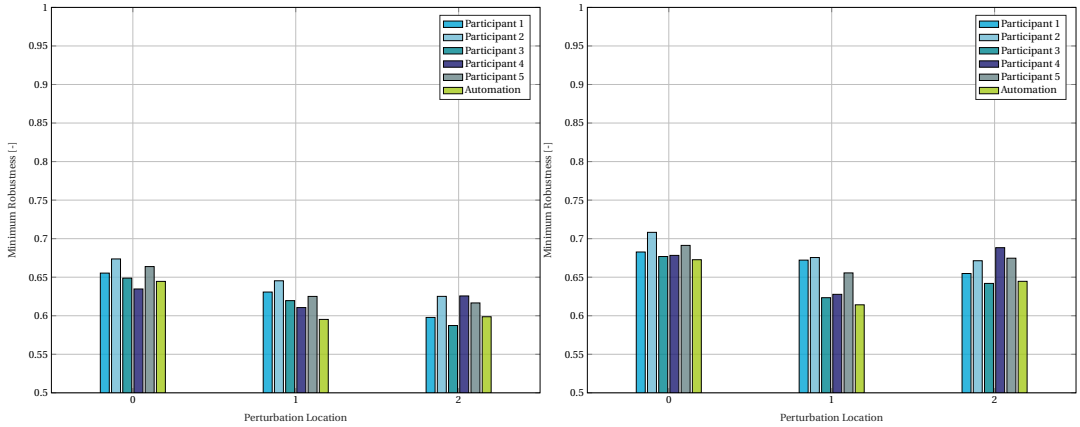
(a) High traffic density (b) Low traffic density

Figure D.3: Per traffic density the average sector robustness shown for perturbation location and participant. Without perturbation in robustness calculation.



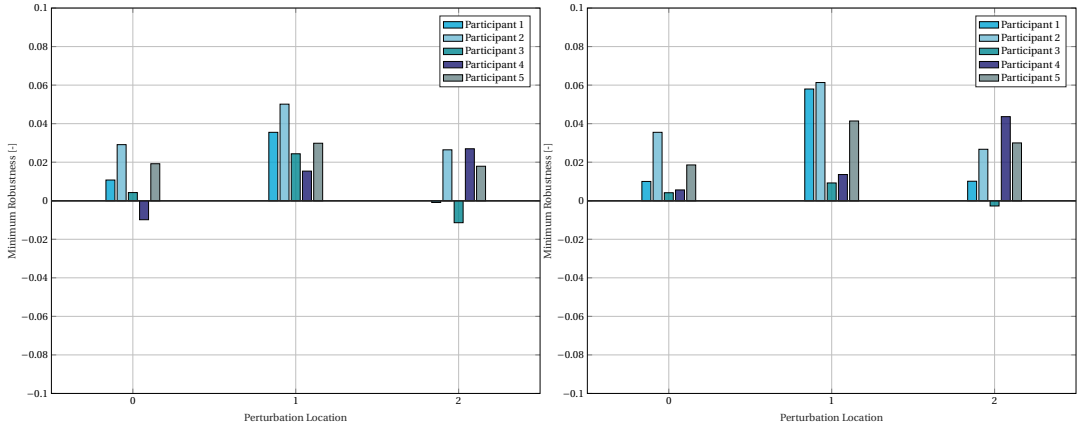
(a) High traffic density (b) Low traffic density

Figure D.4: Per traffic density the difference in average sector robustness with automation shown for perturbation location and participant. Without perturbation in robustness calculation.



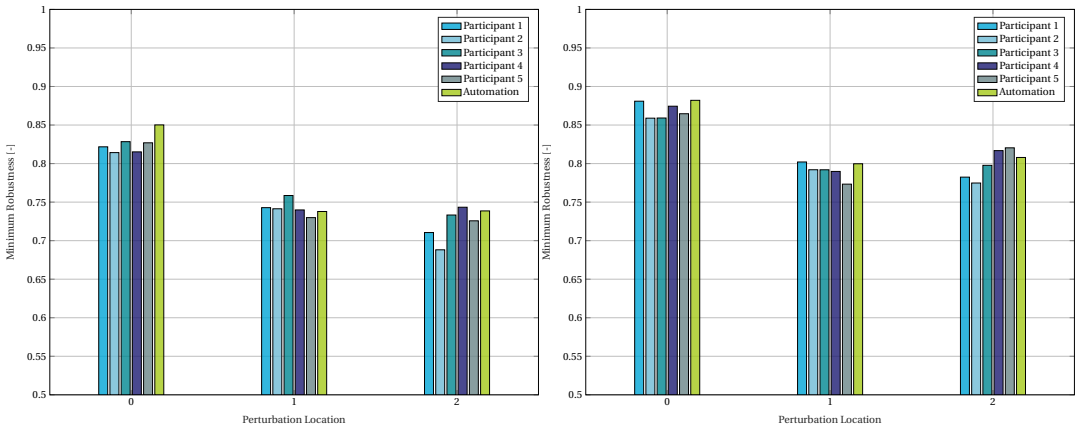
(a) High traffic density (b) Low traffic density

Figure D.5: Per traffic density the minimum sector robustness shown for perturbation location and participant. Including perturbation in robustness calculation.



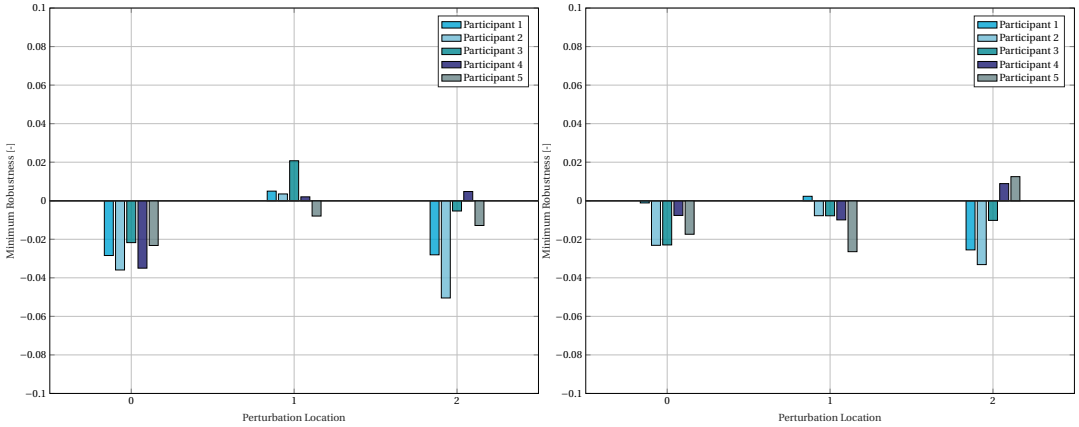
(a) High traffic density (b) Low traffic density

Figure D.6: Per traffic density the difference in minimum sector robustness with automation shown for perturbation location and participant. Including perturbation in robustness calculation.



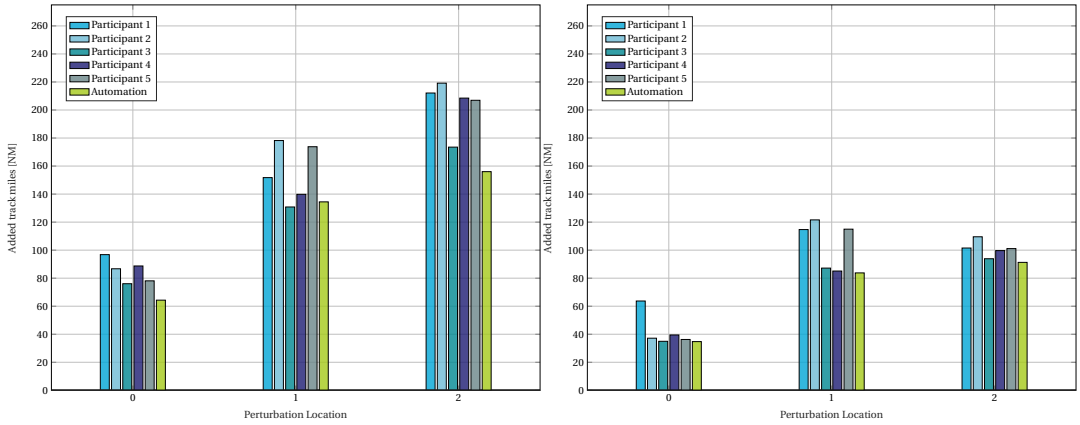
(a) High traffic density (b) Low traffic density

Figure D.7: Per traffic density the minimum sector robustness shown for perturbation location and participant. Without perturbation in robustness calculation.



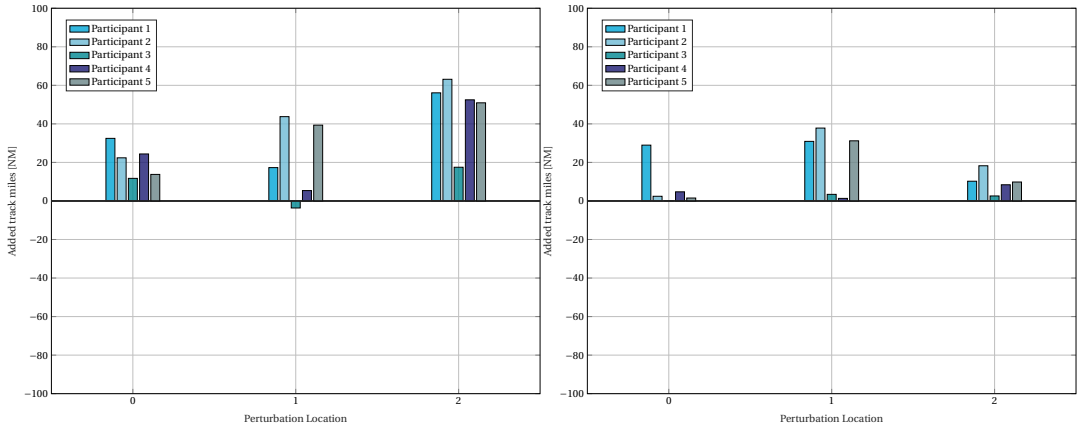
(a) High traffic density (b) Low traffic density

Figure D.8: Per traffic density the difference in minimum sector robustness with automation shown for perturbation location and participant. Without perturbation in robustness calculation.



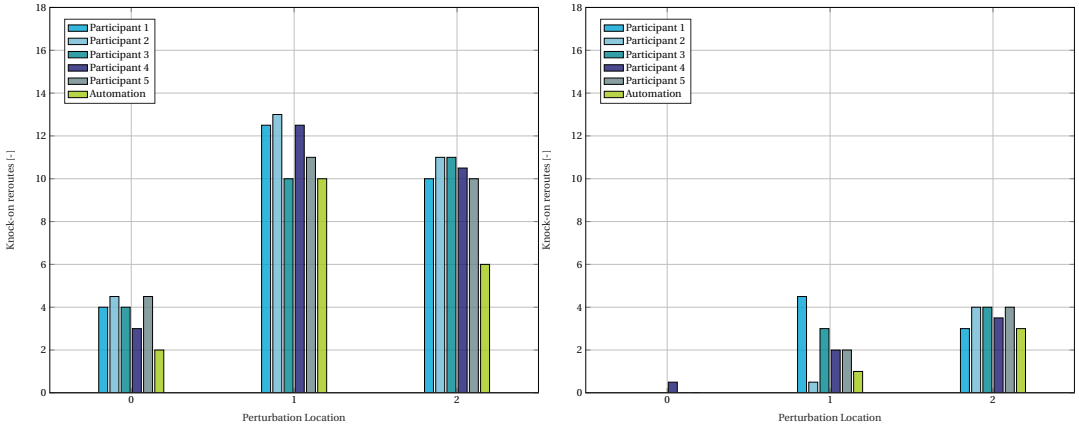
(a) High traffic density (b) Low traffic density

Figure D.9: Per traffic density added track miles shown for perturbation location and participant.



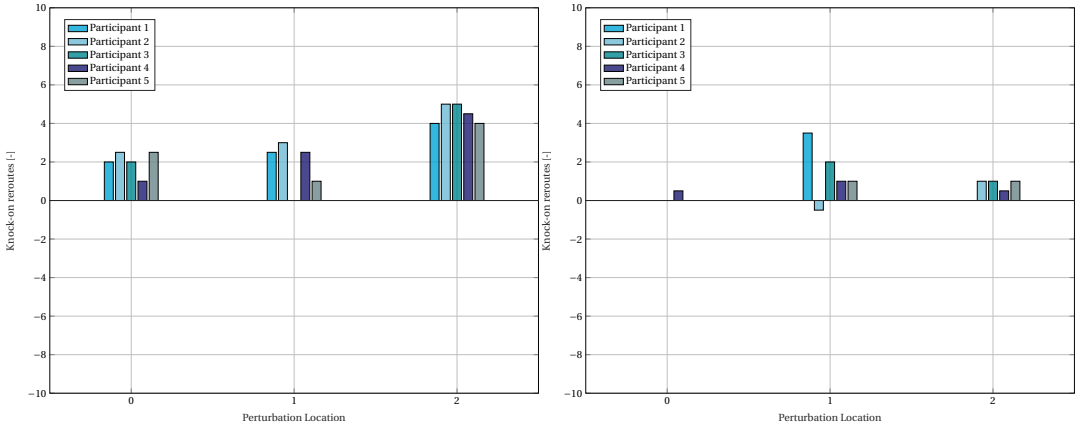
(a) High traffic density (b) Low traffic density

Figure D.10: Per traffic density the difference in added track miles with automation shown for perturbation location and participant.



(a) High traffic density (b) Low traffic density

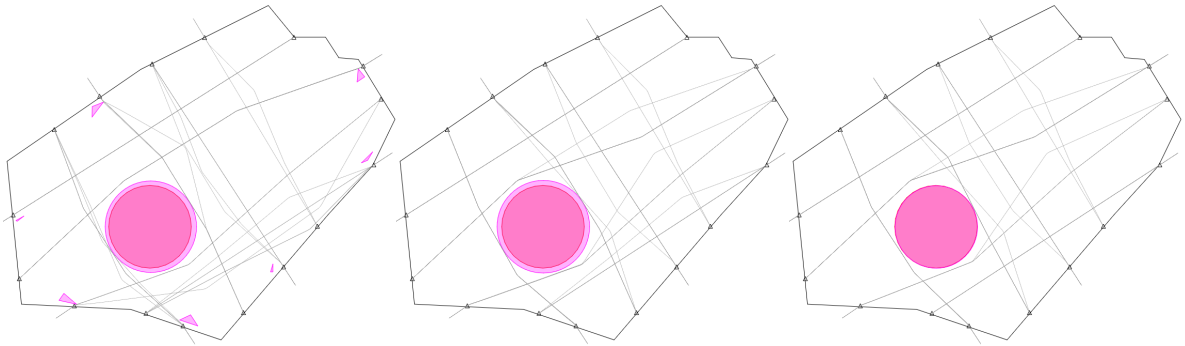
Figure D.11: Per traffic density knock-on reroutes shown for perturbation location and participant.



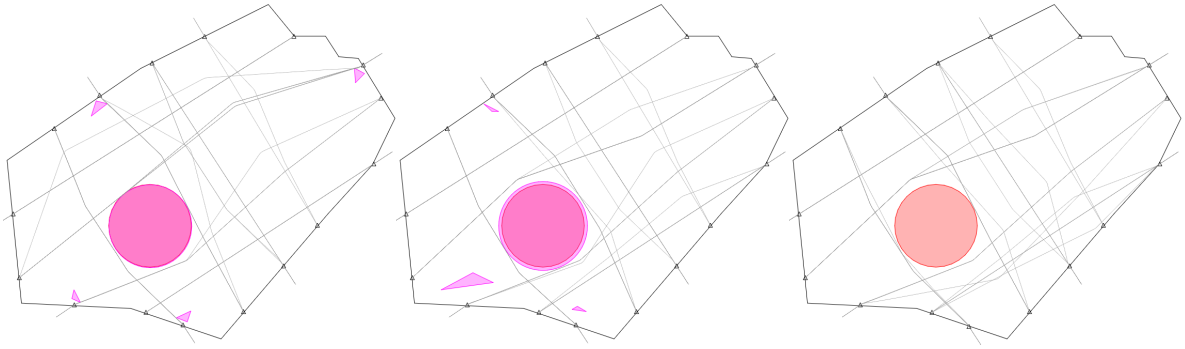
(a) High traffic density (b) Low traffic density

Figure D.12: Per traffic density the difference in knock-on reroutes with automation shown for perturbation location and participant.

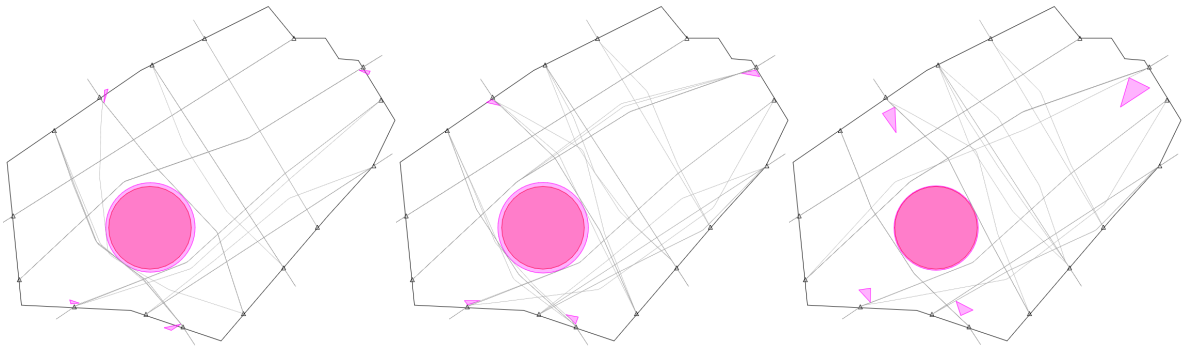
D.2. Traffic scenario results per perturbation location



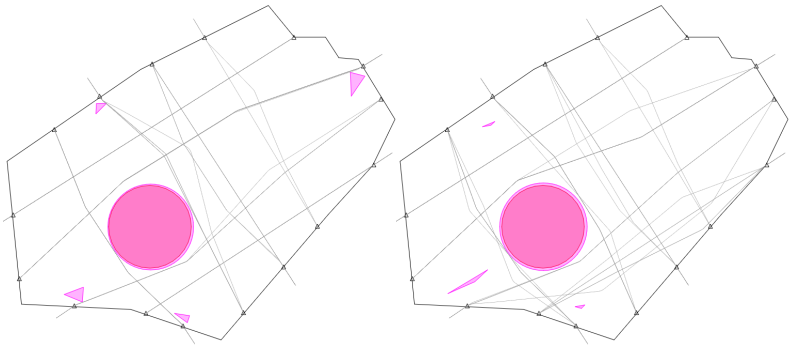
(a) Participant 1 - 0^o (b) Participant 2 - 0^o (c) Participant 3 - 0^o



(d) Participant 4 - 0^o (e) Participant 5 - 0^o (f) Automation

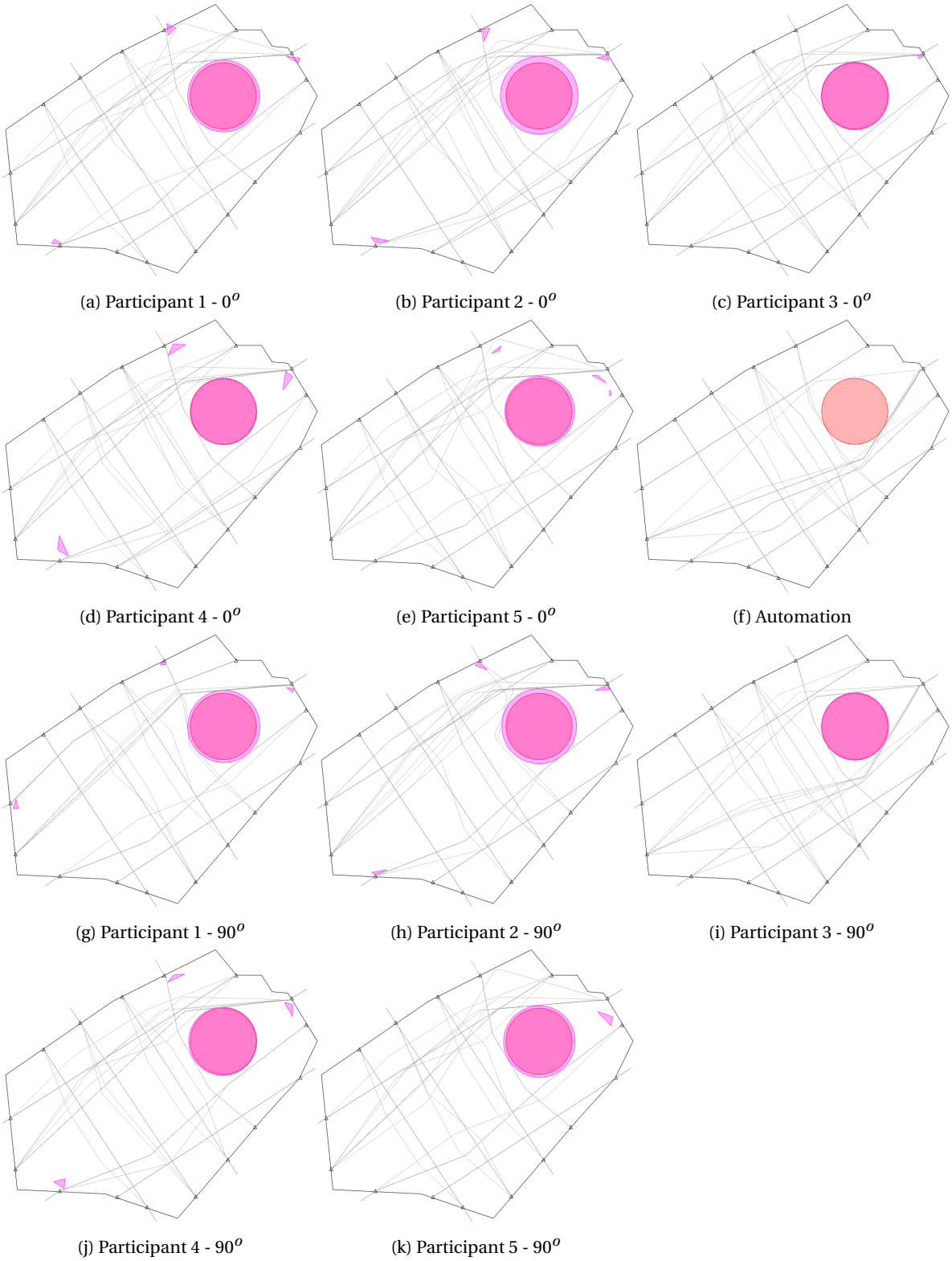


(g) Participant 1 - 90^o (h) Participant 2 - 90^o (i) Participant 3 - 90^o



(j) Participant 4 - 90^o (k) Participant 5 - 90^o

Figure D.13: Result trajectories perturbation location 0 for 0^o and 90^o scenarios.



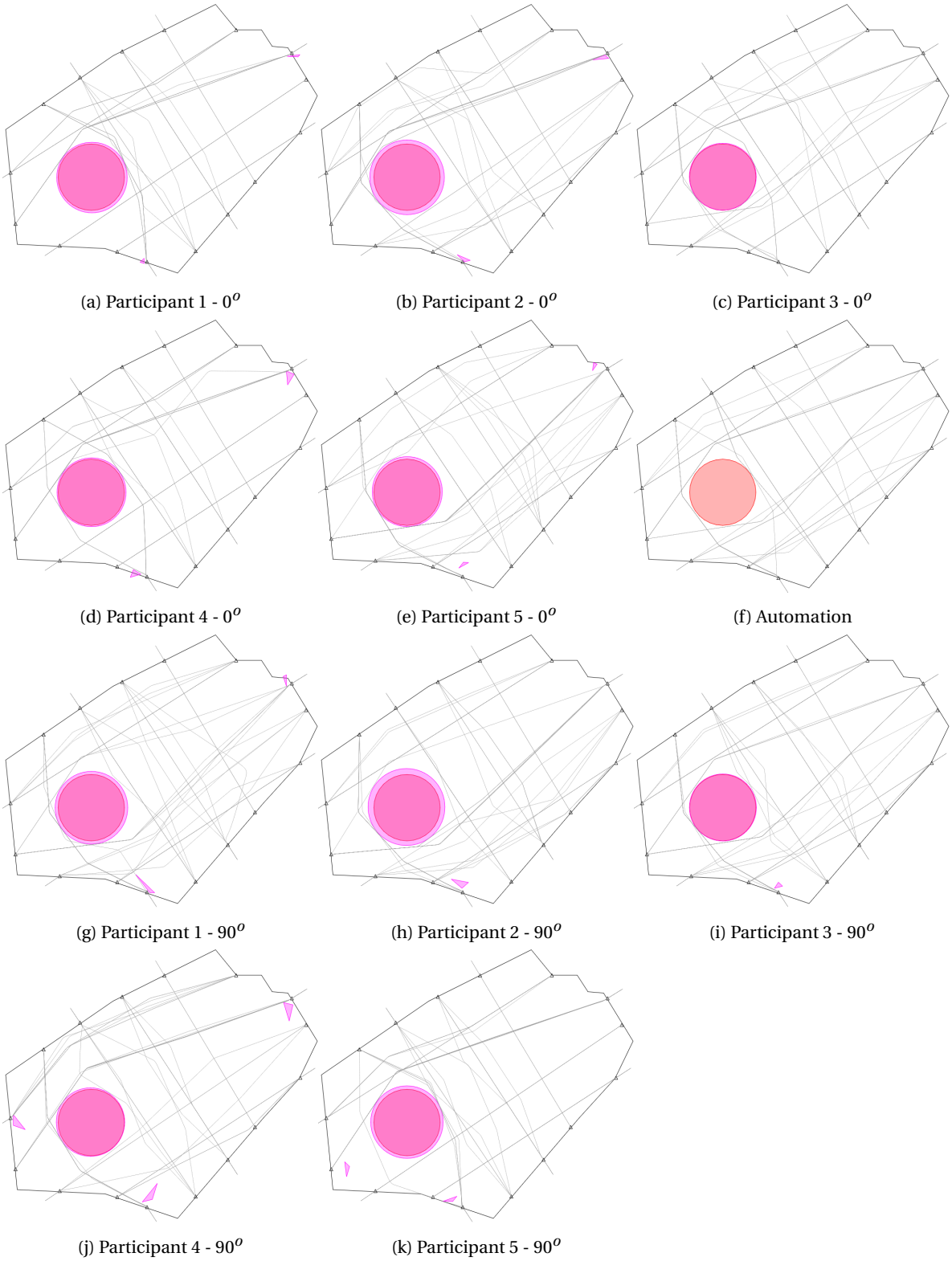
(a) Participant 1 - 0^o (b) Participant 2 - 0^o (c) Participant 3 - 0^o

(d) Participant 4 - 0^o (e) Participant 5 - 0^o (f) Automation

(g) Participant 1 - 90^o (h) Participant 2 - 90^o (i) Participant 3 - 90^o

(j) Participant 4 - 90^o (k) Participant 5 - 90^o

Figure D.14: Result trajectories perturbation location 1 for 0^o and 90^o scenarios.



