COMPARISON OF MODEL CHECKING AND SIMULATION TO EXAMINE AIRCRAFT SYSTEM BEHAVIOR

A Thesis
Presented to
The Academic Faculty

by

Gabriel E. Gelman

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in the
School of Aerospace Engineering

Georgia Institute of Technology
August 2013

TABLE OF CONTENTS

LIST OF TABLES . . . . . **v**

LIST OF FIGURES . . . . . **vi**

GLOSSARY . . . . . **viii**

SUMMARY . . . . . **viii**

I INTRODUCTION **1**

II BACKGROUND **4**

2.1 Mental Models 6

2.2 Model Checking 9

2.3 Simulation 12

III AUTOMATION SURPRISES **15**

3.1 Case Study Description: Airbus Automatic Speed Protection 15

3.1.1 Flight 381 incident involving Automation Surprise 18

3.1.2 Model Checking the Airbus Automatic Speed Protection . . . 19

3.2 Definition of Automation Surprise 20

3.3 Detecting and Modeling Automation Surprises in SAL 22

3.4 Automation Surprises in WMC 24

3.5 Implementing Automation Surprise in WMC 25

3.6 Comparison of WMC to SAL 28

IV METHOD **30**

4.1 Method to translate between Frameworks 31

4.2 Make SAL scenario appear in WMC 35

4.3 Changes in WMC 36

4.3.1 Aircraft Performance Model 36

4.3.2 Workmodels 38

4.4 Experimental Design 38

4.4.1 Variables for Scenarios . . . 39
4.4.2 Go Around / Missed Approach altitude . . . 39
4.4.3 When is Go Around altitude set . . . 41
4.4.4 Time at which Flaps are extended . . . 42
4.4.5 Altitude at which Level Off is commanded . . . 43
4.4.6 Duration of the level off . . . 45
4.4.7 Inputs from SAL . . . 45
4.4.8 Total number of scenarios . . . 48

V RESULTS . . . 49

5.1 SAL matching case . . . 49
5.2 Overspeed Causes . . . 51
5.3 Simulation Artifacts . . . 52
5.4 Grouping Results into meaningful Scenarios . . . 52
5.4.1 Scenario 1: OPEN DES, No Automation Surprise, Flaps Extension before Level Off causes Mode Reversion . . . 55
5.4.2 Scenario 2: OPEN CLB, Flaps Extension before Level Off causes Mode Reversion . . . 55
5.4.3 Scenario 3: OPEN DES, After Level Off, Extension of Flaps causes Mode Reversion, Guidance takes over after reaching MCP altitude . . . 57
5.4.4 Scenario 4: OPEN CLB, After Level Off, Extension of Flaps causes Mode Reversion . . . 63
5.4.5 Scenario 5: OPEN DES, Dive causes Overspeed and subsequent Mode Reversion . . . 63
5.4.6 Scenario 6: OPEN CLB, Dive causes Overspeed and subsequent Mode Reversion . . . 65
5.5 Analysis . . . 68
5.5.1 SAL Equivalent Scenario . . . 68
5.5.2 Next Iteration: Modelcheck WMC Scenarios . . . 69

VI CONCLUSION . . . 72

6.1 Summary . . . 72

6.2 Contributions 73
6.3 Future Work 74
APPENDIX A — ANCILLARY 76
REFERENCES 84

LIST OF TABLES

1 Flight modes relevant for automatic speed protection. 17

2 Comparison between WMC and SAL. 28

3 Flight mode logic description. 37

4 Variables used to define the simulation scenarios. 39

5 Inputs from SAL for WMC 47

6 Variable values for scenarios 48

7 Overview of the scenarios that appeared 53

LIST OF FIGURES

1 This figure shows the way automation surprises occur. 7

2 Example of a fault tolerance algorithm model. 9

3 Example of the exploration of the state space. 10

4 Architecture of a model checker. 11

5 Sequence of the automation surprise. 18

6 Reconstitution of the aircraft's path [2]. 19

7 Sequence of events leading to the automation surprise in SAL [1]. . . 20

8 Iterative method starting from an initial scenario for SAL. 33

9 The two methods, SAL and WMC, and how they are connected in the method. 33

10 Mode transition logic when automatic speed protection is engaged. . 37

11 Determining the lower and upper limit for the go around altitude. . . 41

12 Sequence of events leading to the automation surprise in SAL [1]. . . 47

13 Altitude profile WMC with SAL preconditions. 50

14 Altitude profile of SAL model. 50

15 Shows the states and state transitions in the SAL model. 51

16 Altitude profile of a nominal approach. 54

17 Altitude profile of the Tarom 381 approach. 54

18 Exemplary altitude profile of the aircraft in Scenario 1. 56

19 Closeup of the altitude profile of Scenario 1. 56

20 Conditions that make Scenario 1, Scenario 3 and Scenario 6 appear. . 57

21 Exemplary altitude profile of the aircraft in Scenario 2. 58

22 Conditions that make Scenario 2 and Scenario 3 appear. 59

23 Conditions that make Scenario 2 and Scenario 3 appear. 60

24 Exemplary altitude profile of the aircraft in Scenario 3. 61

25 Closeup of the altitude profile of Scenario 3. 61

26 Conditions that make Scenario 3, Scenario 4, Scenario 5 and Scenario 6 appear. 62

27 Exemplary altitude profile of the aircraft in Scenario 4. 64

28 Closeup of the altitude profile of Scenario 4. 64

29 Exemplary altitude profile of the