(a) Participant 1 - 0^o (b) Participant 2 - 0^o (c) Participant 3 - 0^o

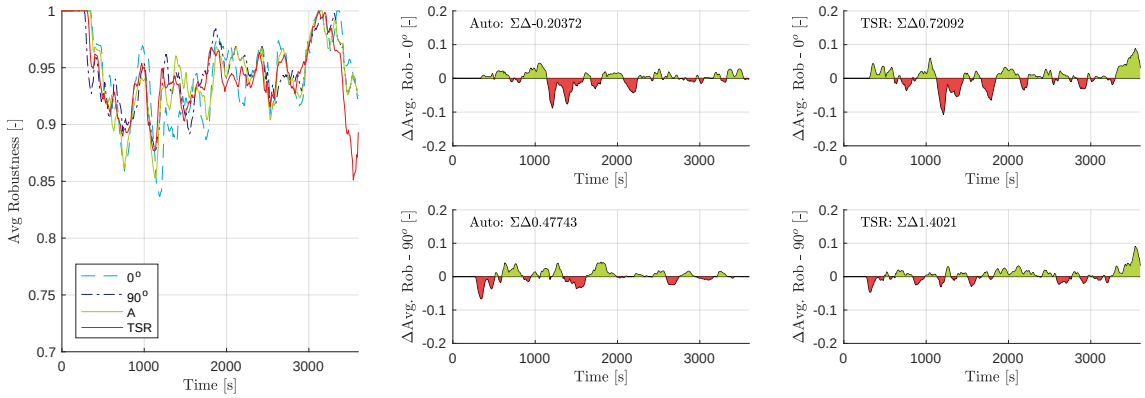
(d) Participant 4 - 0^o (e) Participant 5 - 0^o (f) Automation

(g) Participant 1 - 90^o (h) Participant 2 - 90^o (i) Participant 3 - 90^o

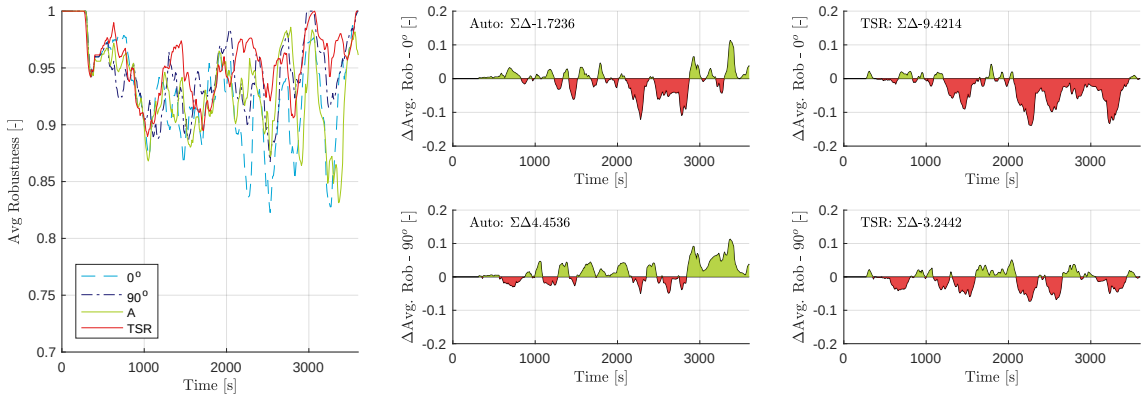
(j) Participant 4 - 90^o (k) Participant 5 - 90^o

Figure D.15: Result trajectories perturbation location 2 for 0^o and 90^o scenarios.

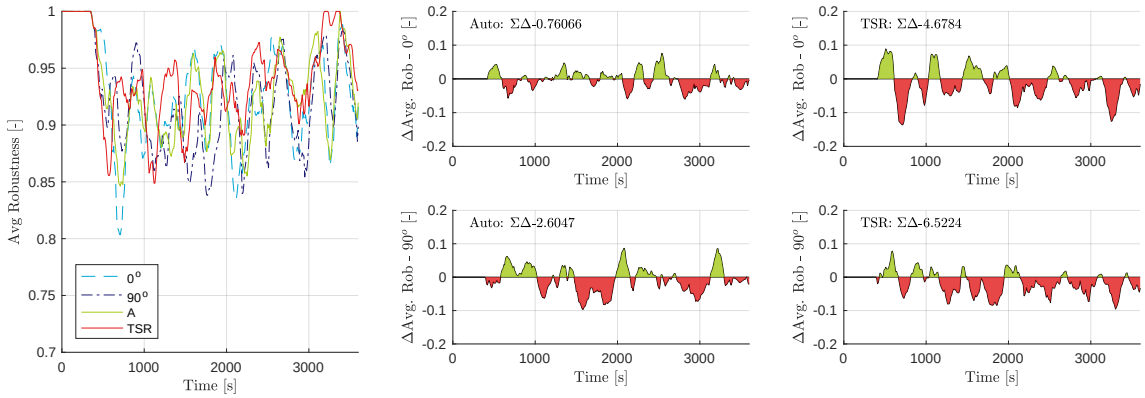
D.3. Participant's average and minimum robustness over time - including perturbation



(a) Perturbation Location 0

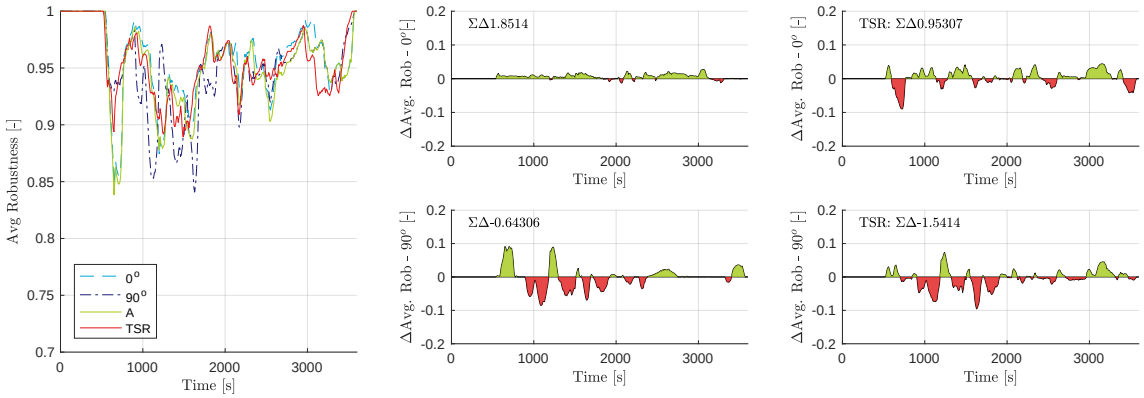


(b) Perturbation Location 1

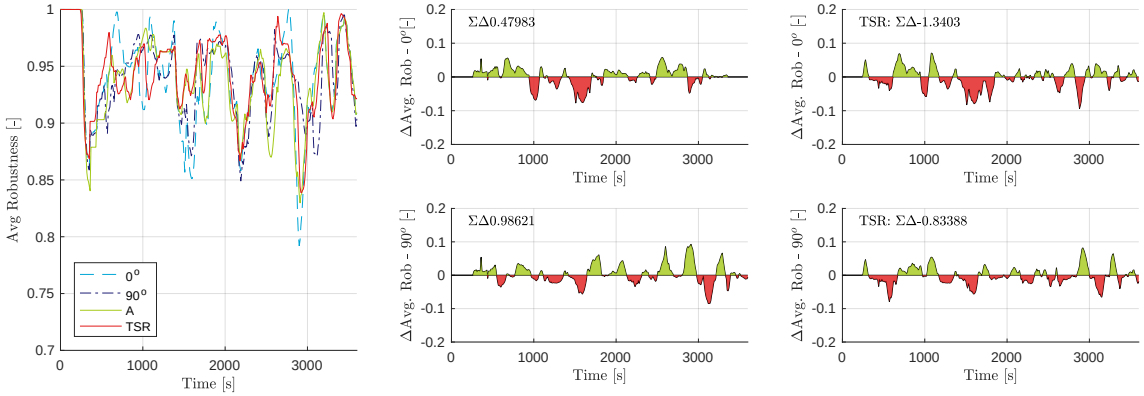


(c) Perturbation Location 2

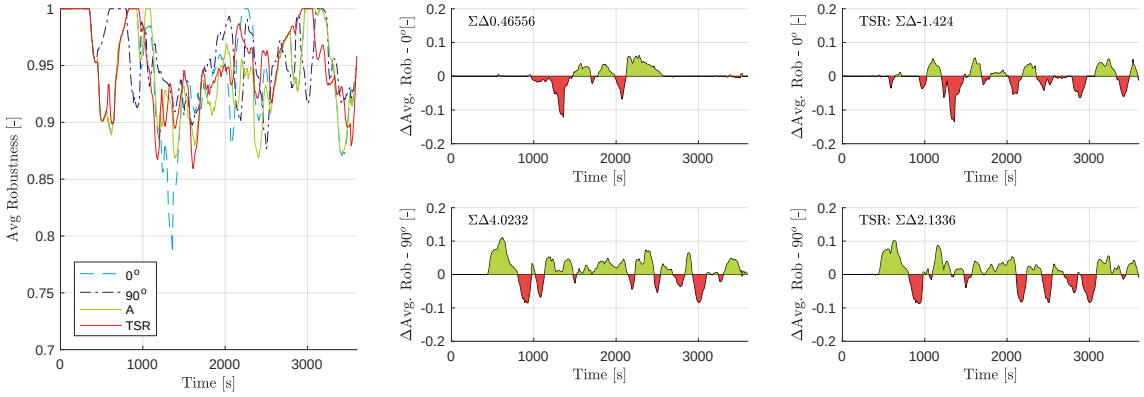
Figure D.16: Average sector robustness high density traffic scenario participant 1



(a) Perturbation Location 0

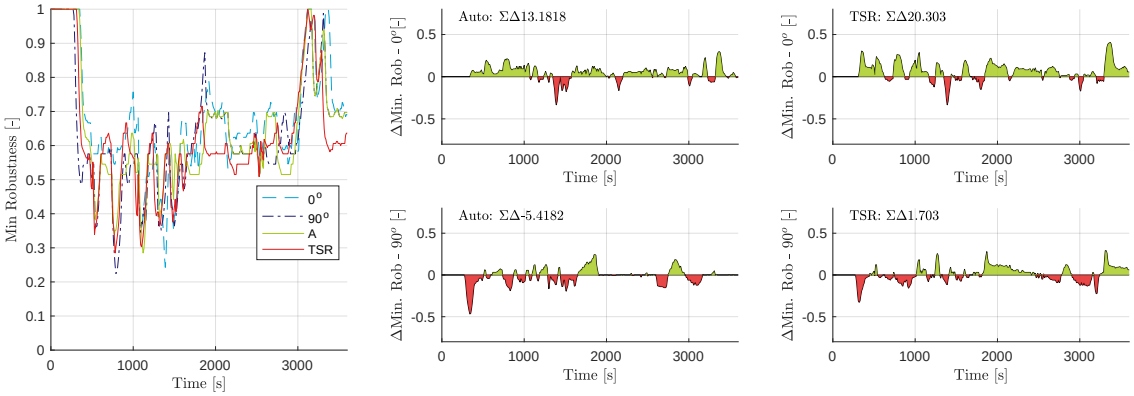


(b) Perturbation Location 1

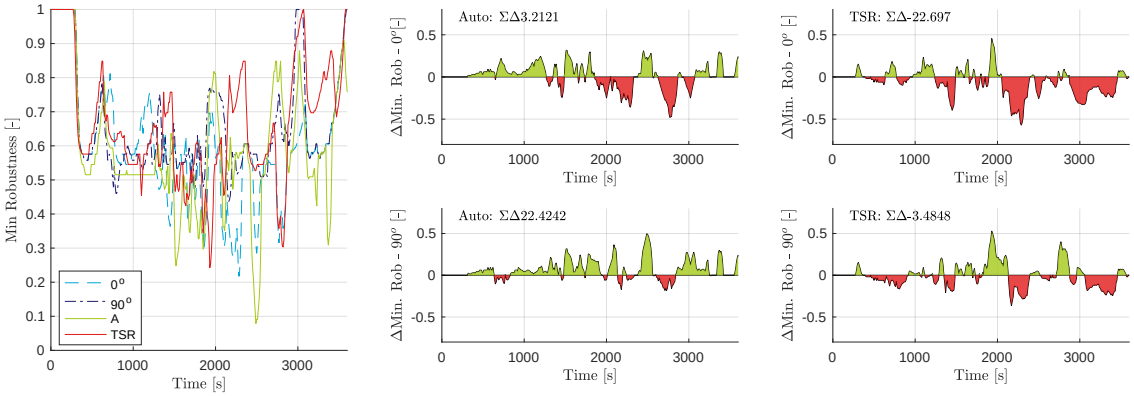


(c) Perturbation Location 2

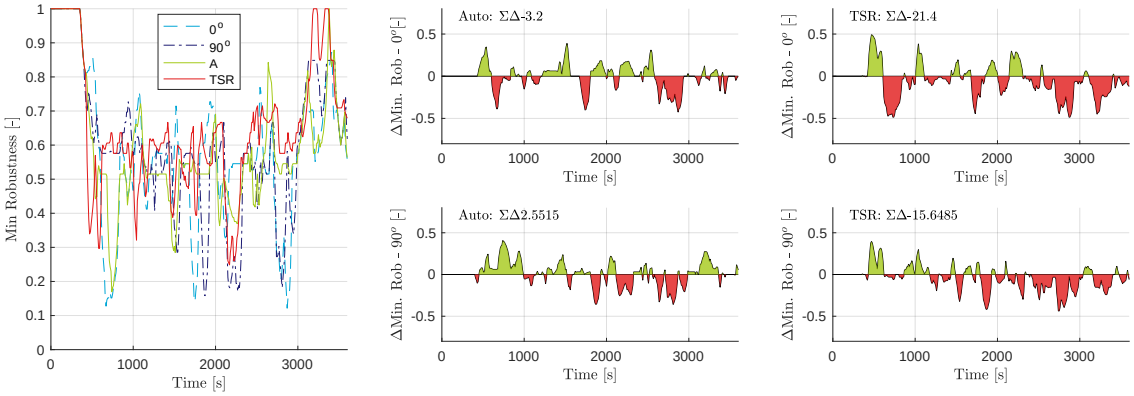
Figure D.17: Average sector robustness low density traffic scenario participant 1



(a) Perturbation Location 0

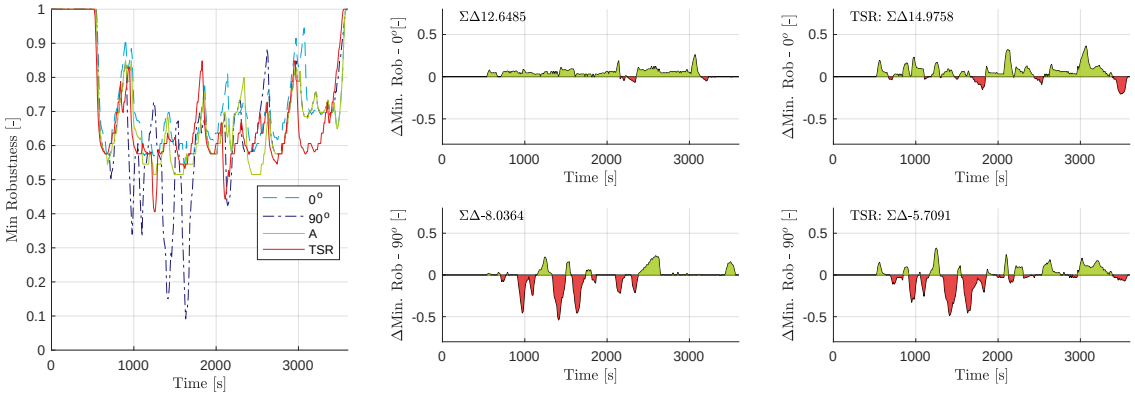


(b) Perturbation Location 1

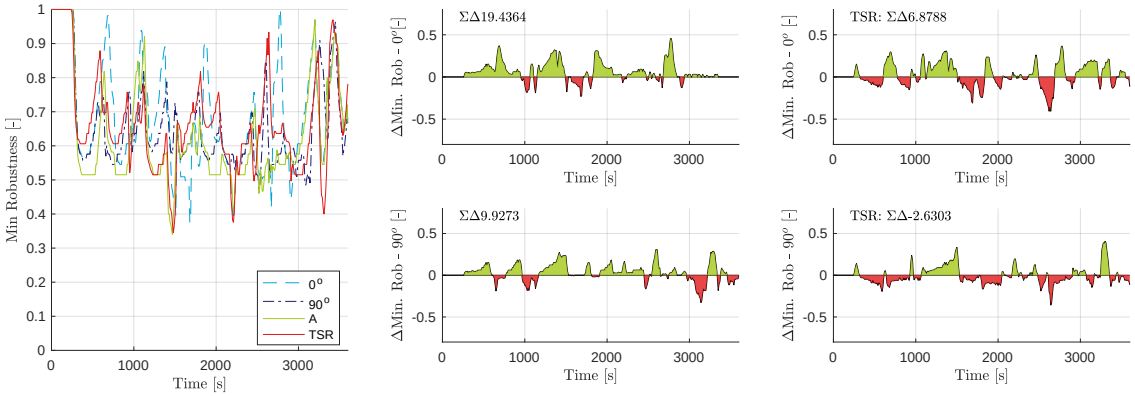


(c) Perturbation Location 2

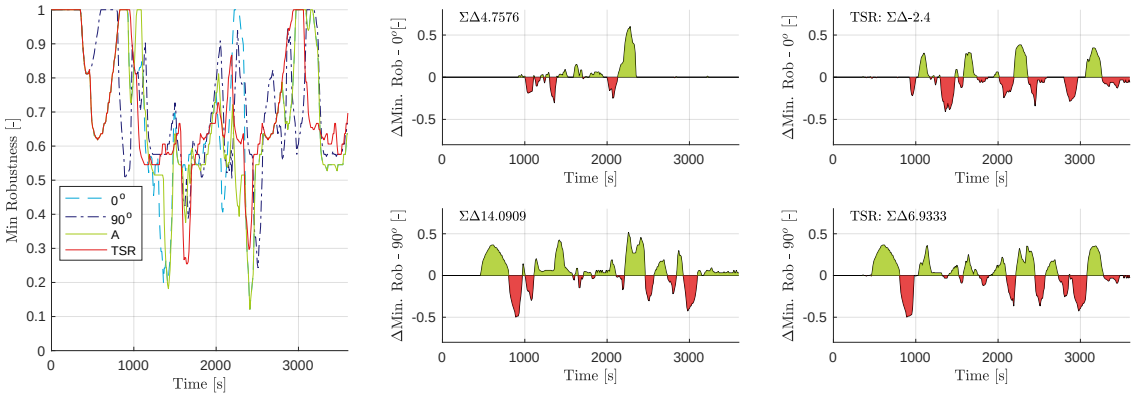
Figure D.18: Minimum sector robustness high density traffic scenario participant 1



(a) Perturbation Location 0

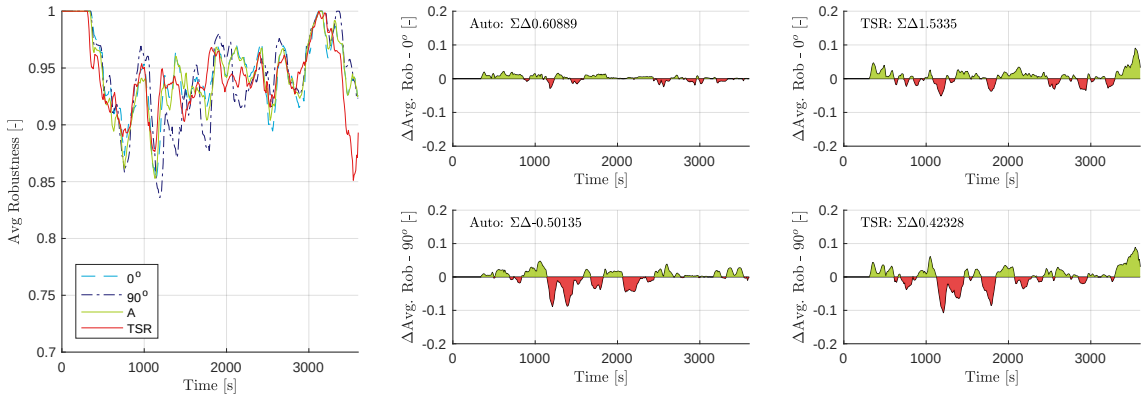


(b) Perturbation Location 1

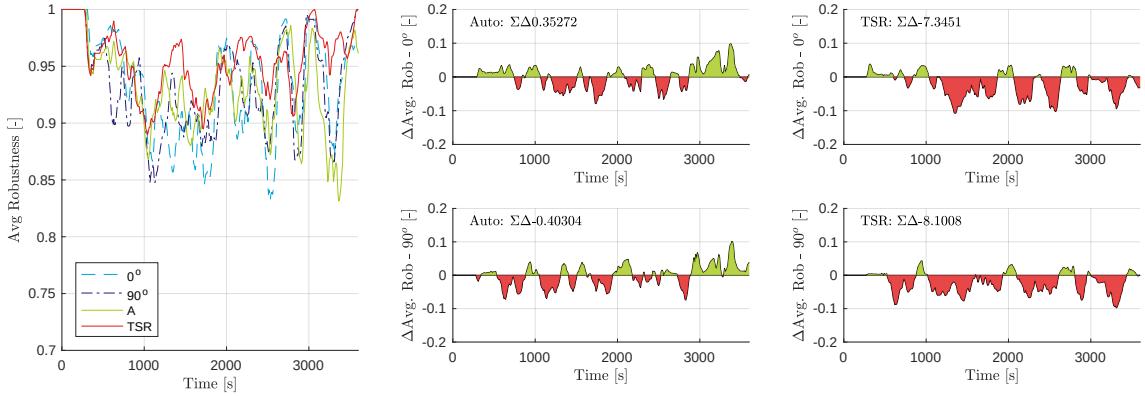


(c) Perturbation Location 2

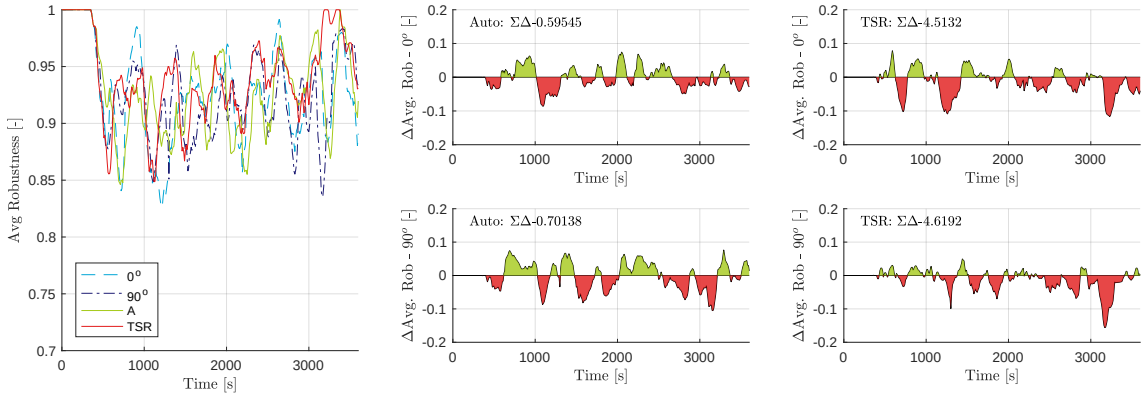
Figure D.19: Minimum sector robustness low density traffic scenario participant 1



(a) Perturbation Location 0

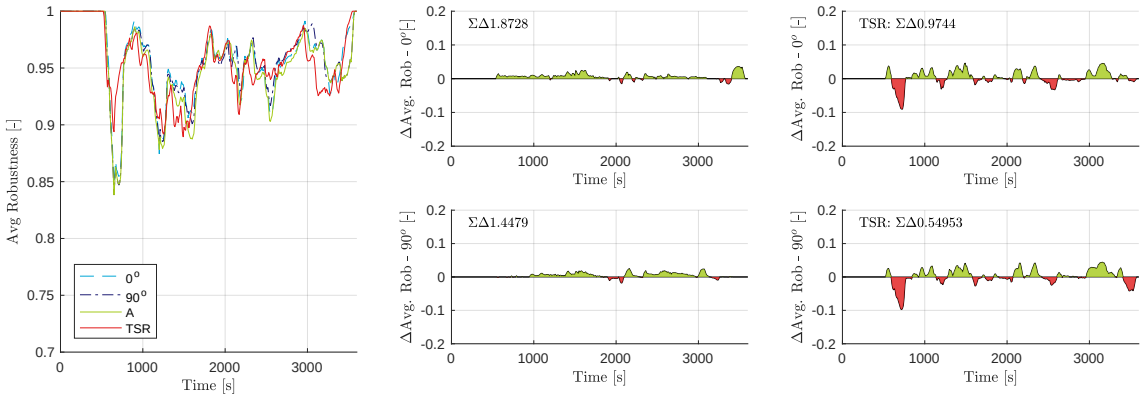


(b) Perturbation Location 1

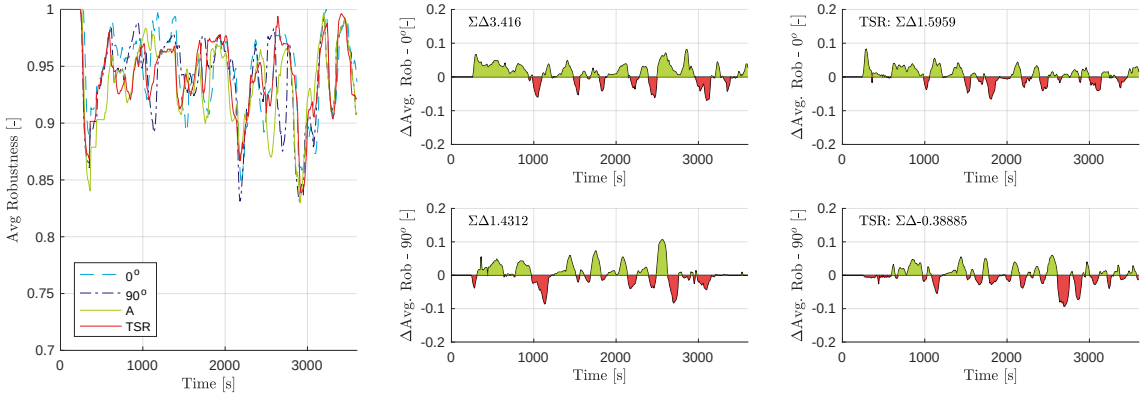


(c) Perturbation Location 2

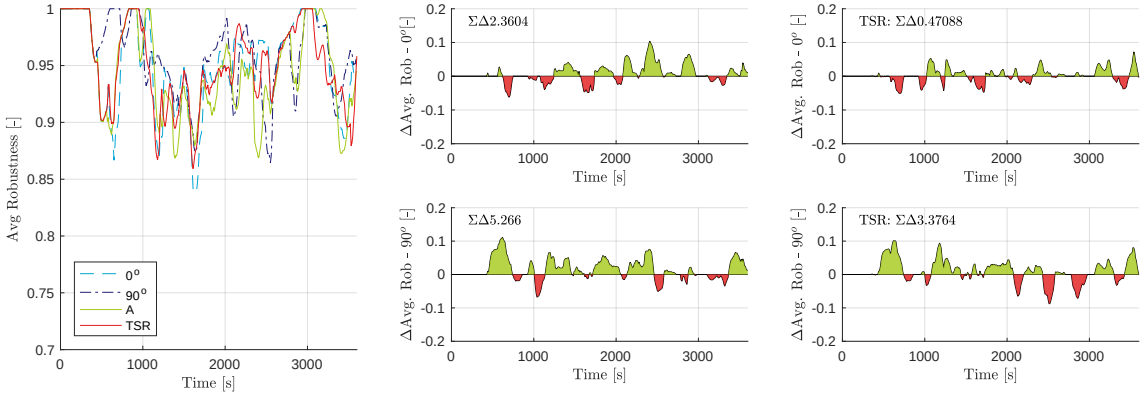
Figure D.20: Average sector robustness high density traffic scenario participant 2



(a) Perturbation Location 0

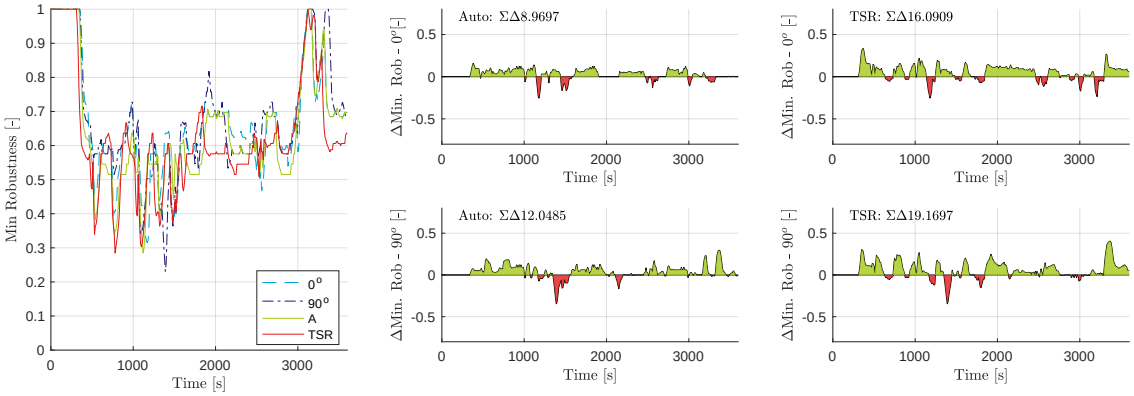


(b) Perturbation Location 1

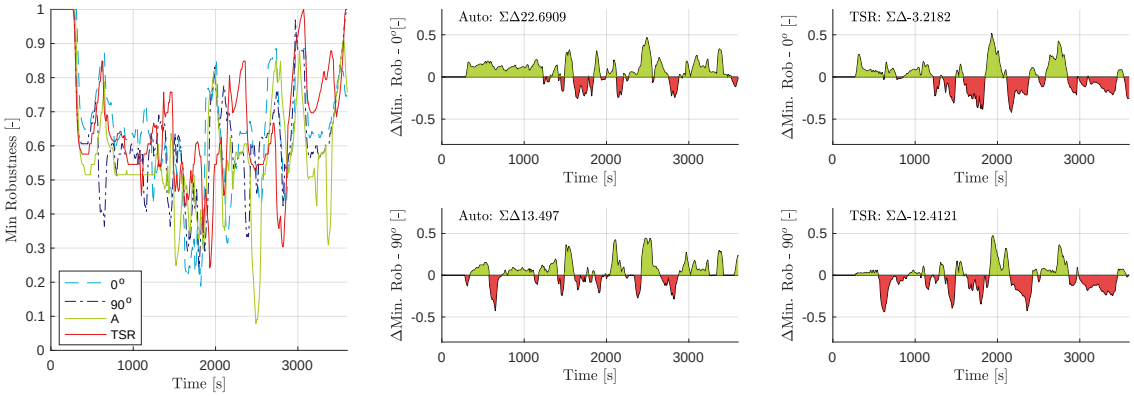


(c) Perturbation Location 2

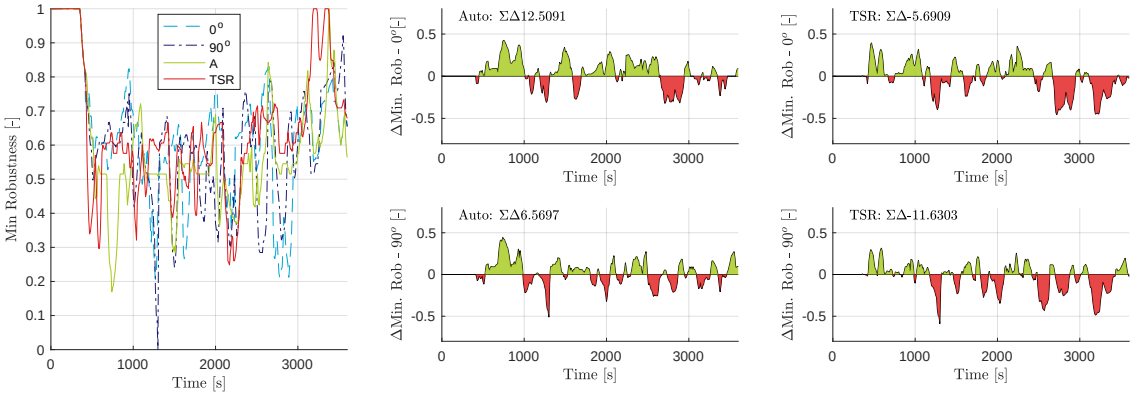
Figure D.21: Average sector robustness low density traffic scenario participant 2



(a) Perturbation Location 0

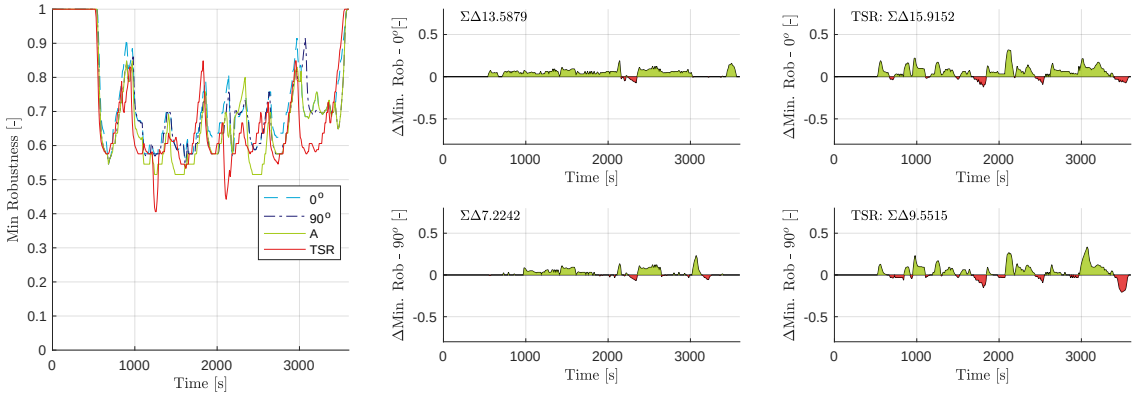


(b) Perturbation Location 1

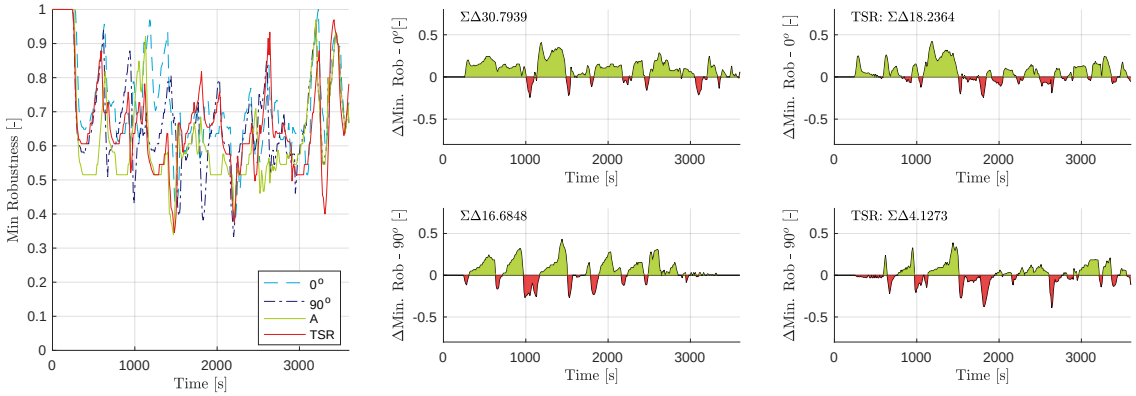


(c) Perturbation Location 2

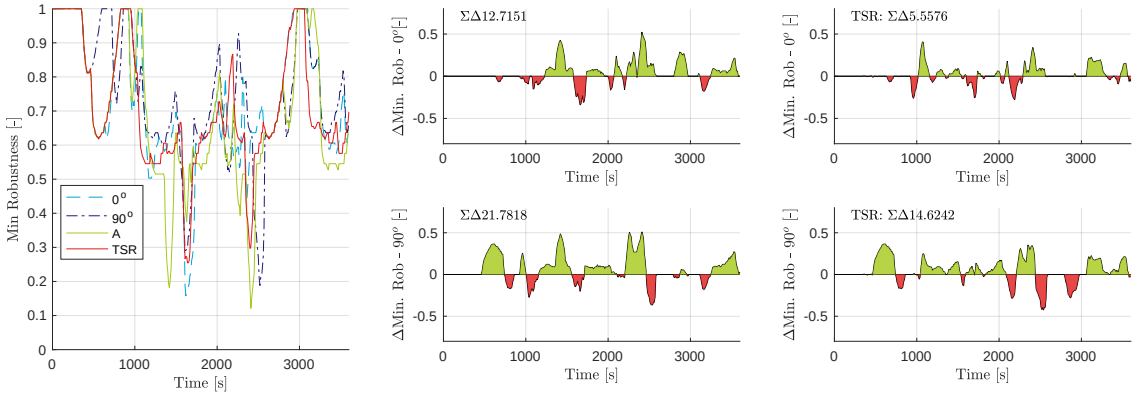
Figure D.22: Minimum sector robustness high density traffic scenario participant 2



(a) Perturbation Location 0

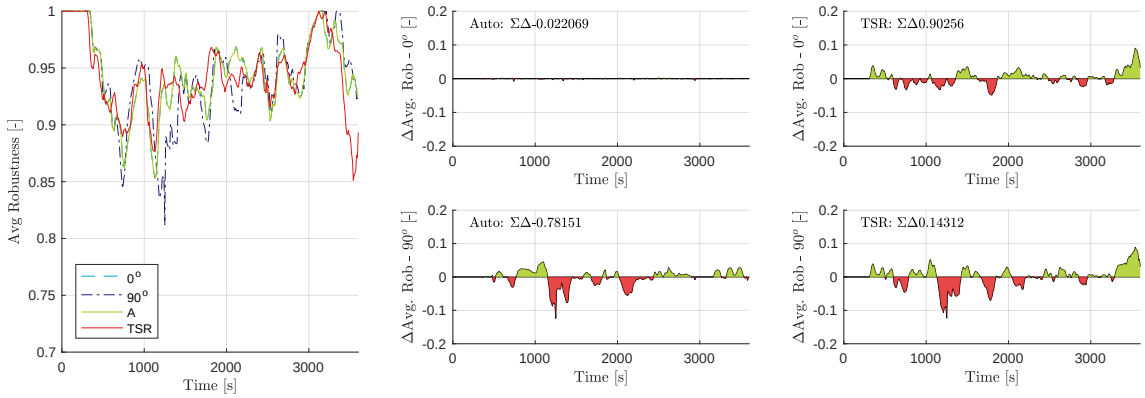


(b) Perturbation Location 1

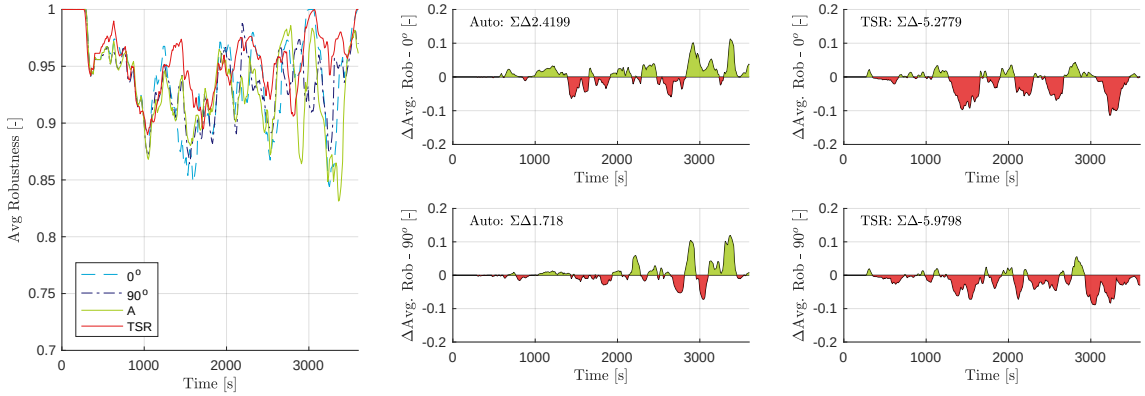


(c) Perturbation Location 2

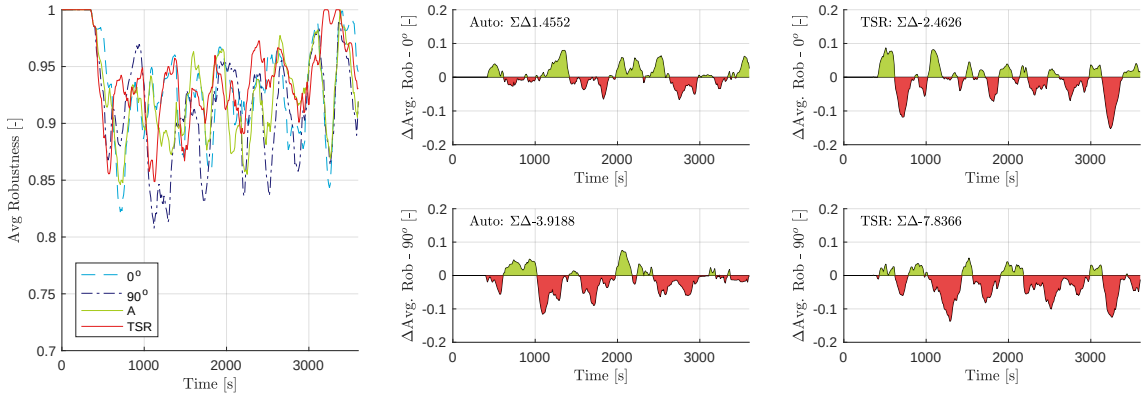
Figure D.23: Minimum sector robustness low density traffic scenario participant 2



(a) Perturbation Location 0

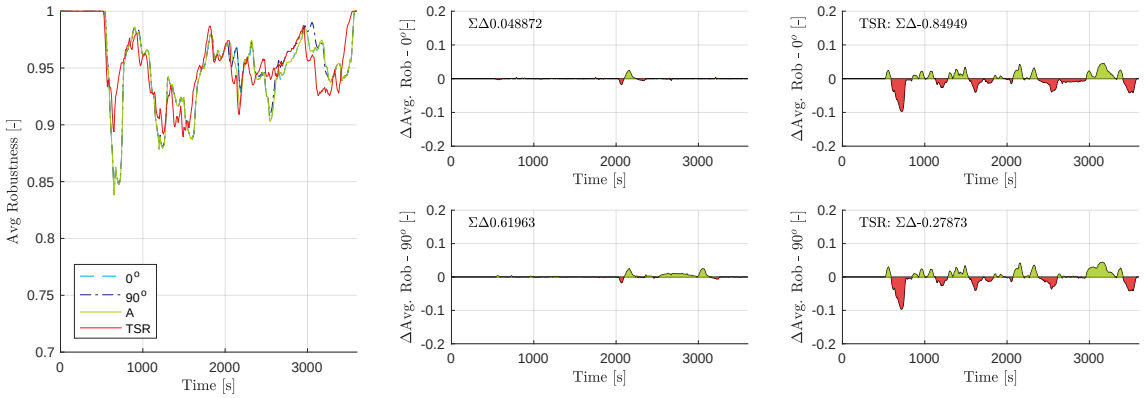


(b) Perturbation Location 1

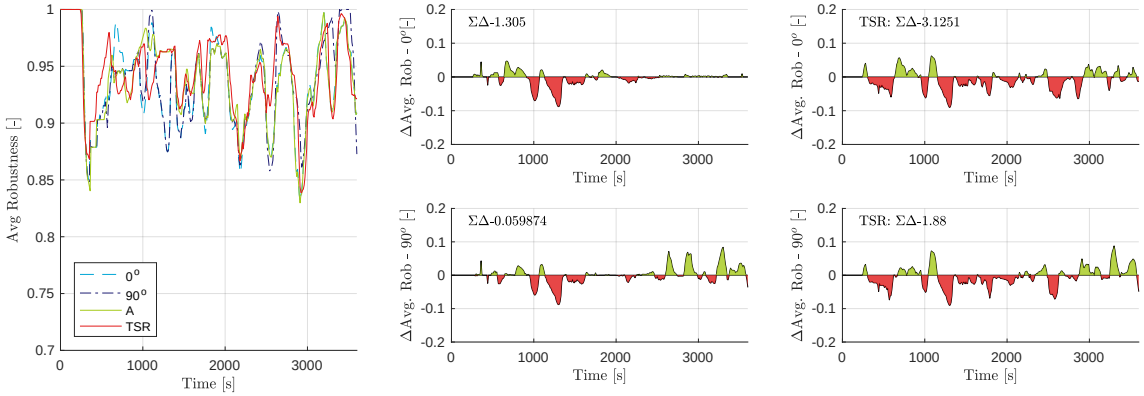


(c) Perturbation Location 2

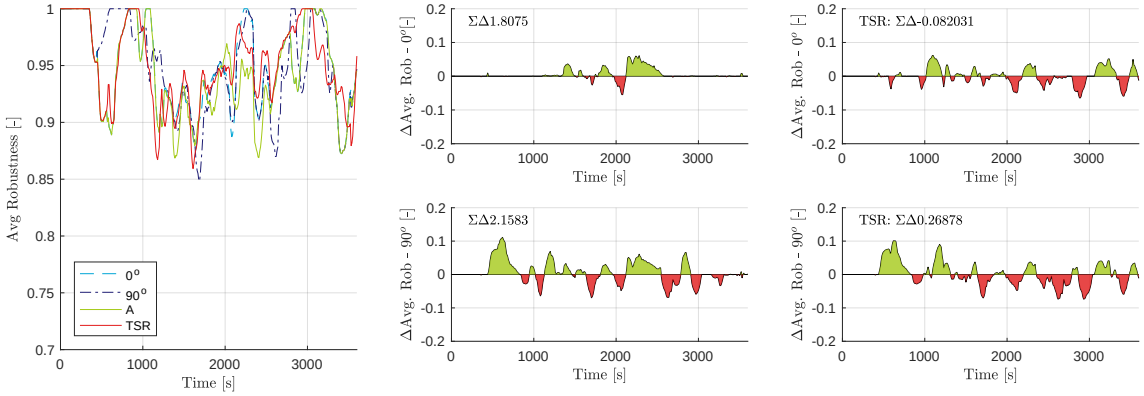
Figure D.24: Average sector robustness high density traffic scenario participant 3



(a) Perturbation Location 0

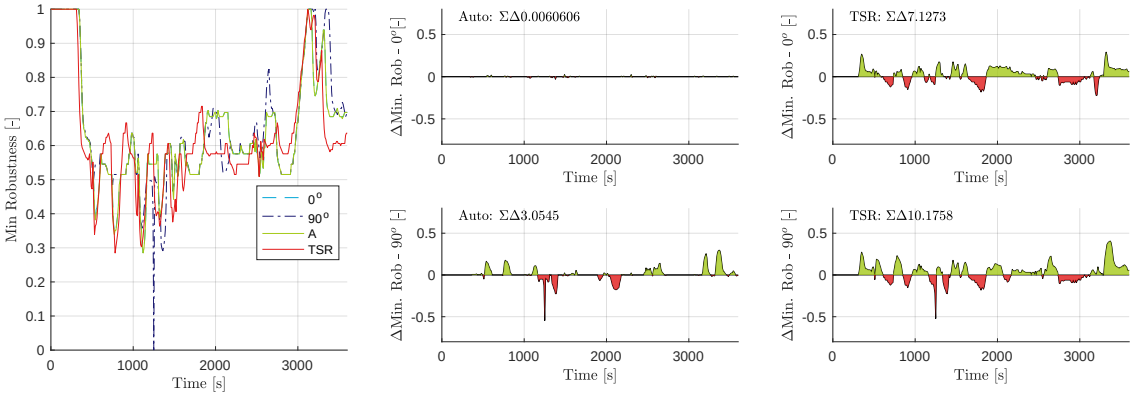


(b) Perturbation Location 1

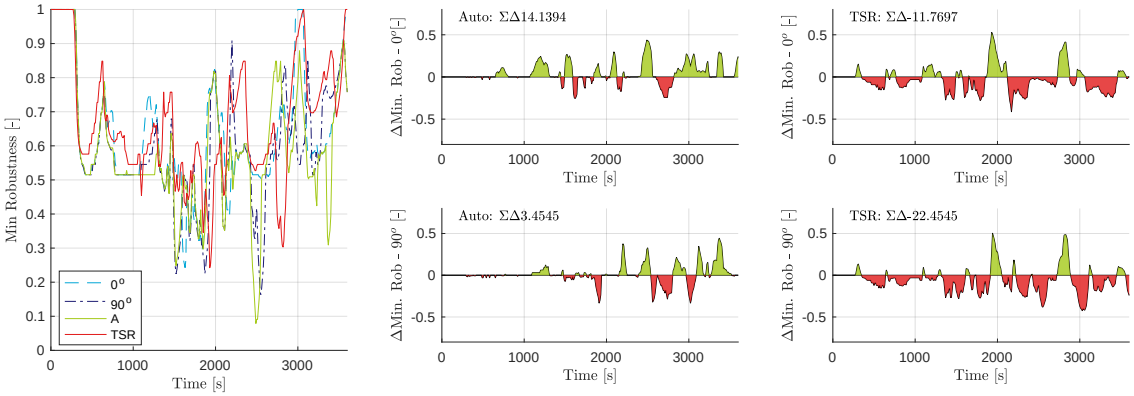


(c) Perturbation Location 2

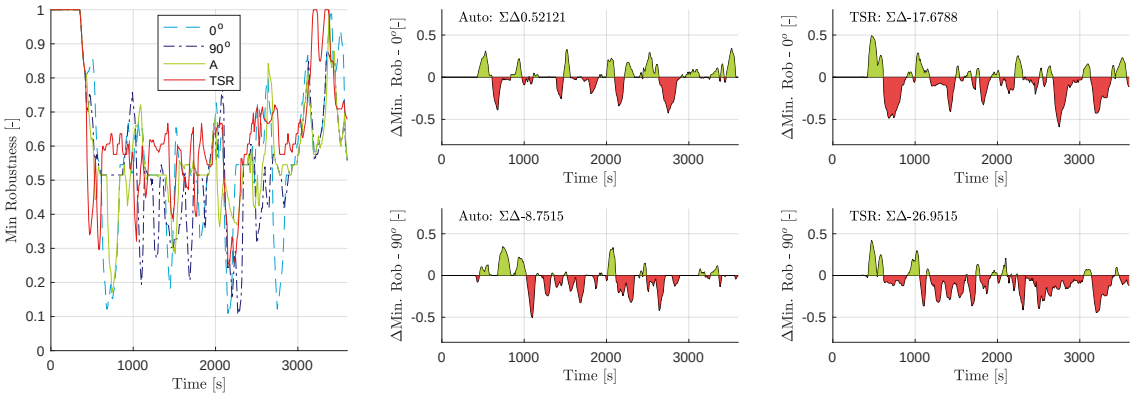
Figure D.25: Average sector robustness low density traffic scenario participant 3



(a) Perturbation Location 0

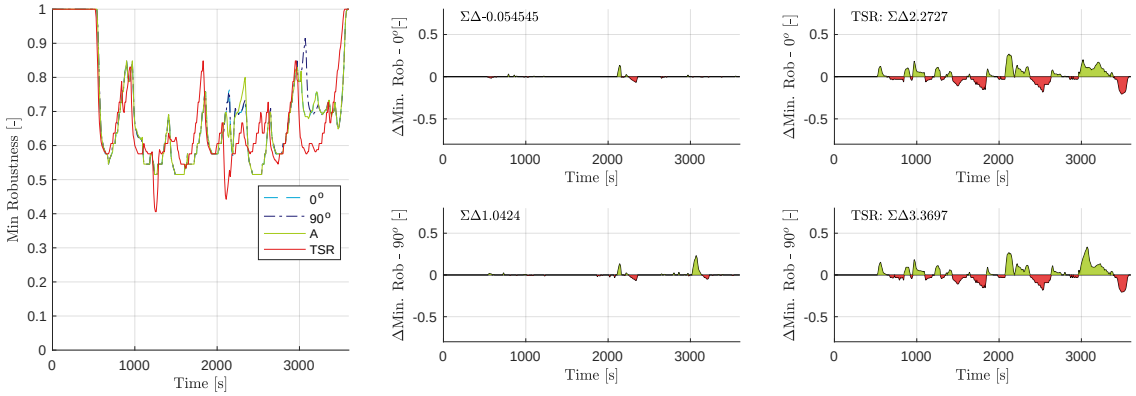


(b) Perturbation Location 1

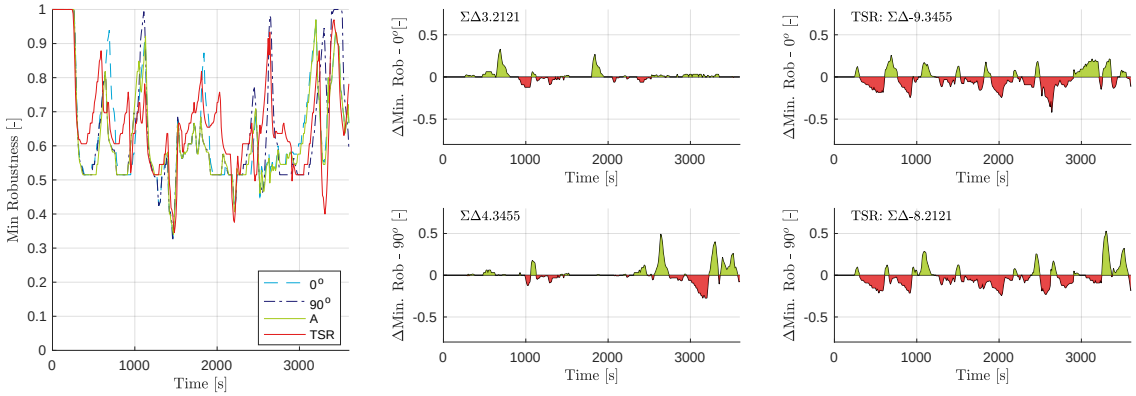


(c) Perturbation Location 2

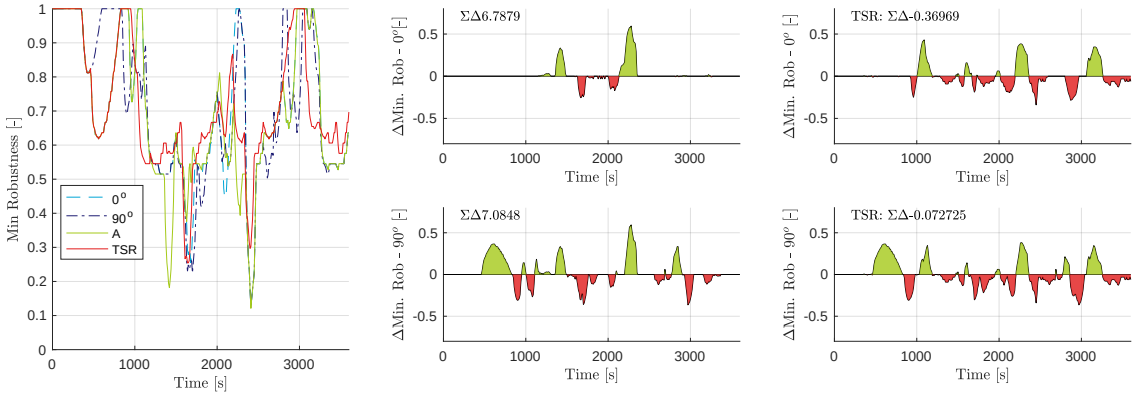
Figure D.26: Minimum sector robustness high density traffic scenario participant 3



(a) Perturbation Location 0

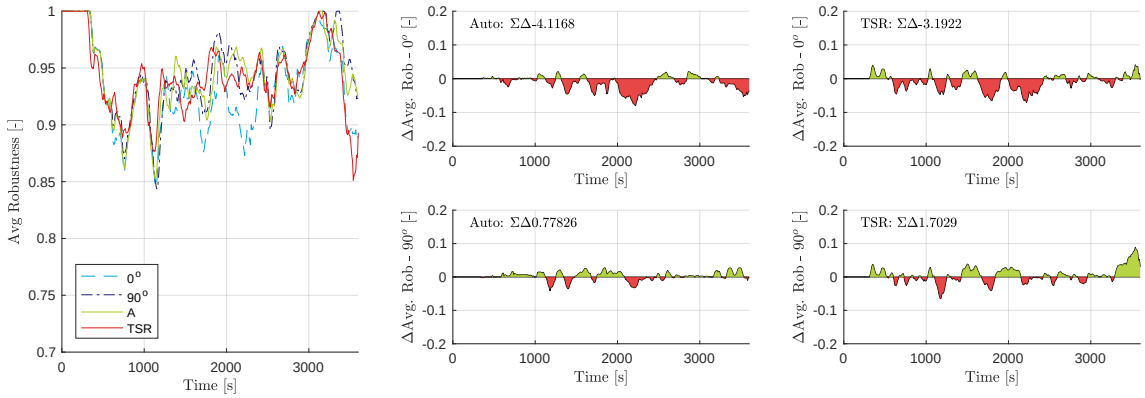


(b) Perturbation Location 1

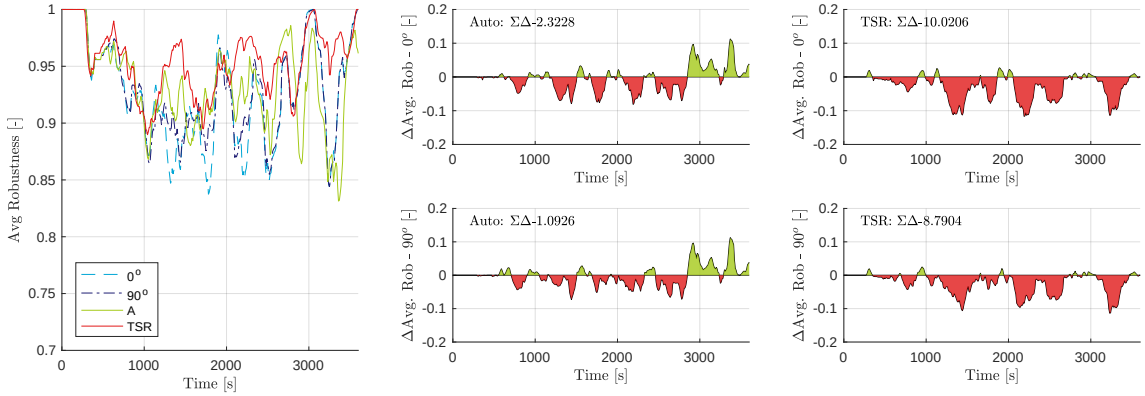


(c) Perturbation Location 2

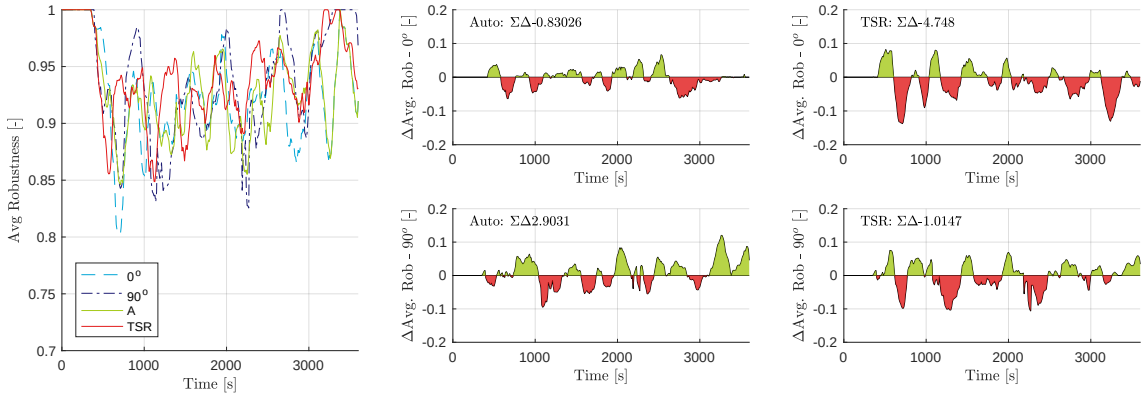
Figure D.27: Minimum sector robustness low density traffic scenario participant 3



(a) Perturbation Location 0

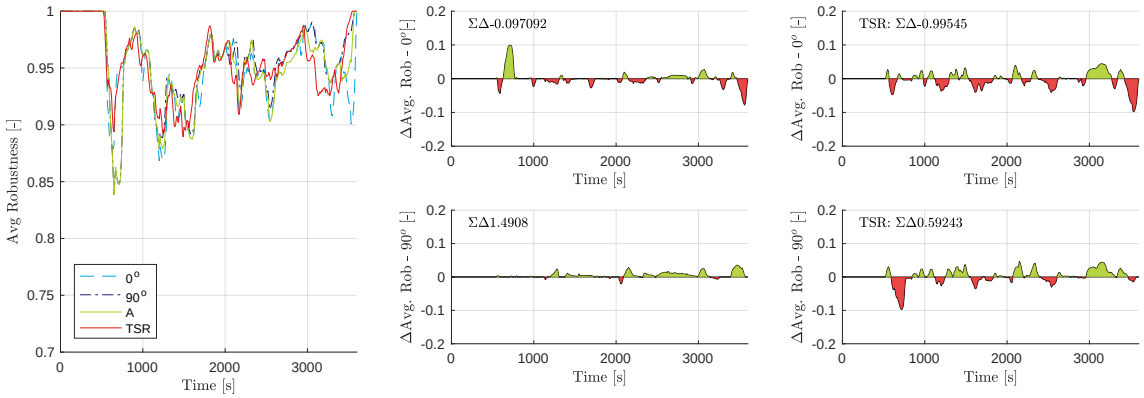


(b) Perturbation Location 1

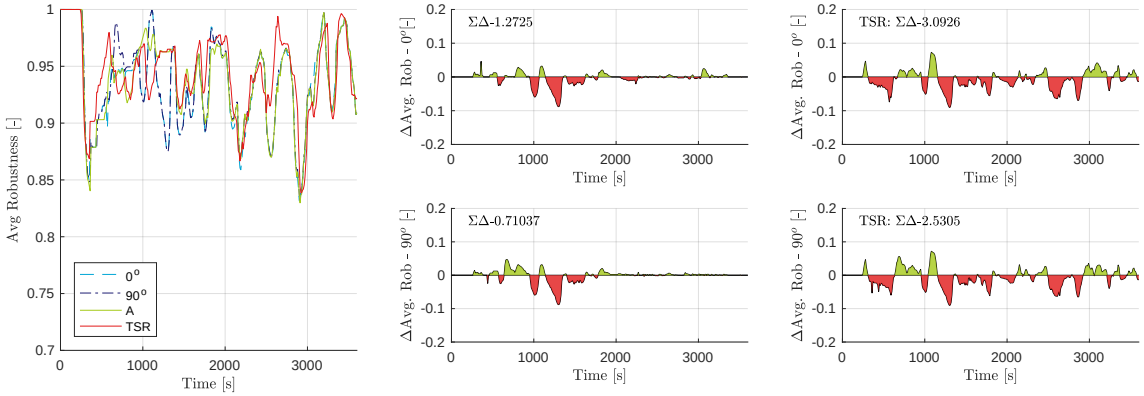


(c) Perturbation Location 2

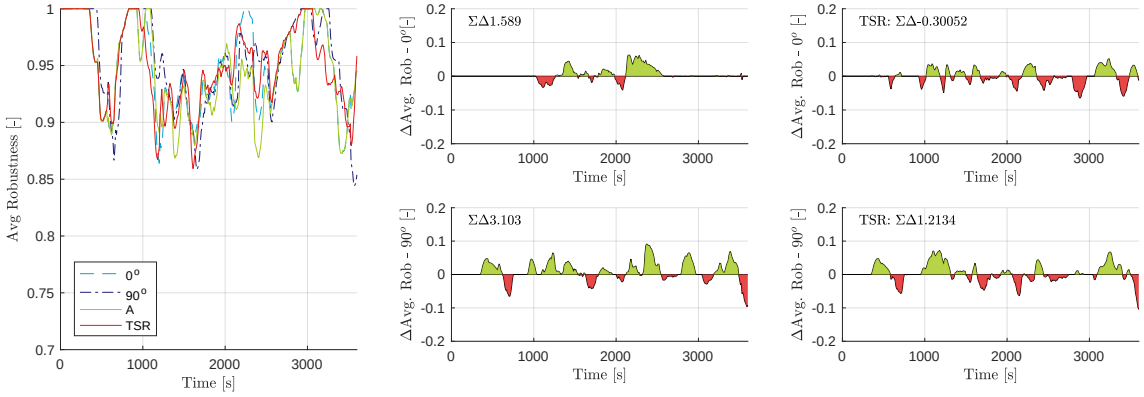
Figure D.28: Average sector robustness high density traffic scenario participant 4



(a) Perturbation Location 0

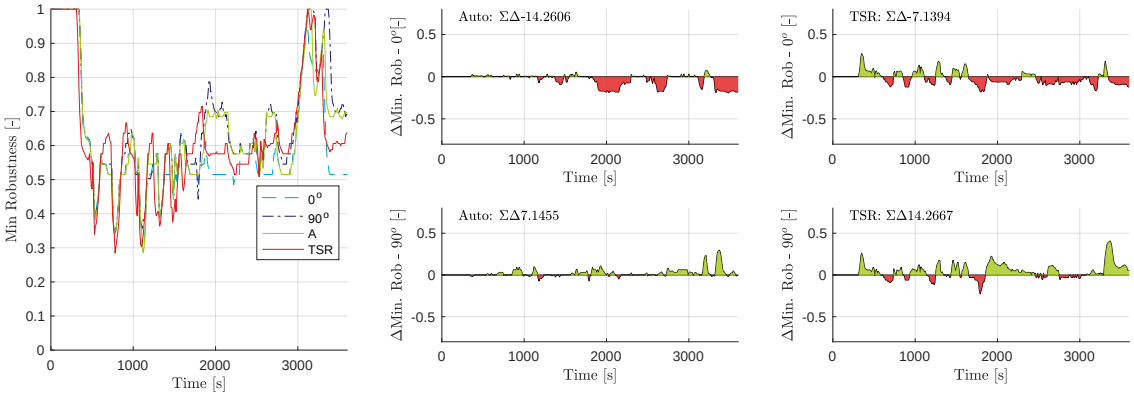


(b) Perturbation Location 1

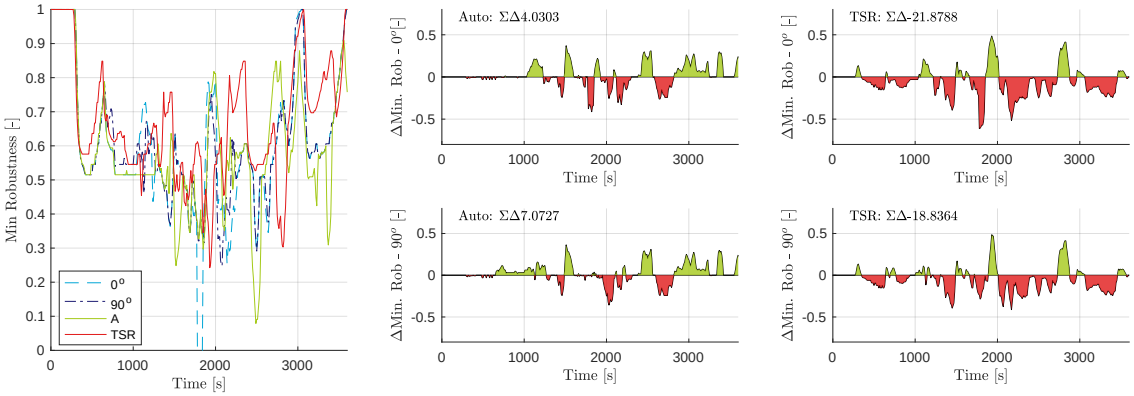


(c) Perturbation Location 2

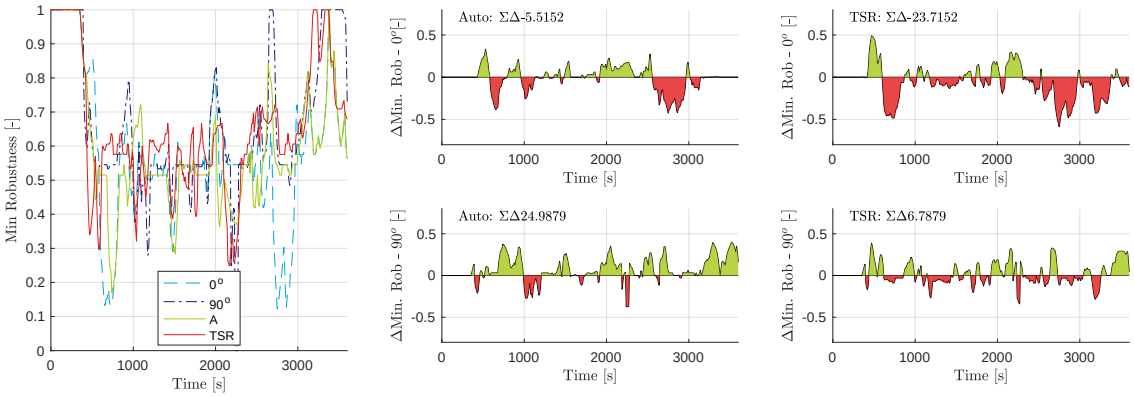
Figure D.29: Average sector robustness low density traffic scenario participant 4



(a) Perturbation Location 0

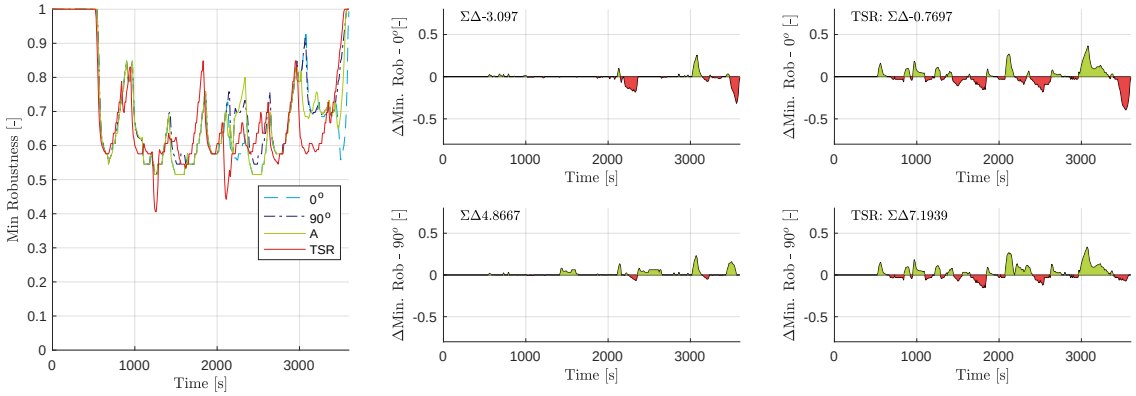


(b) Perturbation Location 1

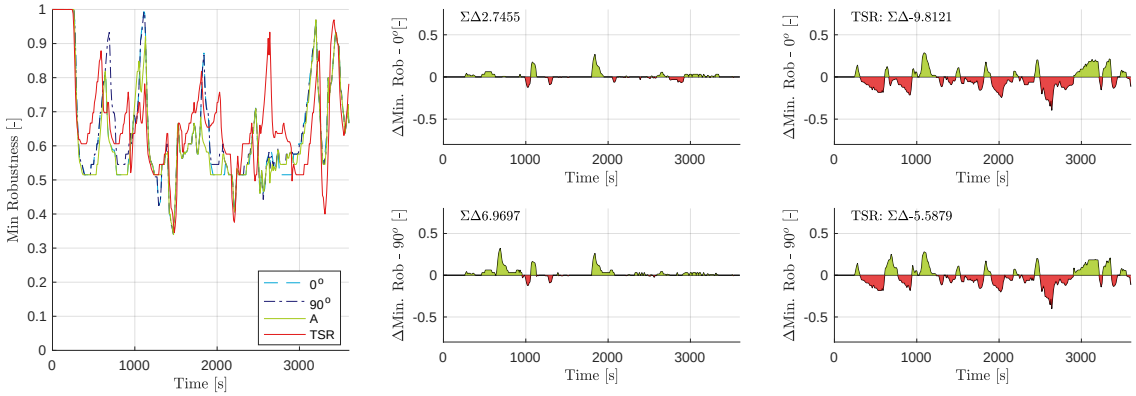


(c) Perturbation Location 2

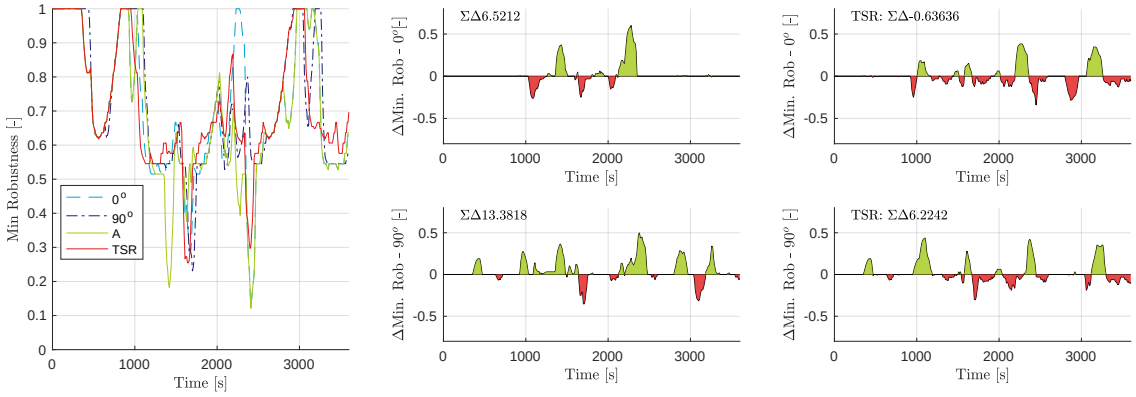
Figure D.30: Minimum sector robustness high density traffic scenario participant 4



(a) Perturbation Location 0

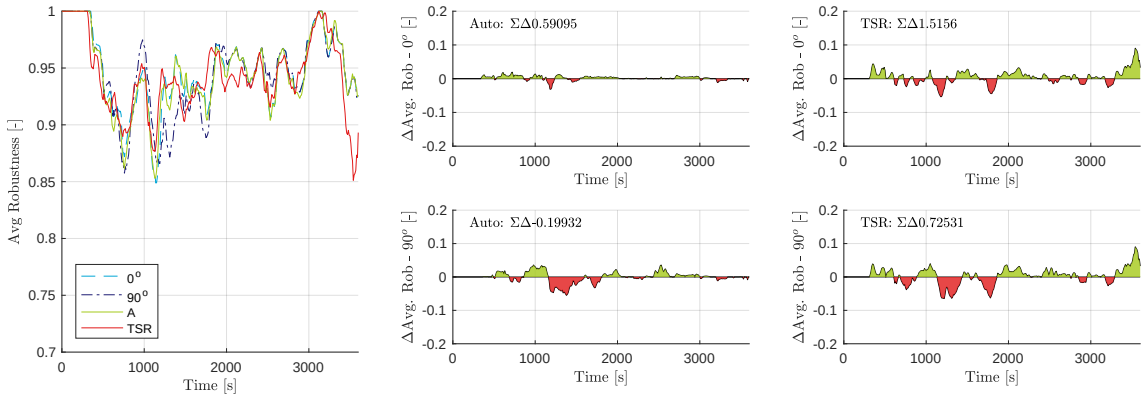


(b) Perturbation Location 1

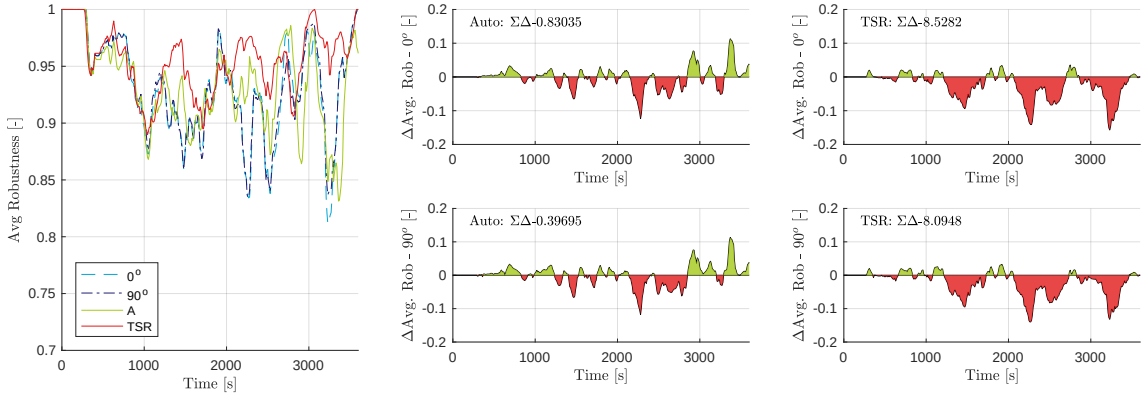


(c) Perturbation Location 2

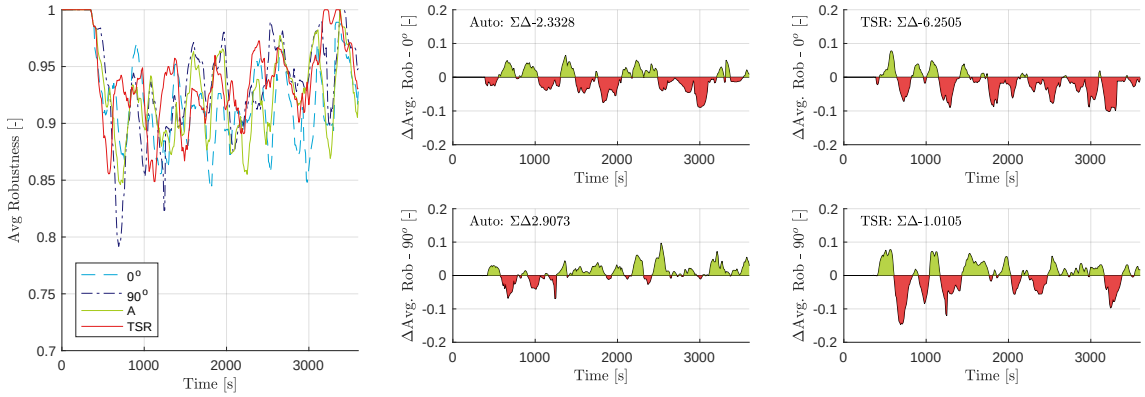
Figure D.31: Minimum sector robustness low density traffic scenario participant 4



(a) Perturbation Location 0

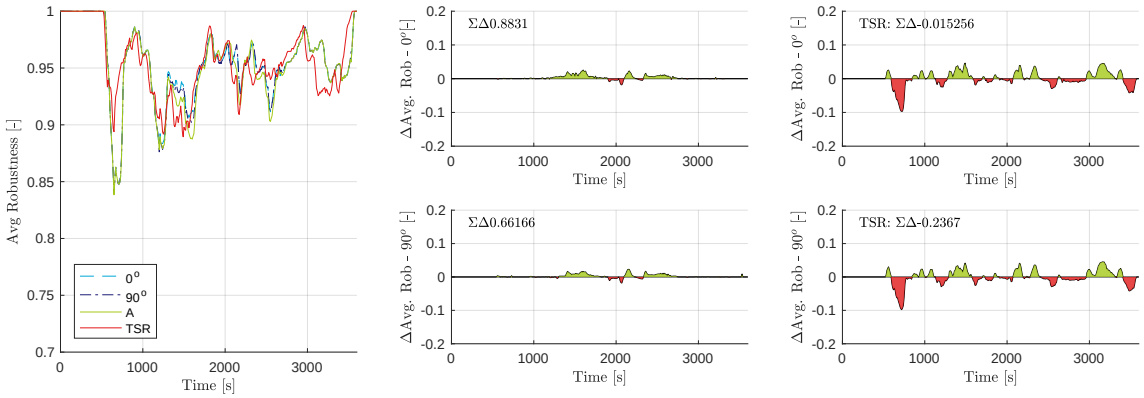


(b) Perturbation Location 1

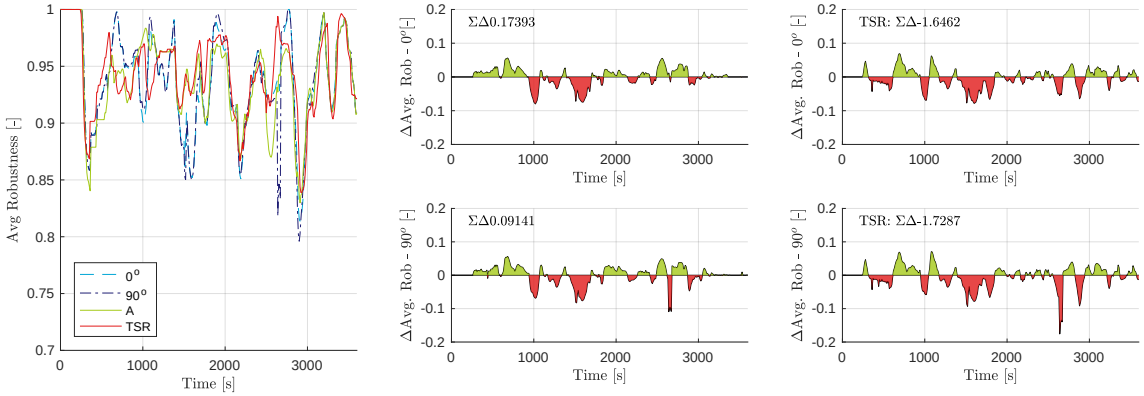


(c) Perturbation Location 2

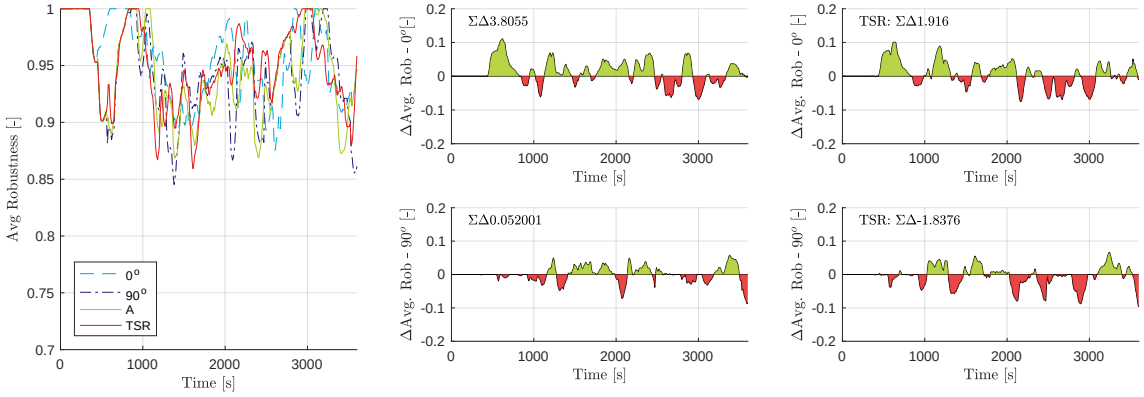
Figure D.32: Average sector robustness high density traffic scenario participant 5



(a) Perturbation Location 0

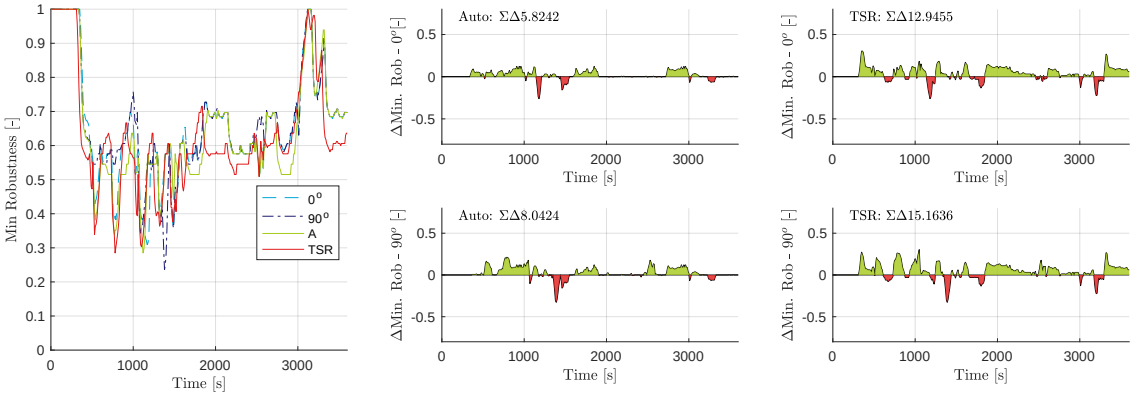


(b) Perturbation Location 1

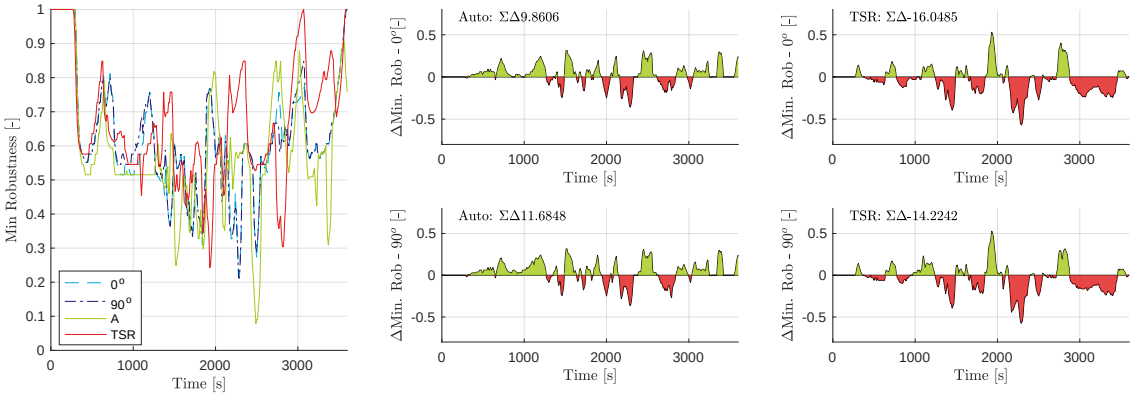


(c) Perturbation Location 2

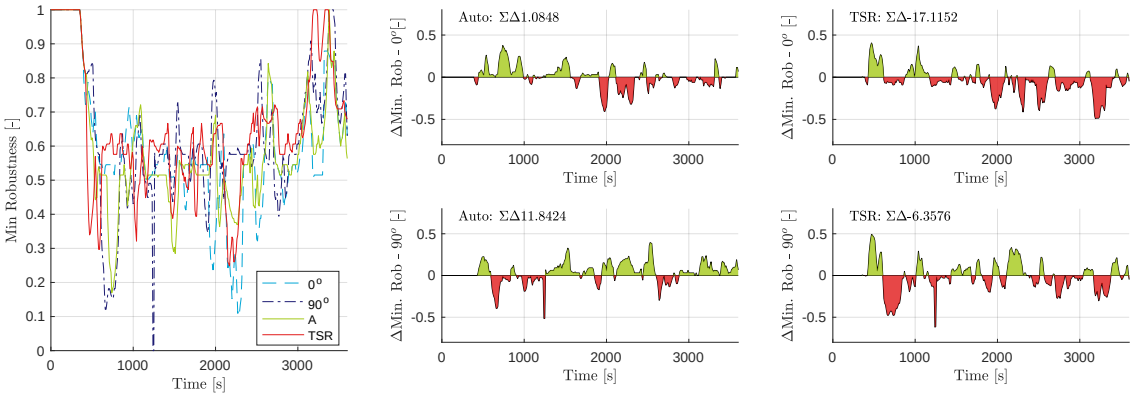
Figure D.33: Average sector robustness low density traffic scenario participant 5



(a) Perturbation Location 0

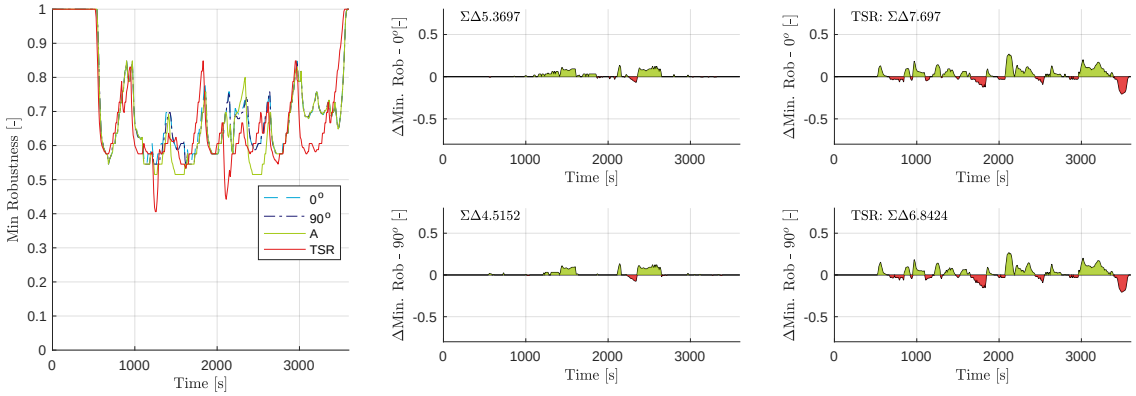


(b) Perturbation Location 1

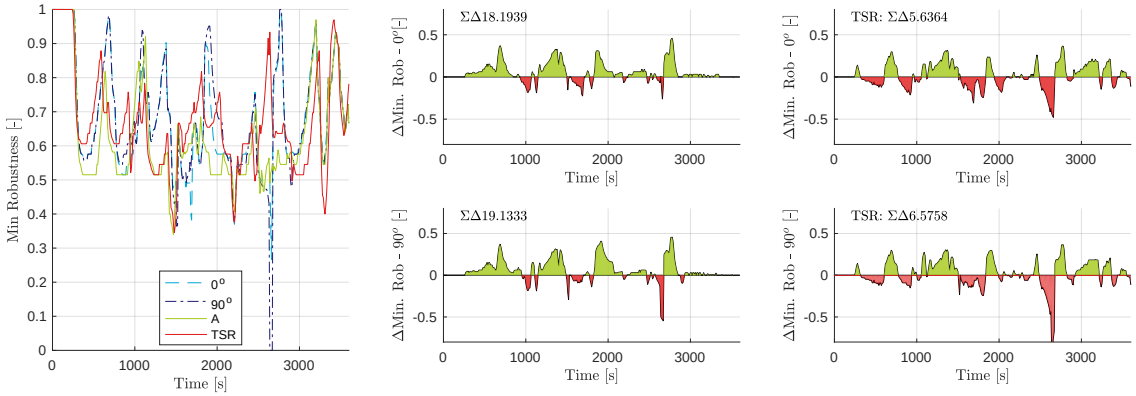


(c) Perturbation Location 2

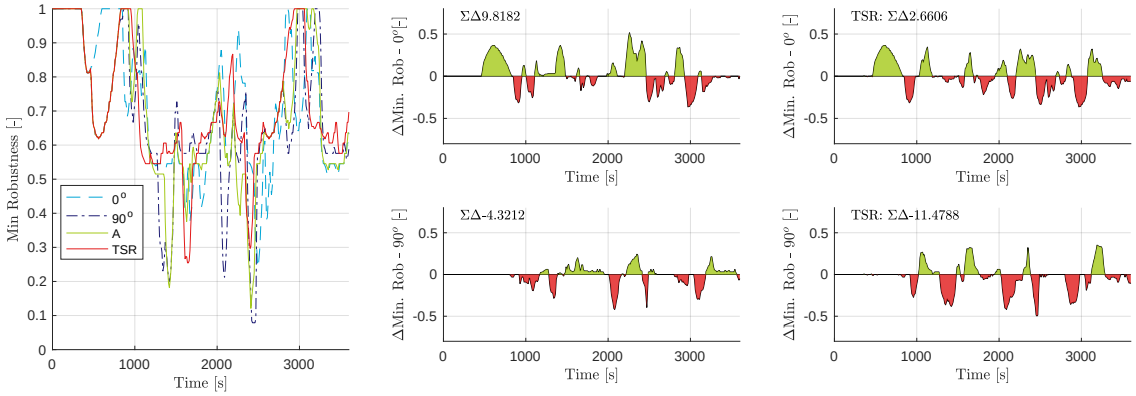
Figure D.34: Minimum sector robustness high density traffic scenario participant 5



(a) Perturbation Location 0



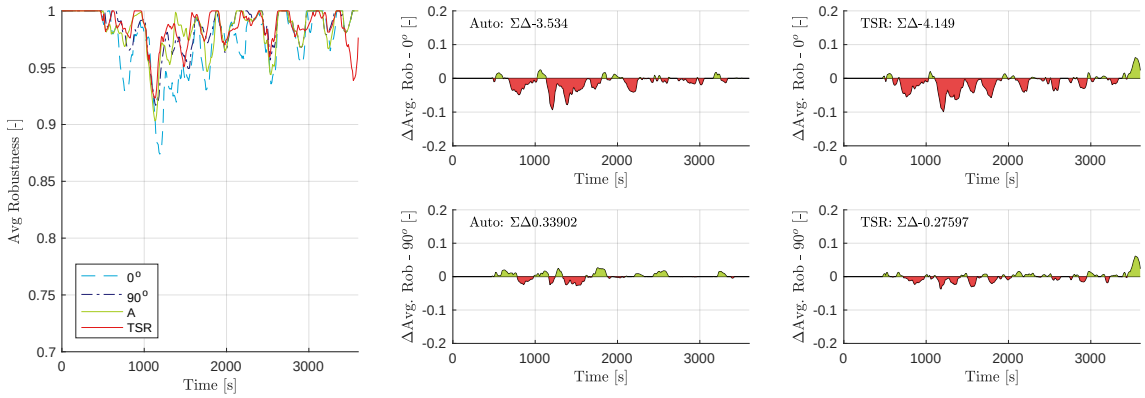
(b) Perturbation Location 1



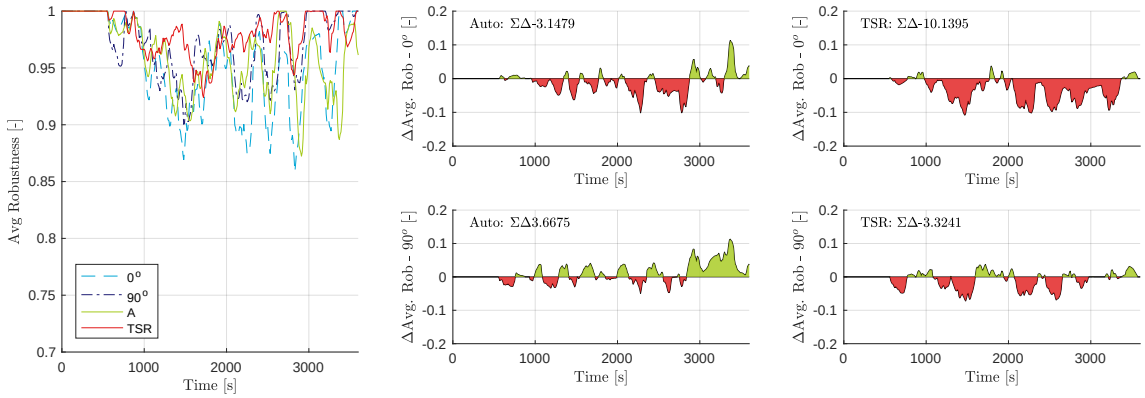
(c) Perturbation Location 2

Figure D.35: Minimum sector robustness low density traffic scenario participant 5

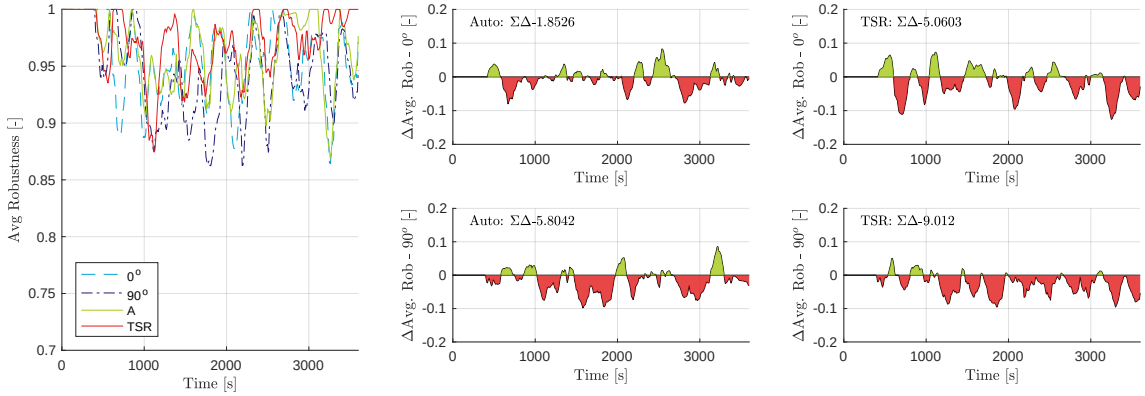
D.4. Participant's average and minimum robustness over time - without perturbation



(a) Perturbation Location 0

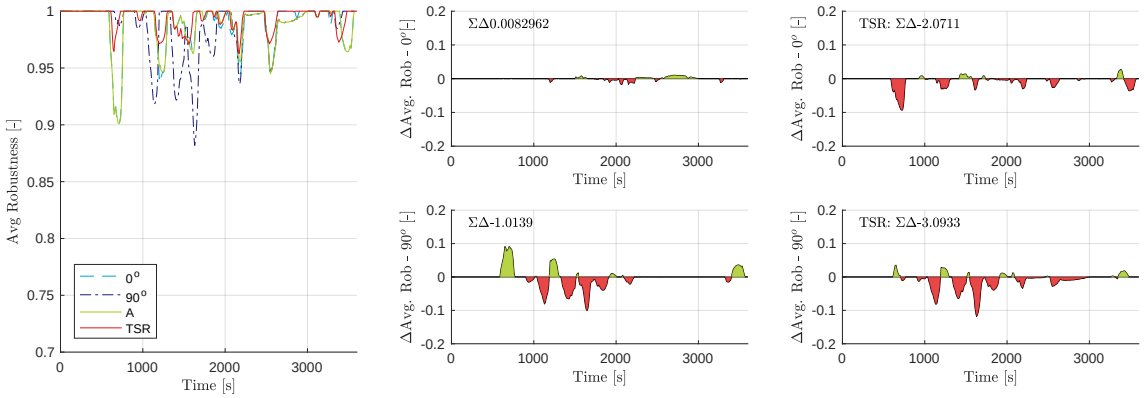


(b) Perturbation Location 1

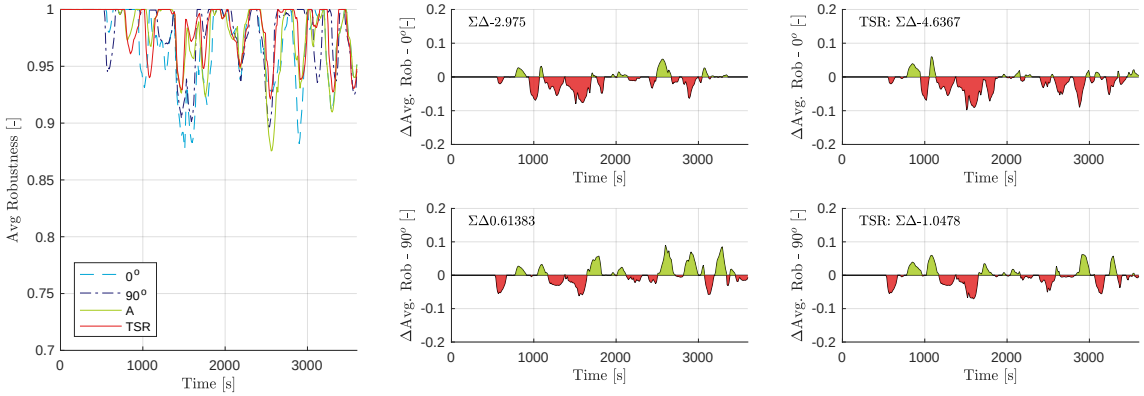


(c) Perturbation Location 2

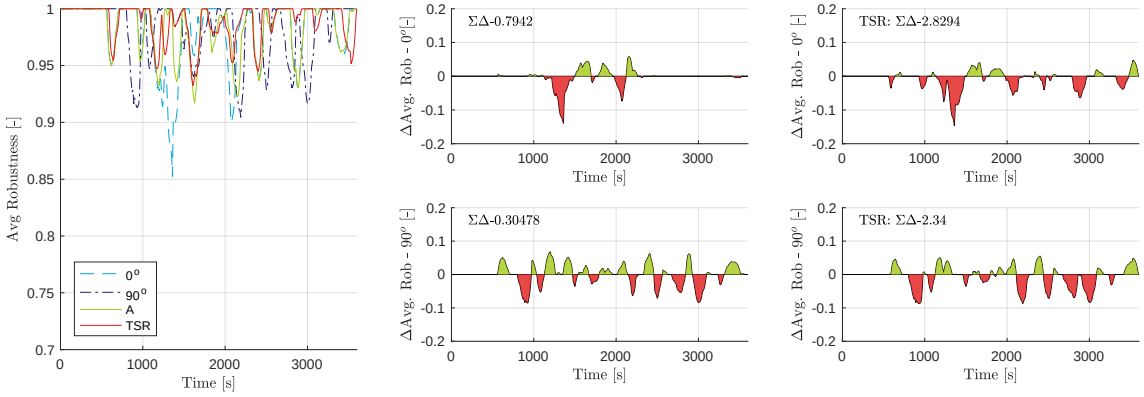
Figure D.36: Average sector robustness (no perturbation) high density traffic scenario participant 1



(a) Perturbation Location 0

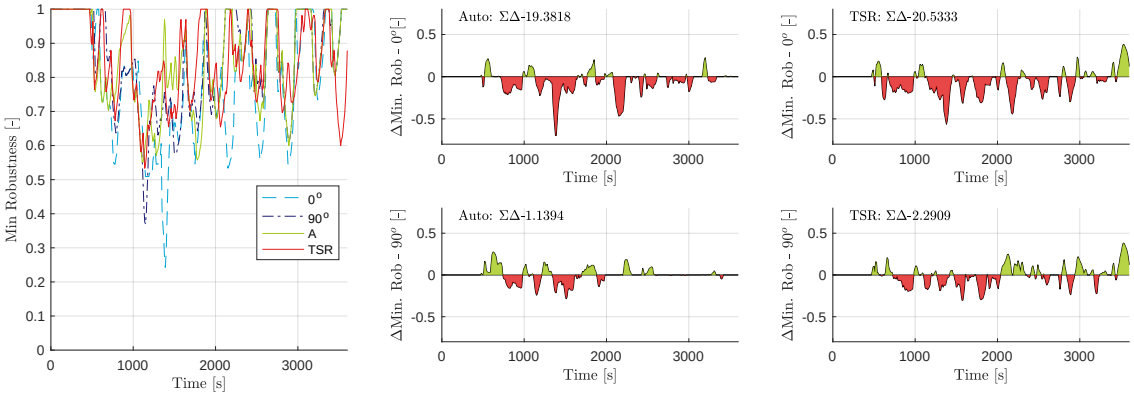


(b) Perturbation Location 1

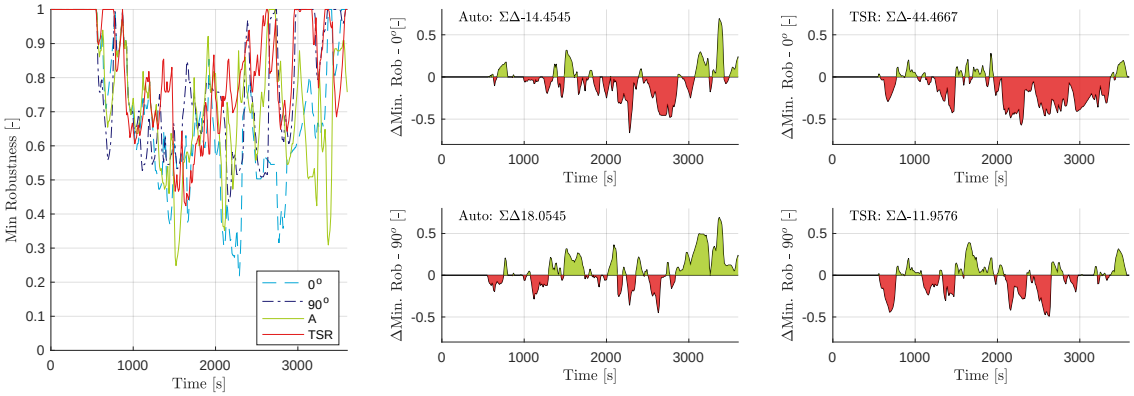


(c) Perturbation Location 2

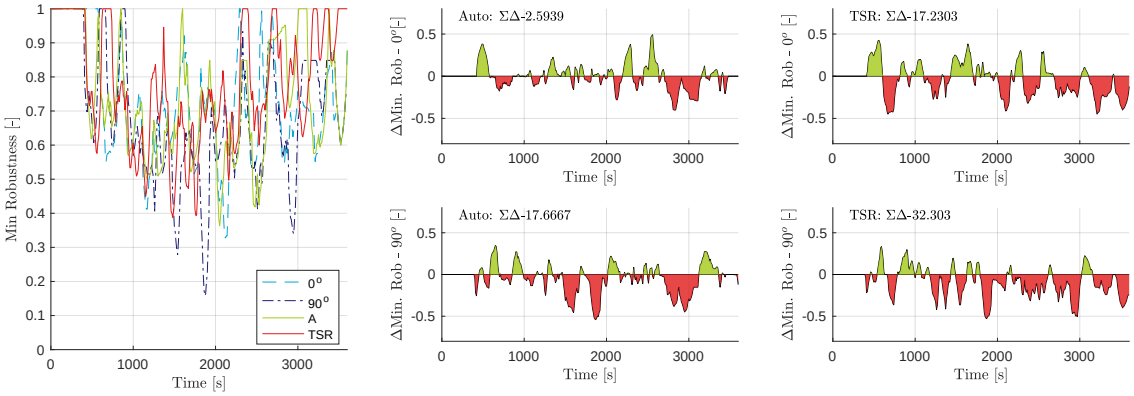
Figure D.37: Average sector robustness (no perturbation) low density traffic scenario participant 1



(a) Perturbation Location 0

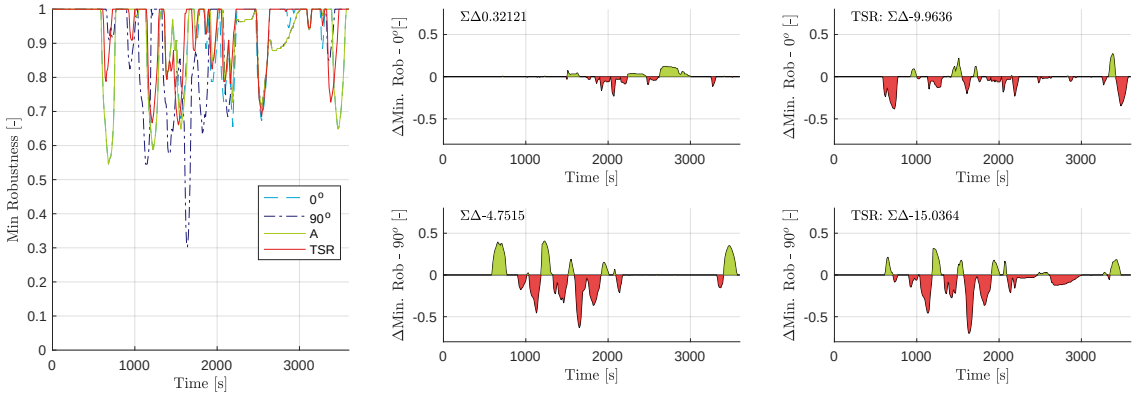


(b) Perturbation Location 1

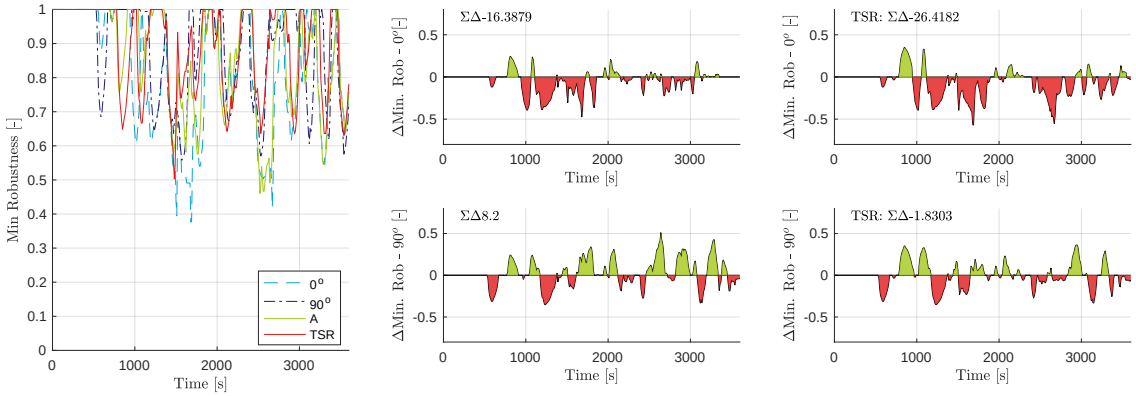


(c) Perturbation Location 2

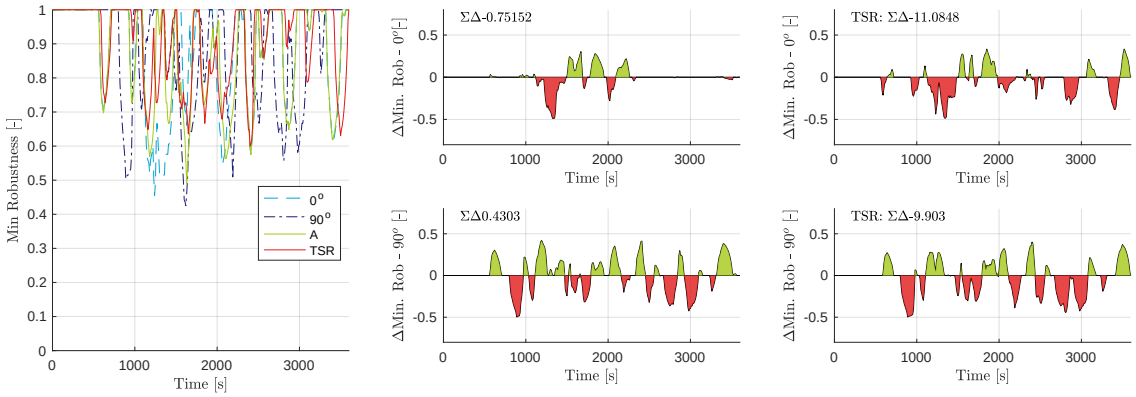
Figure D.38: Minimum sector robustness (no perturbation) high density traffic scenario participant 1



(a) Perturbation Location 0

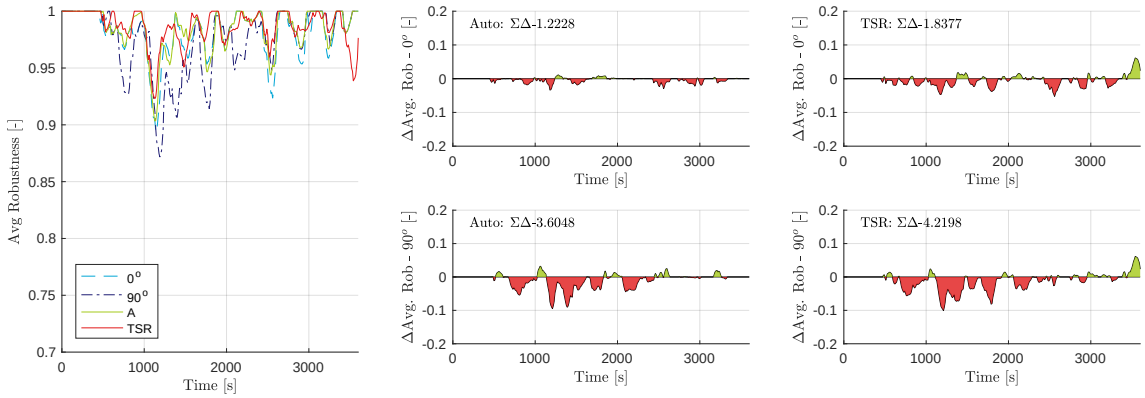


(b) Perturbation Location 1

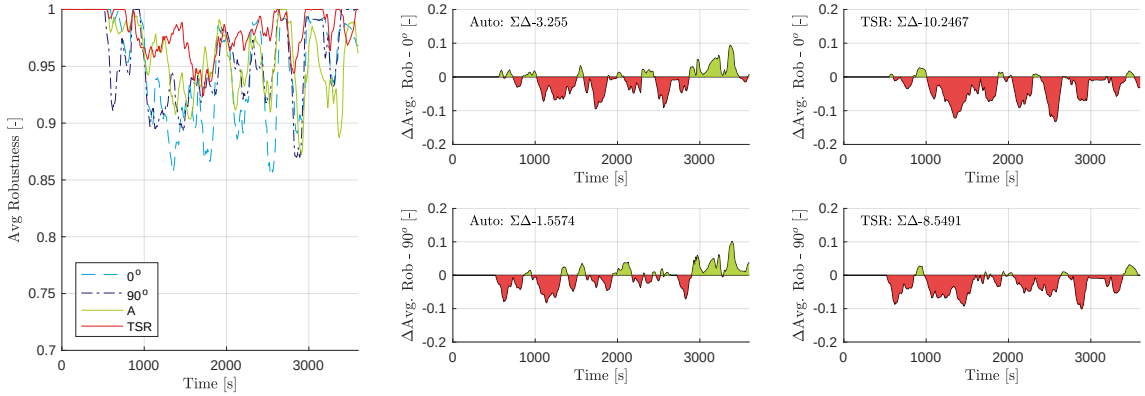


(c) Perturbation Location 2

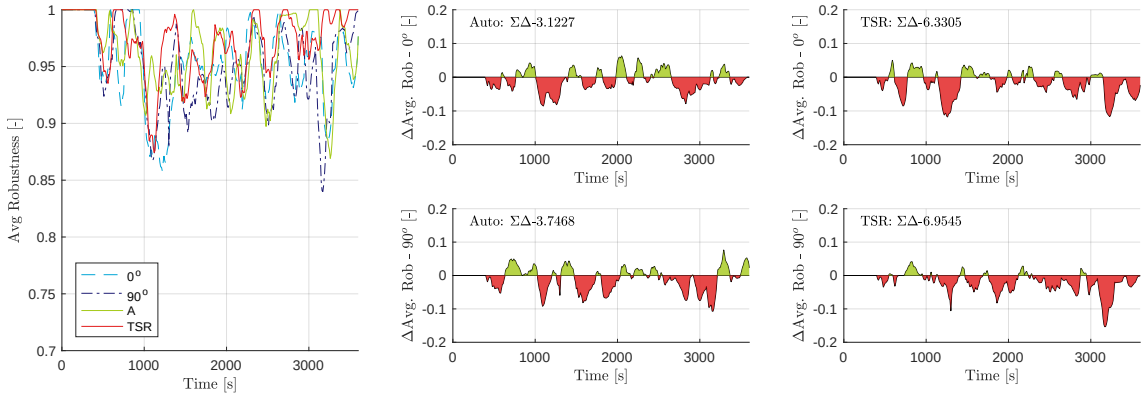
Figure D.39: Minimum sector robustness (no perturbation) low density traffic scenario participant 1



(a) Perturbation Location 0

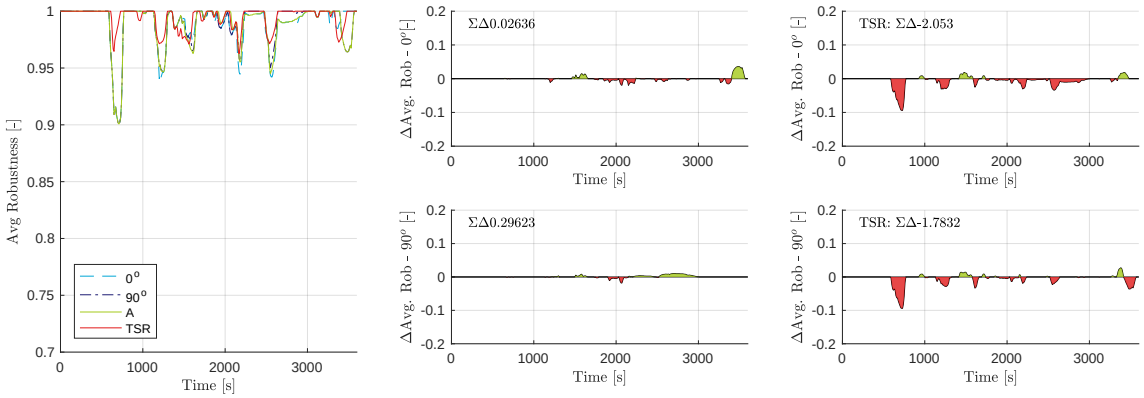


(b) Perturbation Location 1

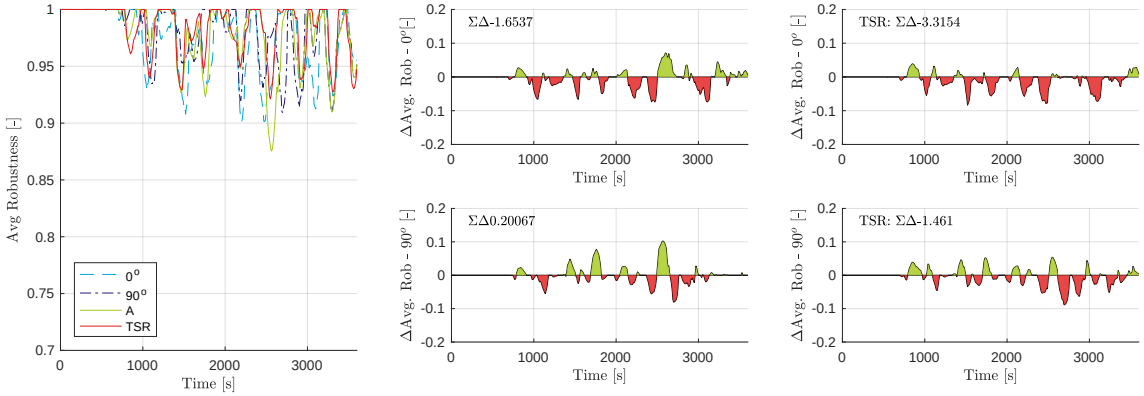


(c) Perturbation Location 2

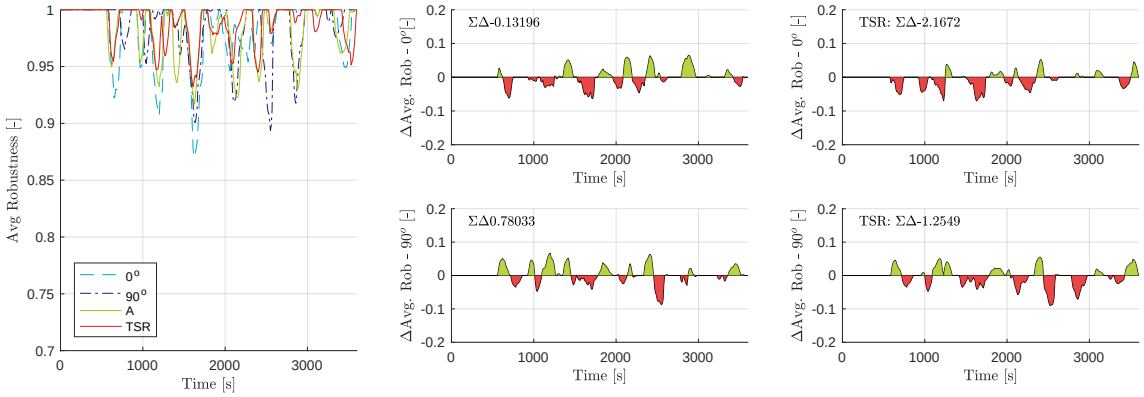
Figure D.40: Average sector robustness (no perturbation) high density traffic scenario participant 2



(a) Perturbation Location 0

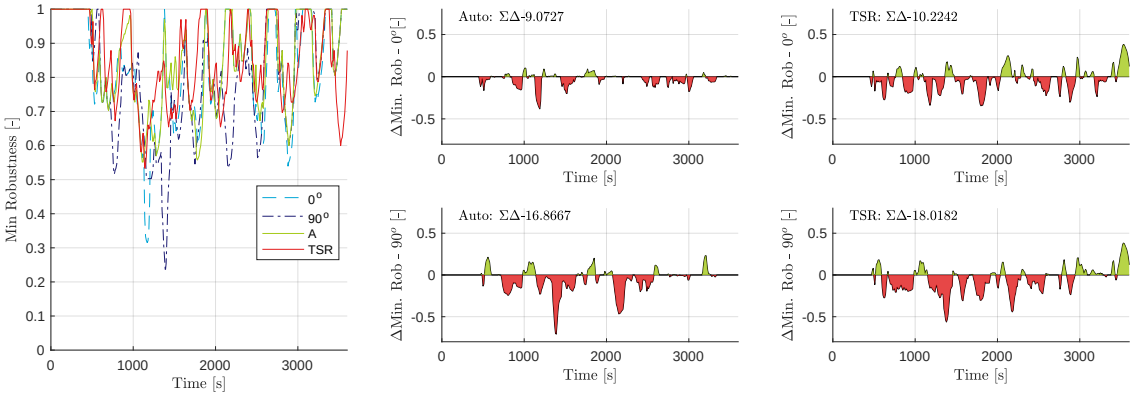


(b) Perturbation Location 1

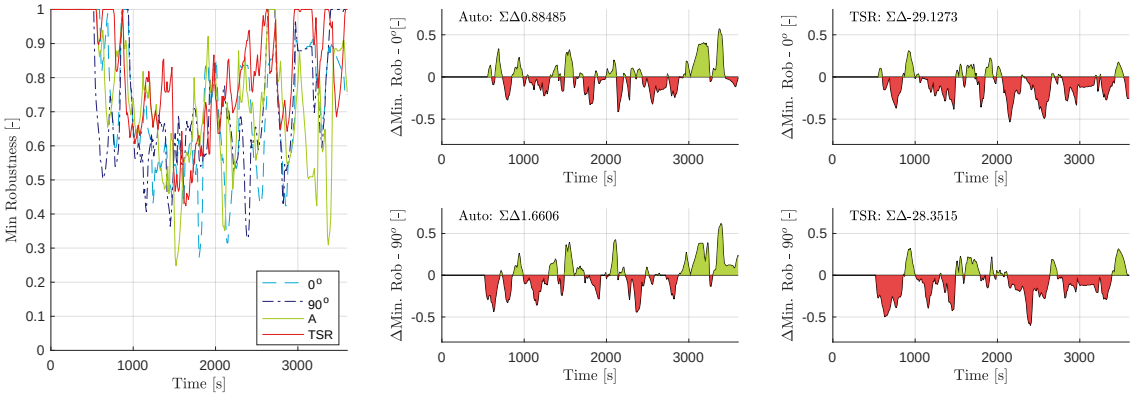


(c) Perturbation Location 2

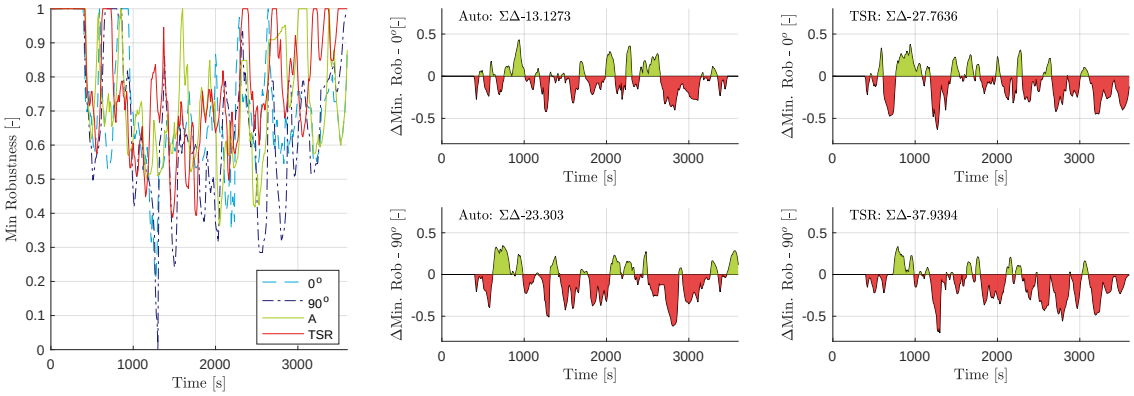
Figure D.41: Average sector robustness (no perturbation) low density traffic scenario participant 2



(a) Perturbation Location 0

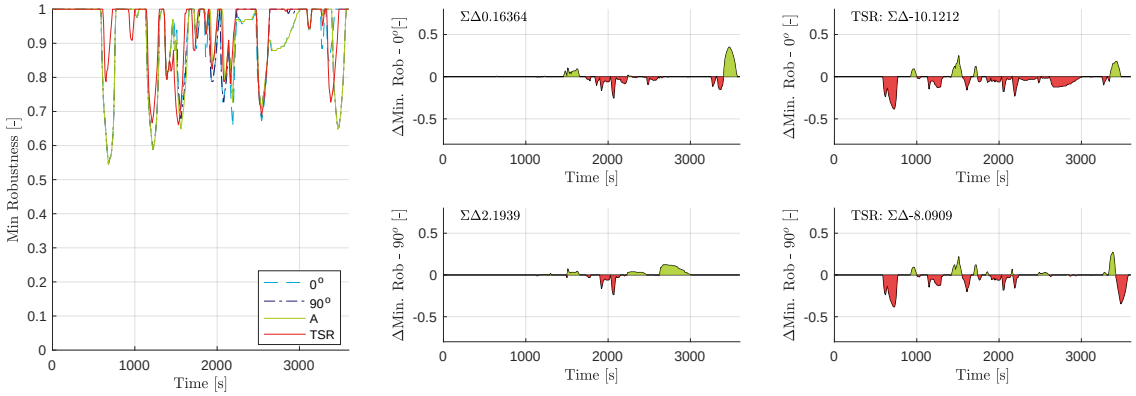


(b) Perturbation Location 1

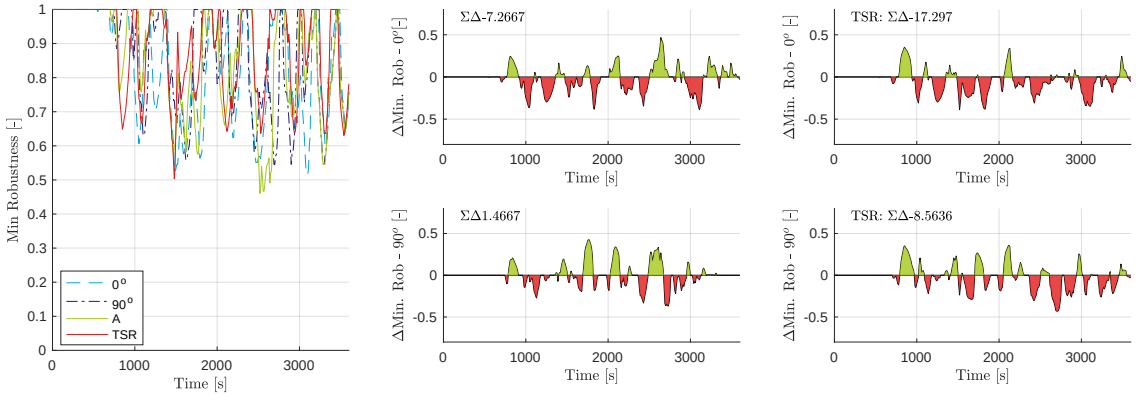


(c) Perturbation Location 2

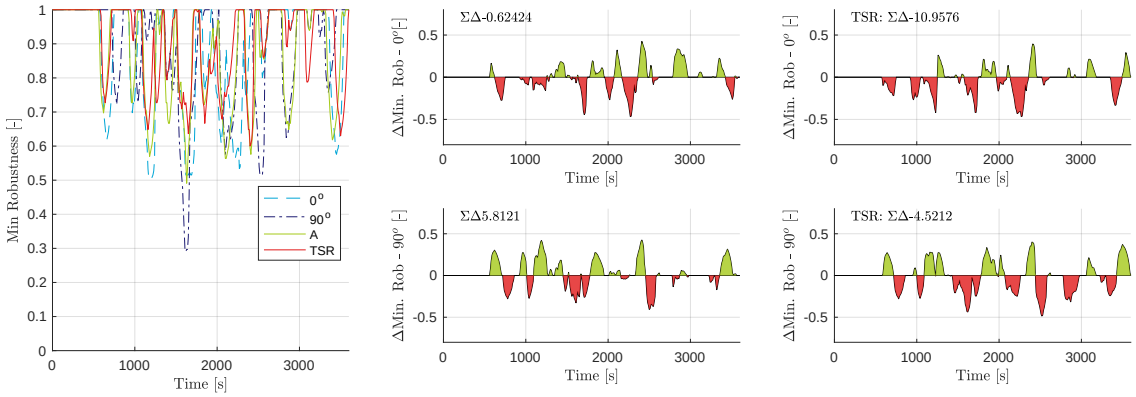
Figure D.42: Minimum sector robustness (no perturbation) high density traffic scenario participant 2



(a) Perturbation Location 0

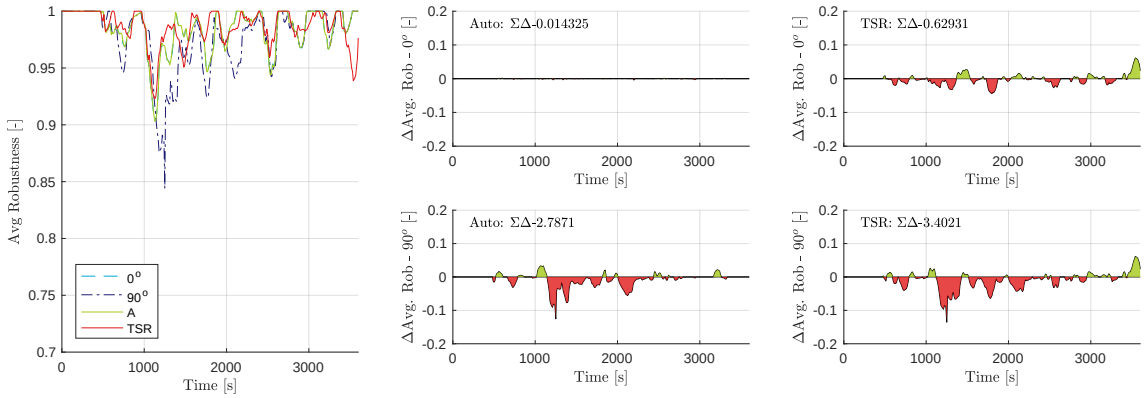


(b) Perturbation Location 1

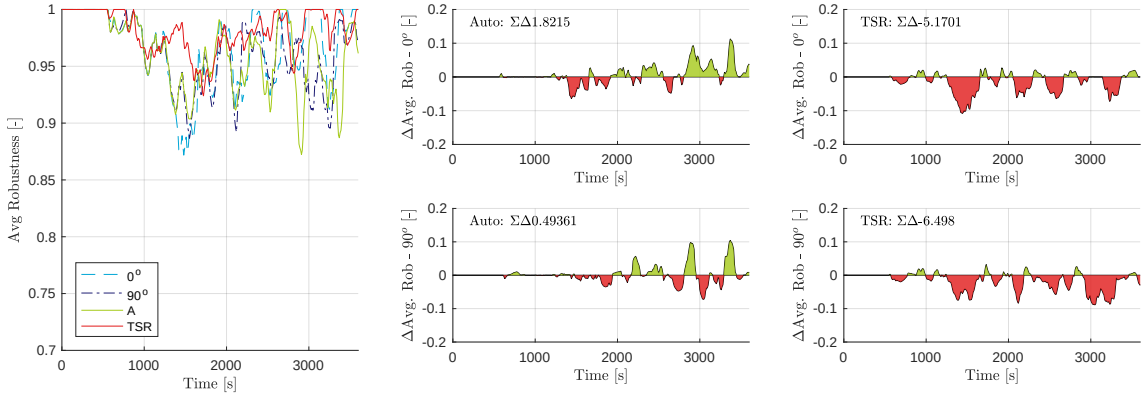


(c) Perturbation Location 2

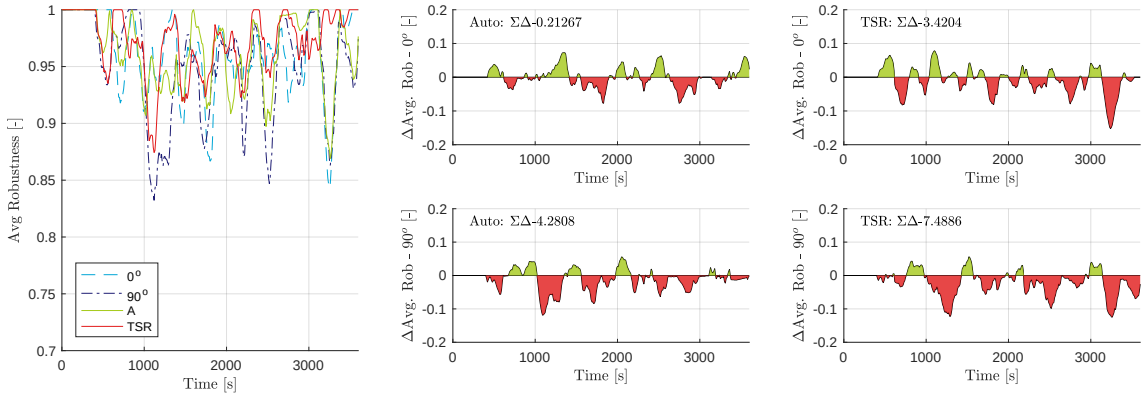
Figure D.43: Minimum sector robustness (no perturbation) low density traffic scenario participant 2



(a) Perturbation Location 0

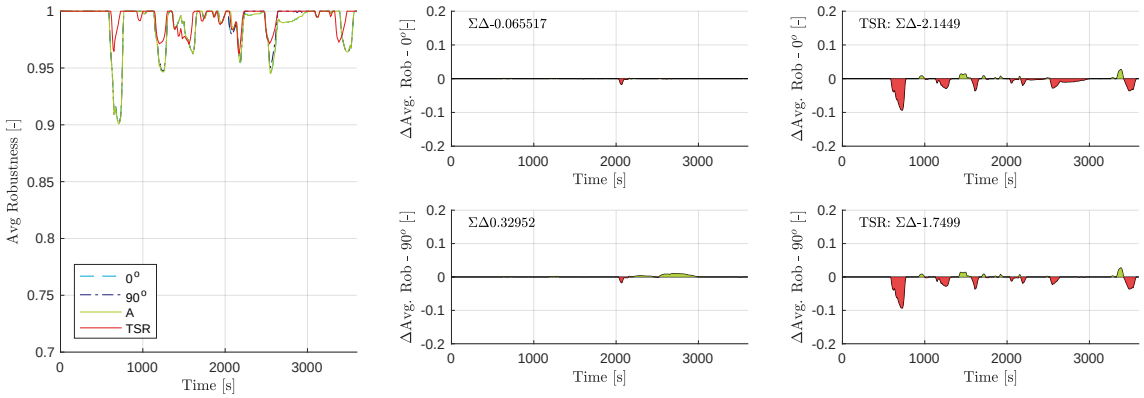


(b) Perturbation Location 1

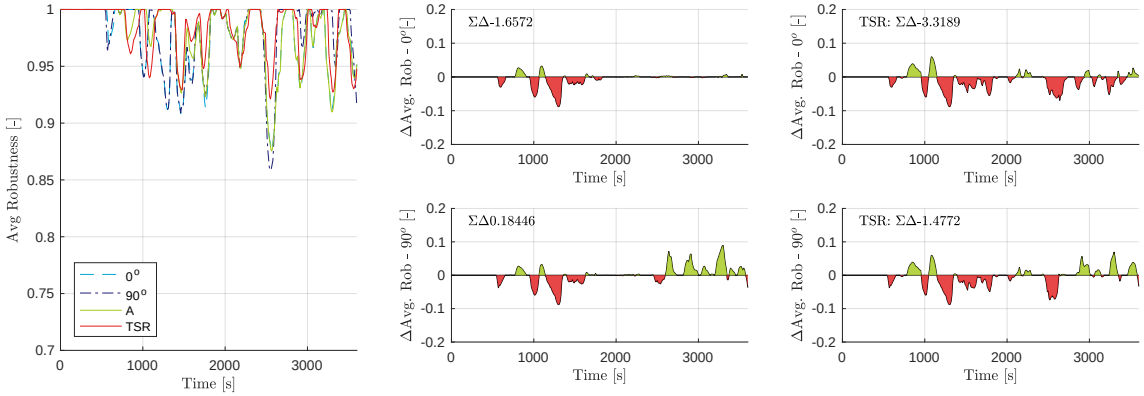


(c) Perturbation Location 2

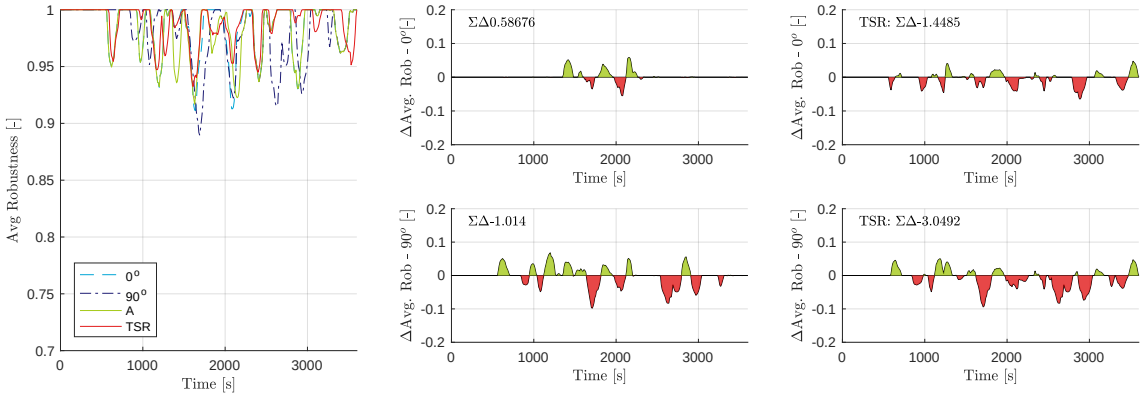
Figure D.44: Average sector robustness (no perturbation) high density traffic scenario participant 3



(a) Perturbation Location 0

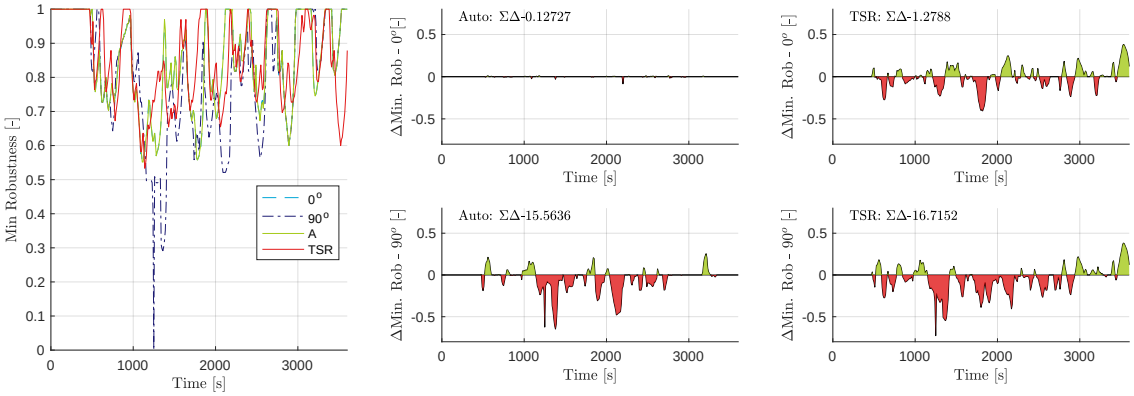


(b) Perturbation Location 1

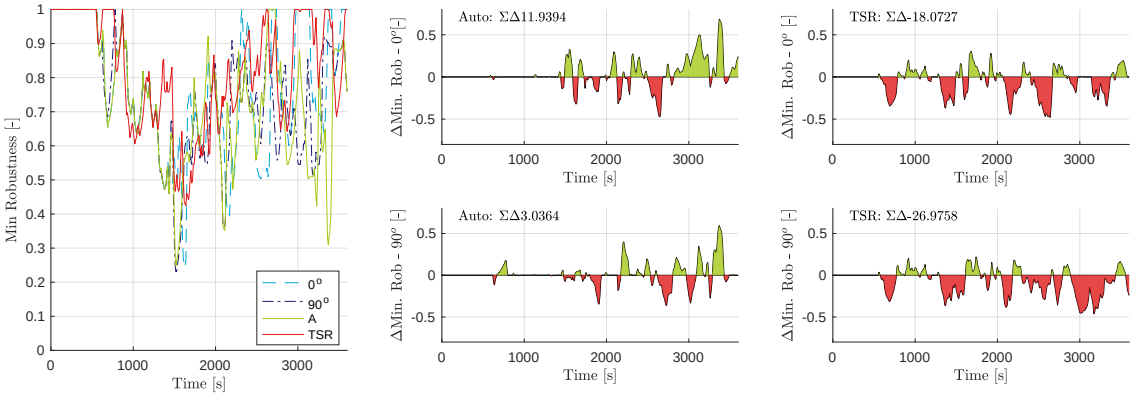


(c) Perturbation Location 2

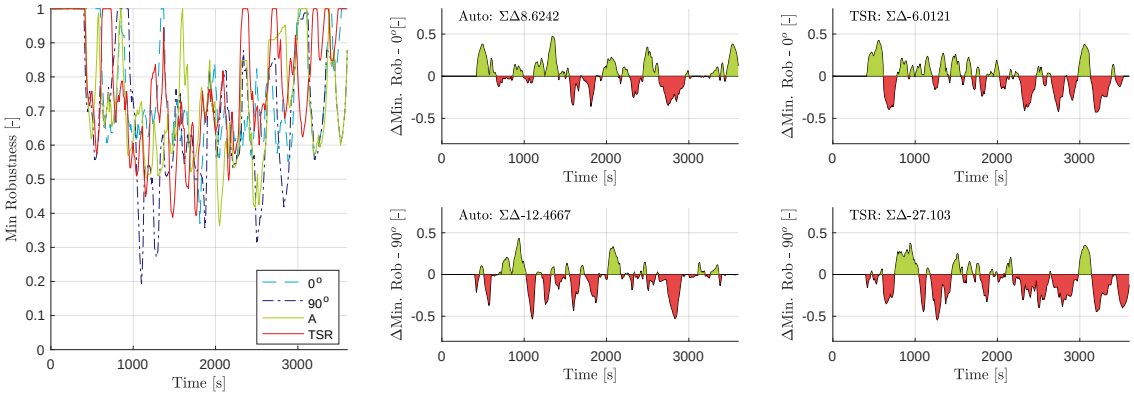
Figure D.45: Average sector robustness (no perturbation) low density traffic scenario participant 3



(a) Perturbation Location 0

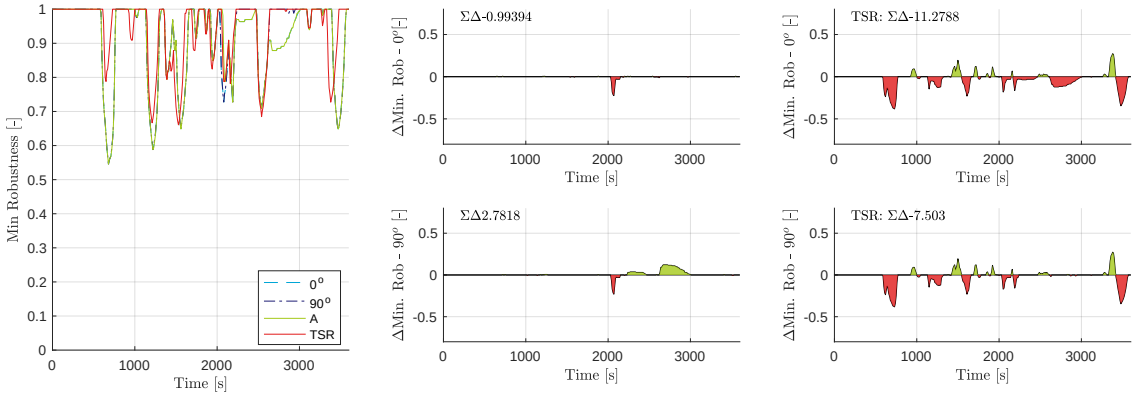


(b) Perturbation Location 1

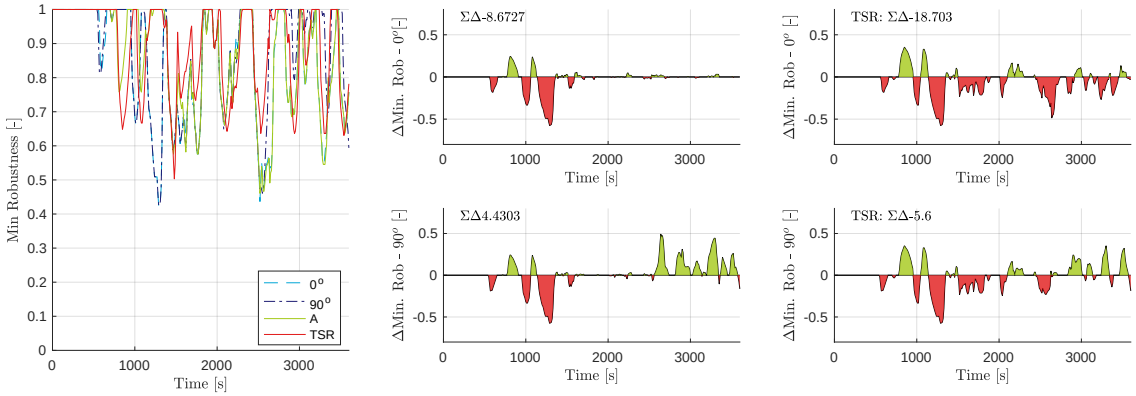


(c) Perturbation Location 2

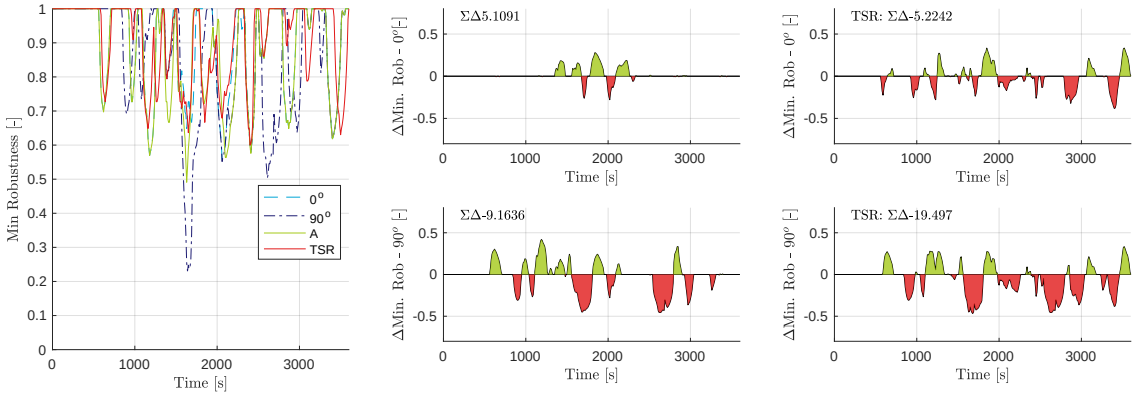
Figure D.46: Minimum sector robustness (no perturbation) high density traffic scenario participant 3



(a) Perturbation Location 0

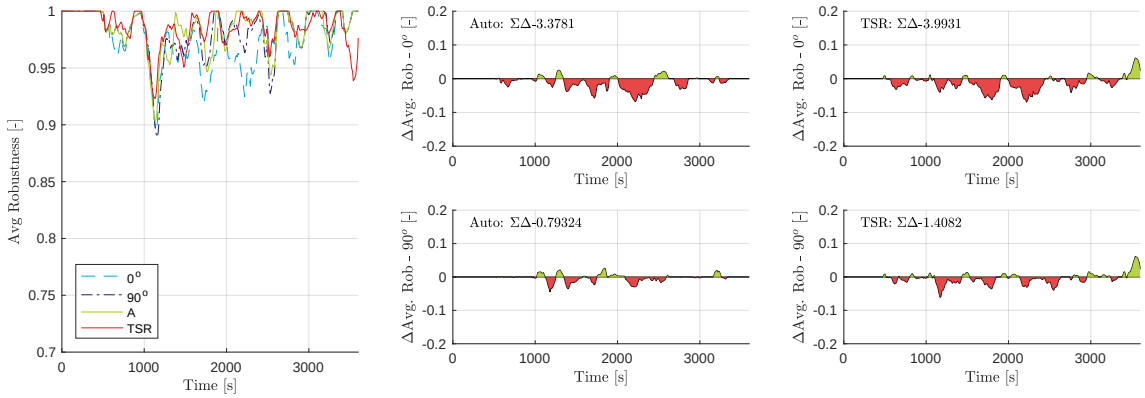


(b) Perturbation Location 1

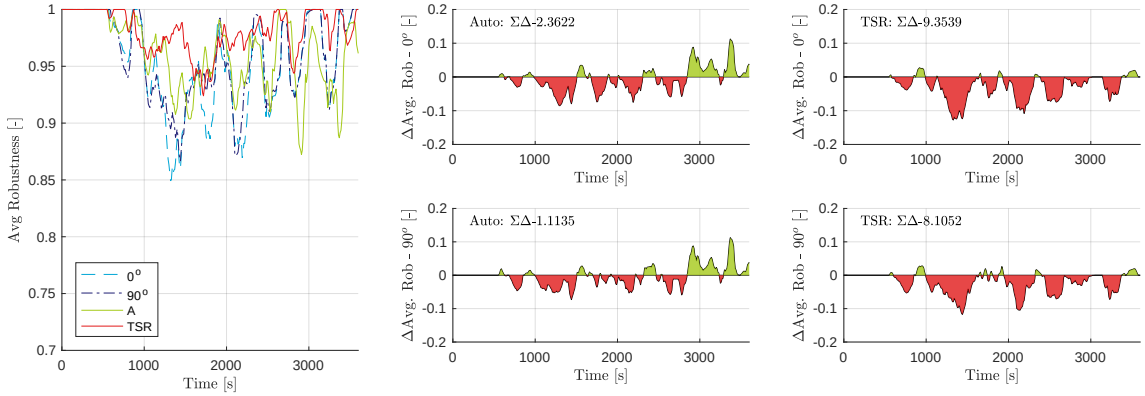


(c) Perturbation Location 2

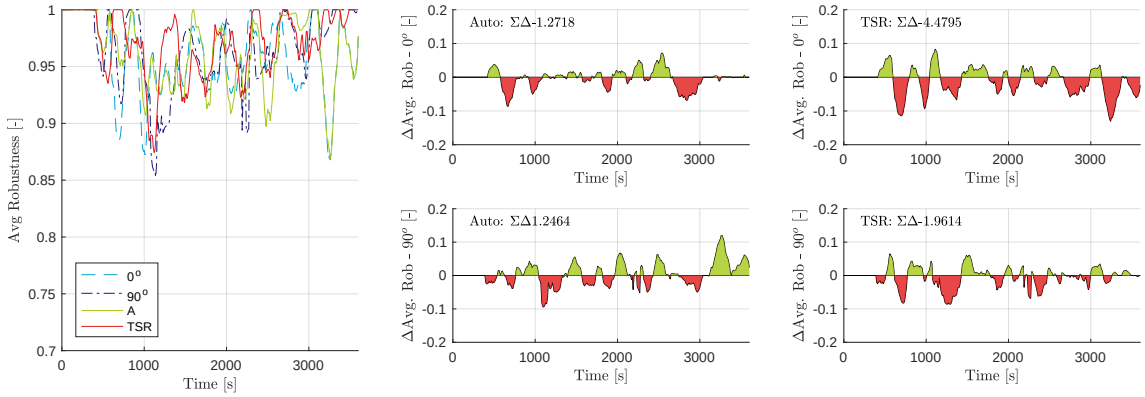
Figure D.47: Minimum sector robustness (no perturbation) low density traffic scenario participant 3



(a) Perturbation Location 0

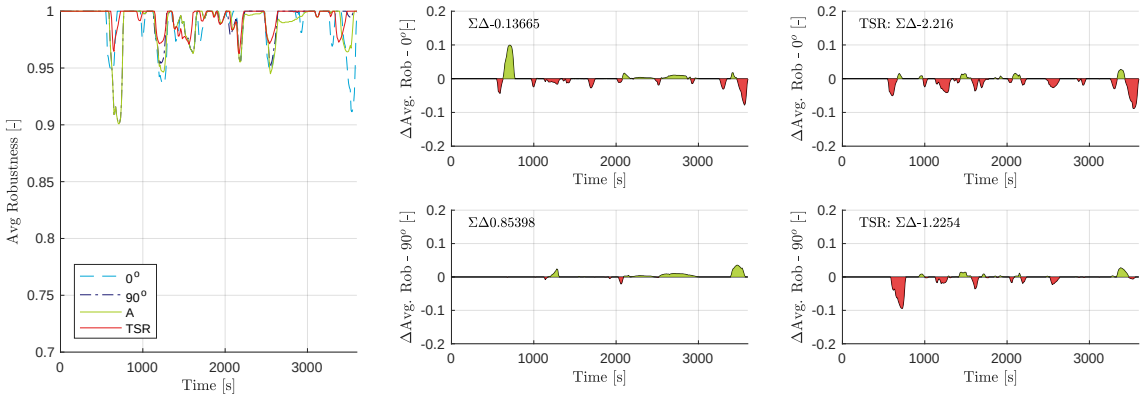


(b) Perturbation Location 1

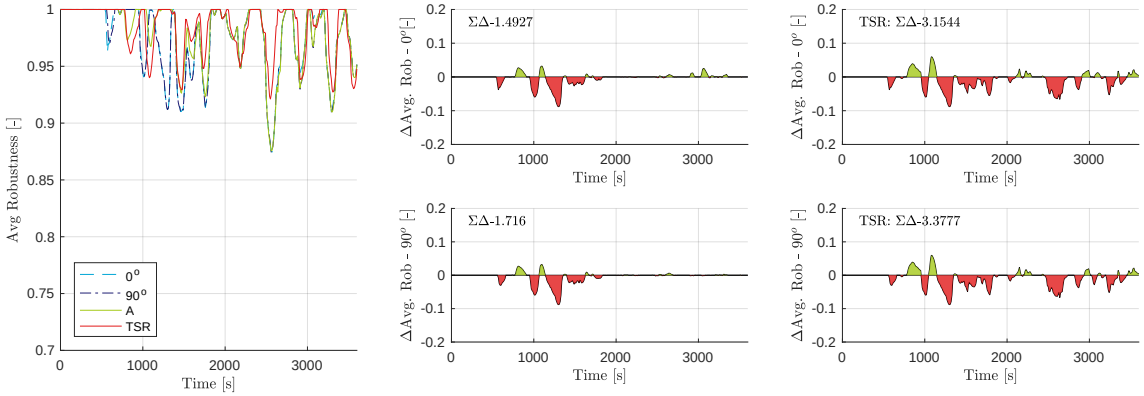


(c) Perturbation Location 2

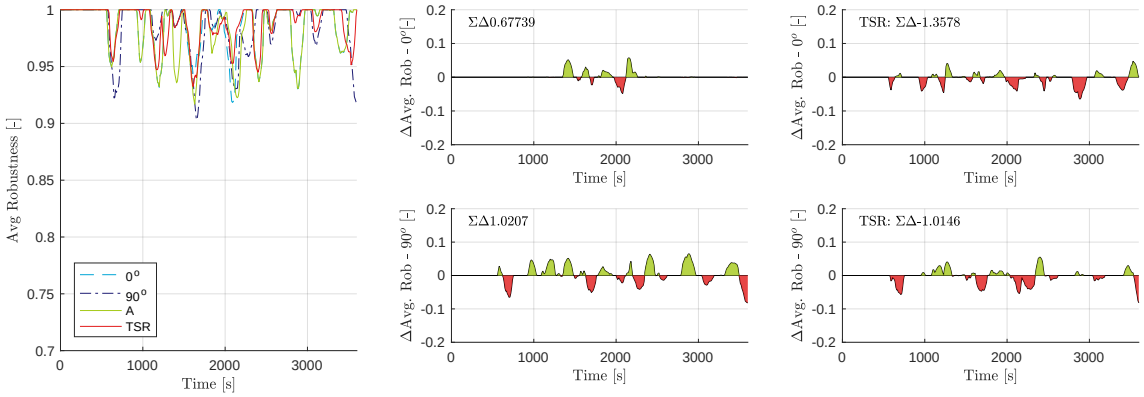
Figure D.48: Average sector robustness (no perturbation) high density traffic scenario participant 4



(a) Perturbation Location 0

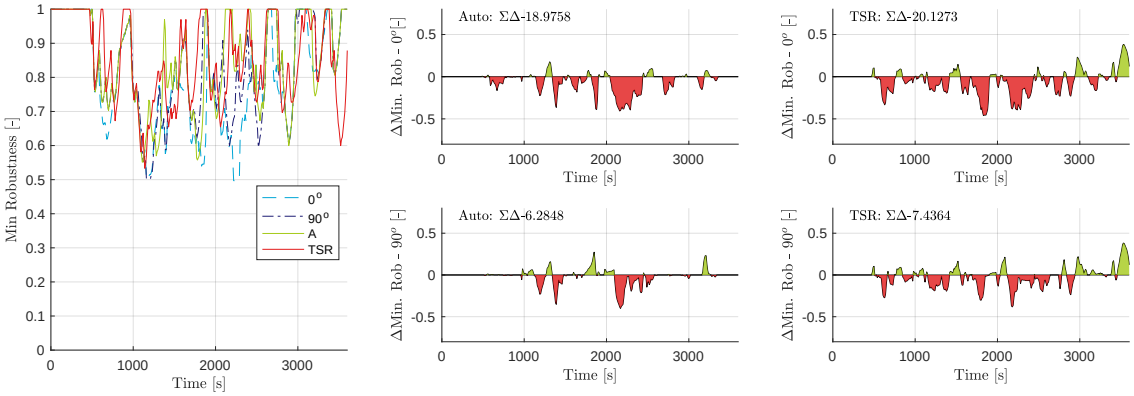


(b) Perturbation Location 1

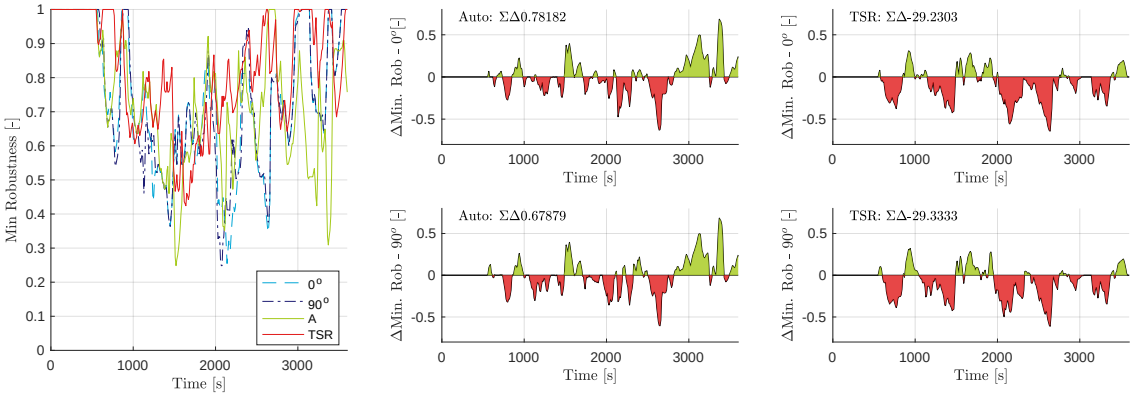


(c) Perturbation Location 2

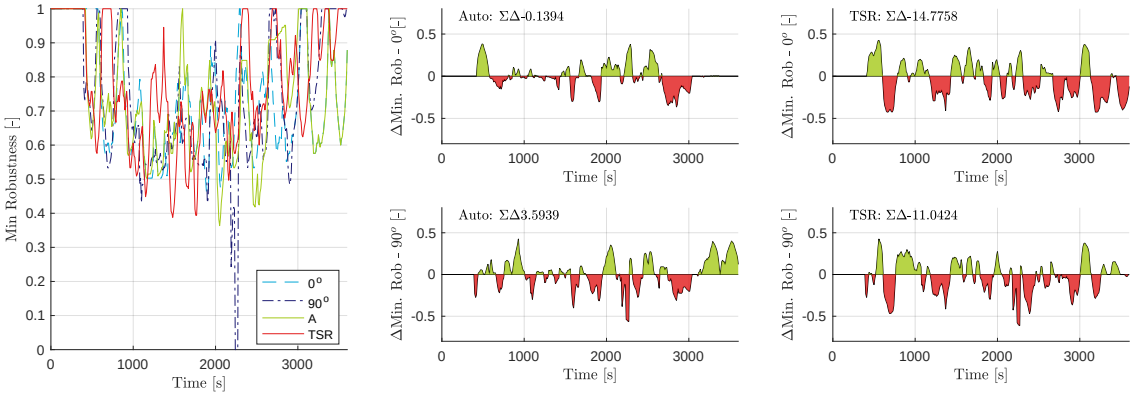
Figure D.49: Average sector robustness (no perturbation) low density traffic scenario participant 4



(a) Perturbation Location 0

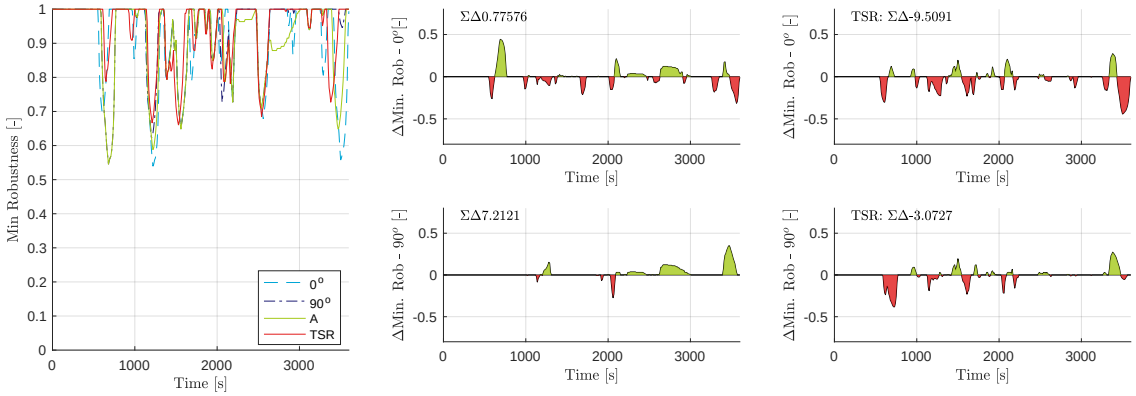


(b) Perturbation Location 1

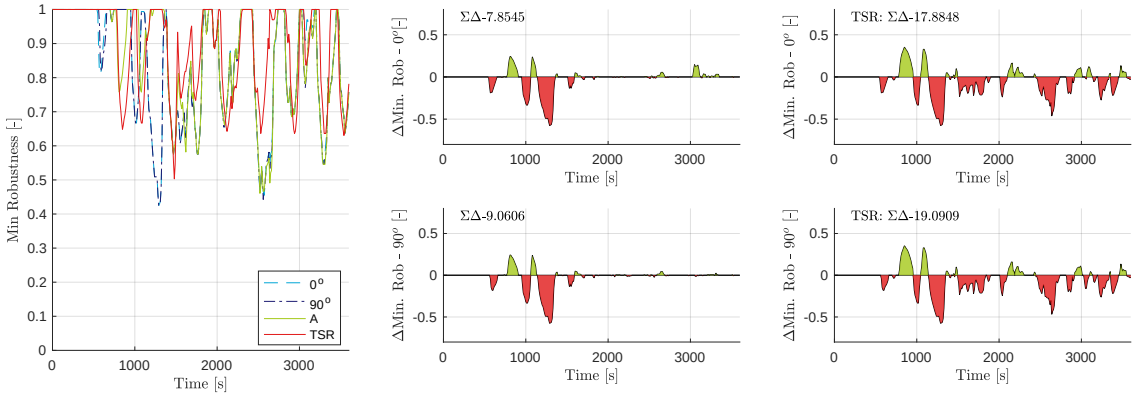


(c) Perturbation Location 2

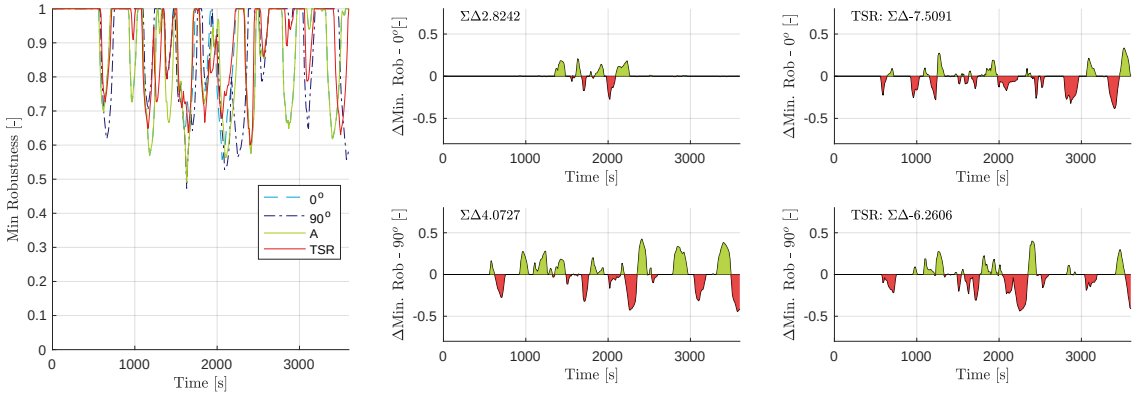
Figure D.50: Minimum sector robustness (no perturbation) high density traffic scenario participant 4



(a) Perturbation Location 0

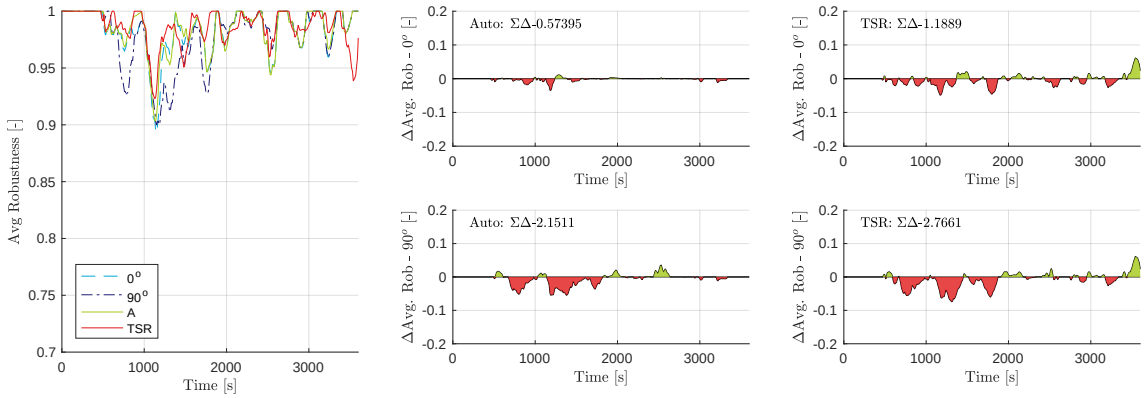


(b) Perturbation Location 1

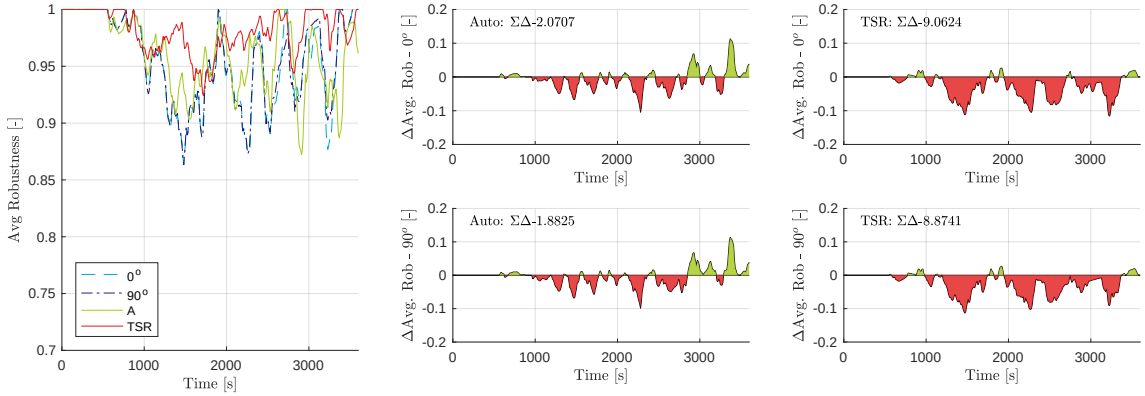


(c) Perturbation Location 2

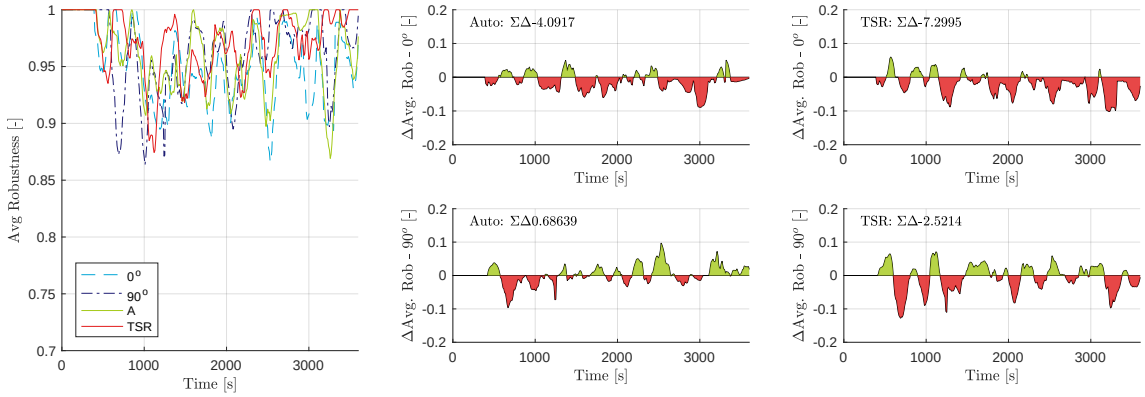
Figure D.51: Minimum sector robustness (no perturbation) low density traffic scenario participant 4



(a) Perturbation Location 0

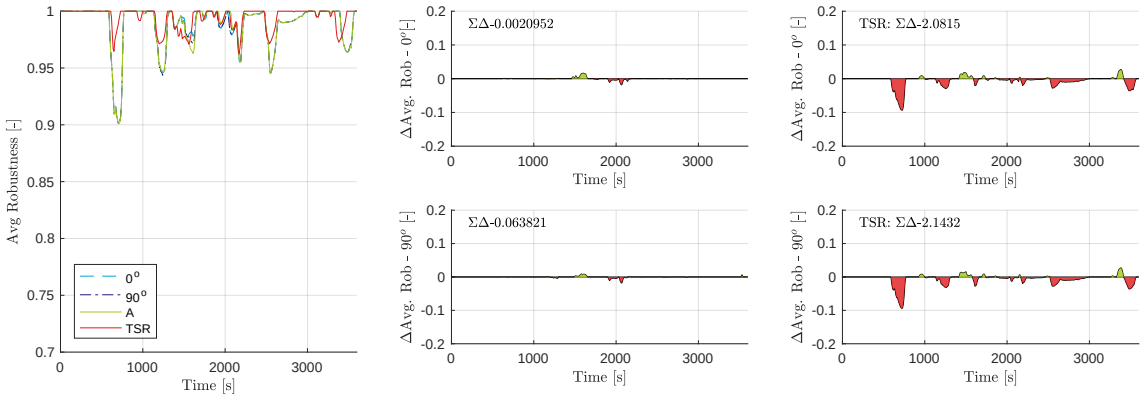


(b) Perturbation Location 1

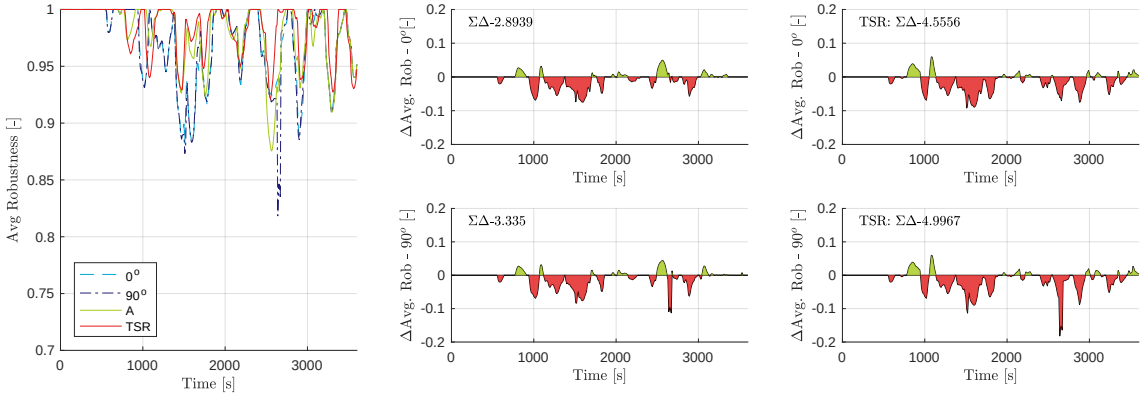


(c) Perturbation Location 2

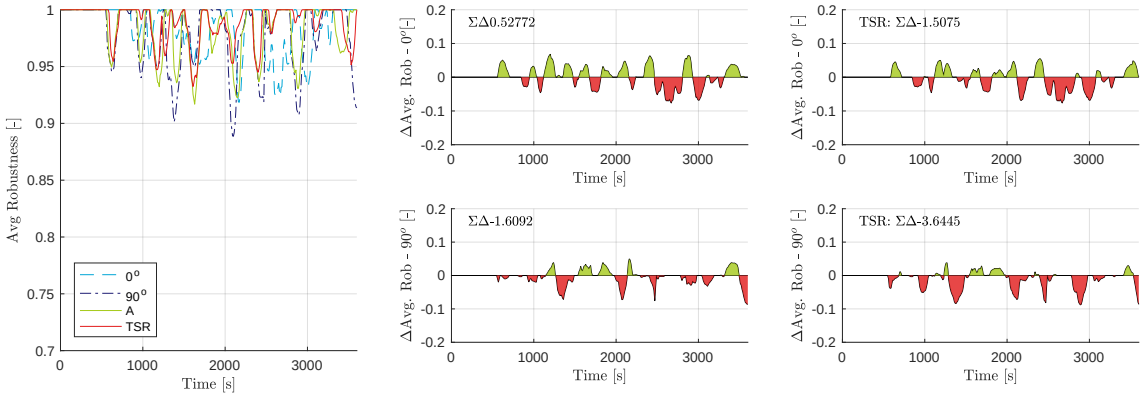
Figure D.52: Average sector robustness (no perturbation) high density traffic scenario participant 5



(a) Perturbation Location 0

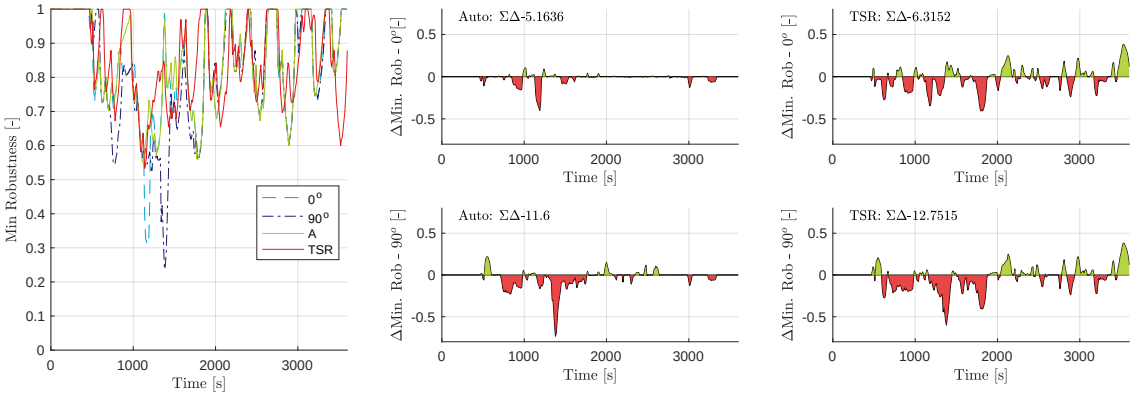


(b) Perturbation Location 1

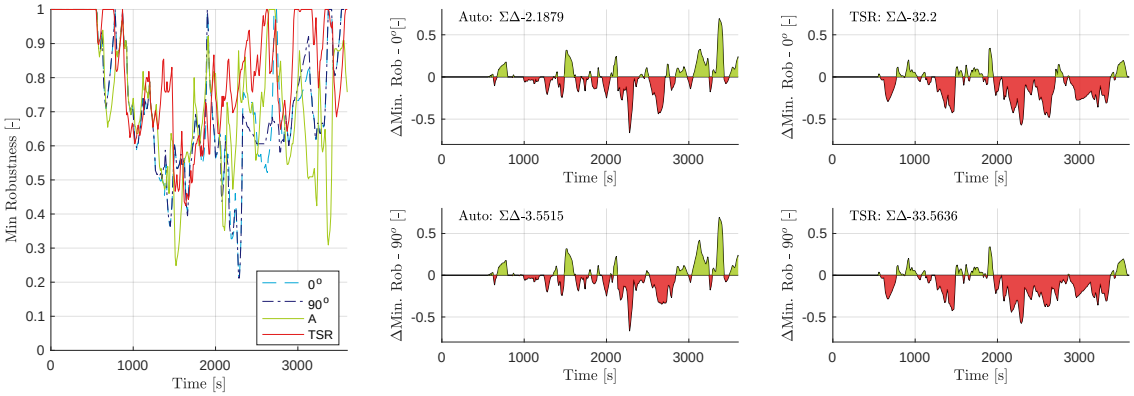


(c) Perturbation Location 2

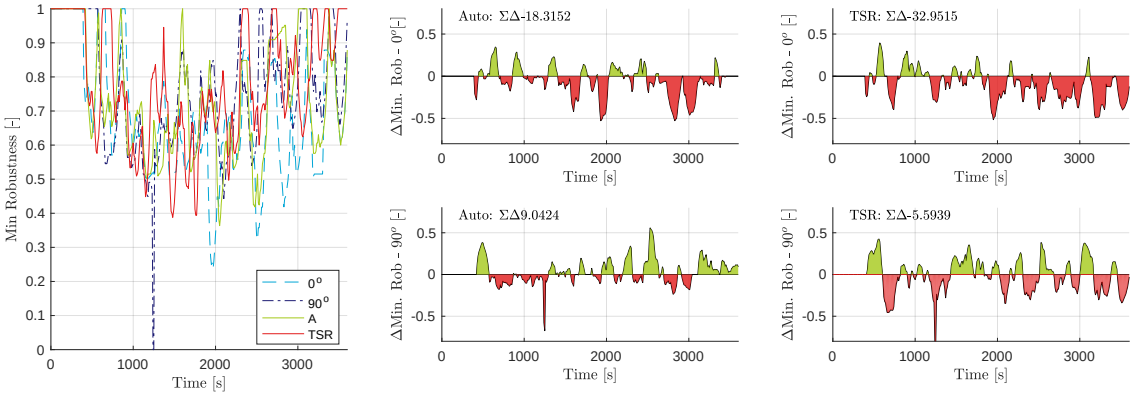
Figure D.53: Average sector robustness (no perturbation) low density traffic scenario participant 5



(a) Perturbation Location 0

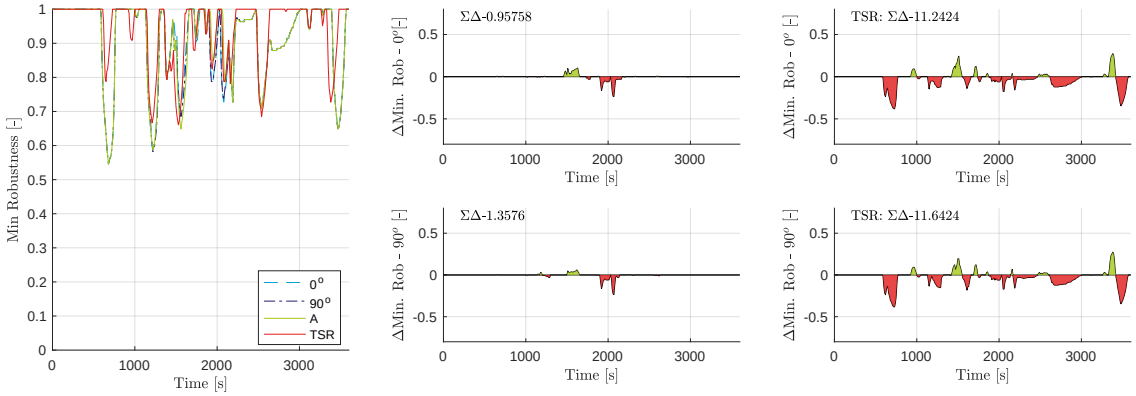


(b) Perturbation Location 1

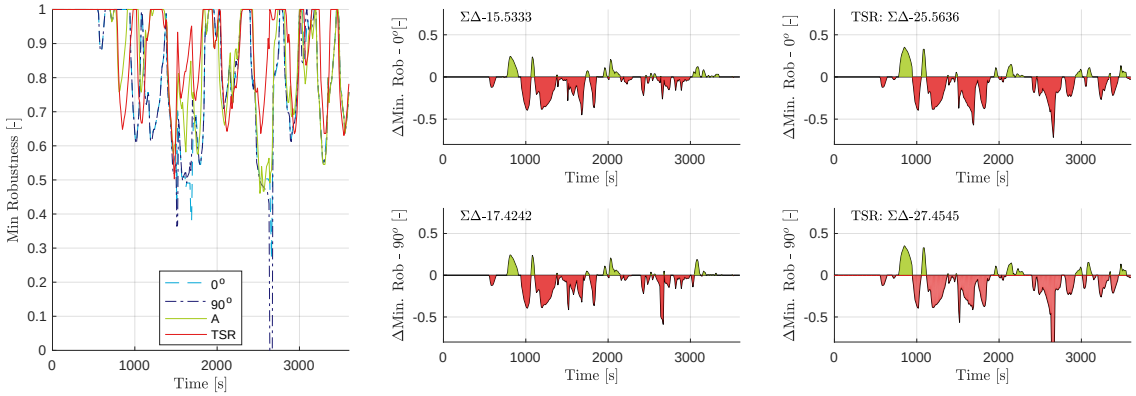


(c) Perturbation Location 2

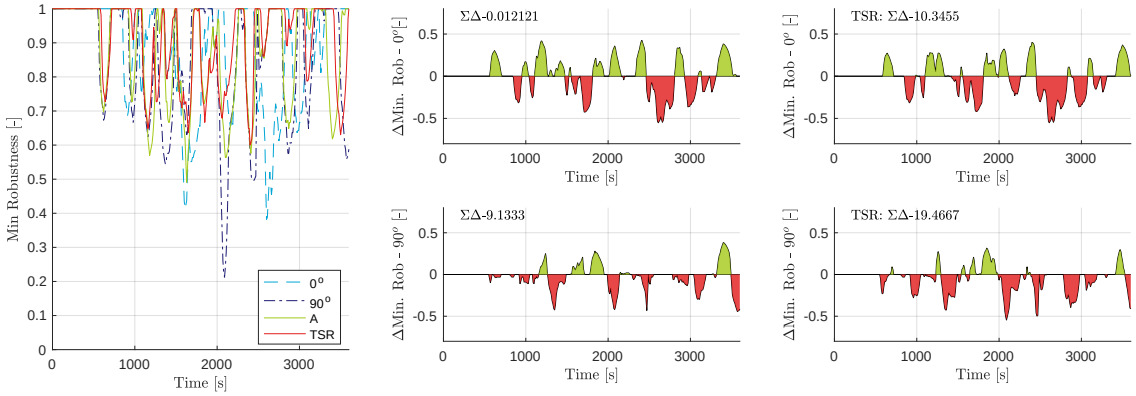
Figure D.54: Minimum sector robustness (no perturbation) high density traffic scenario participant 5



(a) Perturbation Location 0



(b) Perturbation Location 1



(c) Perturbation Location 2

Figure D.55: Minimum sector robustness (no perturbation) low density traffic scenario participant 5

Bibliography

[1] Vicente, K.J., Rasmussen, J.: Ecological interface design: Theoretical foundations. IEEE transactions on systems, man, and cybernetics. **22**(4) (1992) 589–606

[2] Westin, C., Hilburn, B., Borst, C.: Mismatches between Automation and Human Strategies: An Investigation into Future Air Traffic Management (ATM) Decision Aiding. Proceedings of the first SESAR Innovation days (December) (2011) 1–6

[3] Westin, C., Borst, C., Hilburn, B.: Strategic Conformance: Overcoming Acceptance Issues of Decision Aiding Automation? IEEE Transactions on Human-Machine Systems **46**(1) (2016) 41–52

[4] Rasmussen, J.: Skills, Rules, and Knowledge; Signals, Signs, and Symbols, and Other Distinctions in Human Performance Models. IEEE Transactions on Systems, Man, and Cybernetics **13**(3) (1983) 257–266

[5] Rasmussen, J., Pejtersen, A., Goodstein, L.: Cognitive Systems Engineering. Wiley (1994)

[6] Vicente, K.J.: Cognitive Work Analysis; Toward Safe, Productive, and Healthy Computer-Based Work. Lawrence Erlbaum Associates, Mahwah, New Jersey (1999)

[7] Rasmussen, J.: Information processing and human-machine interaction: an approach to cognitive engineering. Elsevier Science Inc., USA, New York (1986)

[8] Burns, C., Hajdukiewicz, J.: Ecological Interface Design. CRC Press, Boca Raton, FL, USA (2004)

[9] Van Paassen, M.M., Borst, C., Klomp, R., Mulder, M., Van Leeuwen, P., Mooij, M.: Designing for shared cognition in air traffic management. Journal of Aerospace Operations **2**(1-2) (2013) 39–51

[10] Klomp, R., Van Paassen, M.M., Borst, C., Mulder, M., Bos, T., Van Leeuwen, P., Mooij, M.: Joint human-automation cognition through a shared representation of 4D trajectory management. Proceedings of the SESAR Innovation Days (2012) 1–7

[11] Klomp, R., Borst, C., Mulder, M., Praetorius, G., Mooij, M., Nieuwenhuisen, D.: Experimental Evaluation of a Joint Cognitive System for 4D Trajectory Management. Proceedings of the Third SESAR Innovation Days (November) (2013) 1–8

[12] Klomp, R., Borst, C., Van Paassen, R., Mulder, M.: Expertise Level, Control Strategies, and Robustness in Future Air Traffic Control Decision Aiding. IEEE Transactions on Human-Machine Systems **46**(2) (2016) 255–266

[13] Pinto, J., Klomp, R., Borst, C., van Paassen, M.M., Mulder, M.: Design of an Ecological Flow-based Interface for 4D Trajectory Management in Air Traffic Control. 5th CEAS Air and Space Conference (141) (2015) 1–10

[14] Nagaraj, R., Borst, C., Paassen, M.M.V., Mulder, M.: Cooperative 4D-Trajectory Management for Future Air Traffic Control. PhD thesis, Delft University of Technology (2016)

[15] Billings, C.E.: Aviation Automation: The Search for A Human-Centered Approach. CRC Press (1997)

[16] Wickens, C.D., Mavor, A.S., Parasuraman, R., Mcgee, J.P.: The Future of Air Traffic Control: Human Operators and Automation. National Academy Press, Washington D.C. (1998)

[17] Parasuraman, R., Sheridan, T.B., Wickens, C.D.: A model for types and levels of human interaction with automation. IEEE transactions on systems, man, and cybernetics. Part A, Systems and humans : a publication of the IEEE Systems, Man, and Cybernetics Society **30**(3) (2000) 286–297

[18] Amelink, M.H.J., Mulder, M., van Paassen, M.M.: Designing for Human-Automation Interaction: Abstraction-Sophistication Analysis for UAV Control. Proceedings of the International Multi Conference of Engineers and Computer Scientiests 2008 (IMECS 2008) **I**(January 2008) (2008) 19–21

[19] Amelink, M.H.J.: Ecological Automation Design, Extending Work Domain Analysis. PhD thesis, Delft University of Technology (2010)

[20] Yang, L., Qi, J., Xiao, J., Yong, X.: A literature review of UAV 3D path planning. Proceedings of the World Congress on Intelligent Control and Automation (WCICA) (March) (2015) 2376–2381

[21] Ferguson, D., Likhachev, M., Stentz, A.: A guide to heuristic-based path planning. Proceedings of the International Workshop on Planning under Uncertainty for Autonomous Systems, International Conference on Automated Planning and Scheduling (ICAPS) (2005) 1 – 10

[22] Yang, L., Qi, J., Song, D., Xiao, J., Han, J., Xia, Y.: Survey of Robot 3D Path Planning Algorithms. Journal of Control Science and Engineering (2016)

[23] Karaman, S., Frazzoli, E.: Sampling-based Algorithms for Optimal Motion Planning. International Journal of Robotics Research **30**(7) (2011) 846–894

[24] Kavraki, L.E., Švestka, P., Latombe, J.C., Overmars, M.H.: Probabilistic roadmaps for path planning in high-dimensional configuration spaces (1996)

[25] Lavalle, S.M.: Rapidly-exploring random trees: A new tool for path planning. Technical report (1998)

[26] Dijkstra, E.W.: A note on two problems in connexion with graphs. Numerische Mathematik **1**(1) (1959) 269–271

[27] Hart, P.E., Nils, J.: Formal Basis for the Heuristic Determination of Minimum Cost Paths. IEEE Transactions of Systems Science and Cybernetics **4**(2) (1968) 100–107

[28] Nilsson, N.J.: Principles of Artificial Intelligence. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1980)

[29] Stentz, A., Mellon, I.C.: Optimal and efficient path planning for unknown and dynamic environments. International Journal of Robotics and Automation **10** (1993) 89–100

[30] Yue, R., Xiao, J., Joseph, S.L., Wang, S.: Modeling and path planning of the city-climber robot part II: 3D path planning using mixed integer linear programming. IEEE International Conference on Robotics and Biomimetics (2009) 2391–2396

[31] Anderson, S.J., Peters, S.C., Pilutti, T.E., Iagnemma, K.: An optimal-control-based framework for trajectory planning, threat assessment, and semi-autonomous control of passenger vehicles in hazard avoidance scenarios. Int. J. Vehicle Autonomous Systems **8**(2/3/4) (2010) 190–216

[32] Tisdale, J., Kim, Z.K.Z., Hedrick, J.: Autonomous UAV path planning and estimation. IEEE Robotics & Automation Magazine **16**(2) (2009) 35–42

[33] Mitchell, M.: An Introduction to Genetic Algorithms. MIT Press, Cambridge, MA, USA (1998)

[34] Dorigo, M., Di Caro, G., Gambardella, L.M.: Ant algorithms for discrete optimization. Artif. Life **5**(2) (April 1999) 137–172

[35] Glasius, R., Komoda, A., Gielen, S.C.: Neural Network Dynamics for Path Planning and Obstacle Avoidance. Neural Networks **8**(1) (jan 1995) 125–133

[36] Richards, A.: Constraining the Sense of Conflict Resolution: Supervision of Route Optimization. First SESAR Innovation Days (December) (2011) 1–7

[37] Turnbull, O., Richards, A.: Examples of Supervisory Interaction with Route Optimizers. Second SESAR Innovation Days (November) (2012) 1–8

[38] Idris, H.R., El-wakil, T., Wing, D.J.: Trajectory Planning by Preserving Flexibility: Metrics and Analysis. Proceedings of the AIAA Guidance Navigation and Control (GNC) Conference (August) (2008) 1–14

[39] Idris, H., Shen, N., El-Wakil, T., Wing, D.: Analysis of Trajectory Flexibility Preservation Impact on Traffic Complexity. AIAA Guidance, Navigation, and Control Conference (August) (2009) 1–14

[40] Idris, H.R., Shen, N., Wing, D.J.: Improving Separation Assurance Stability through Trajectory Flexibility Preservation. Aviation (September) (2010) 1–11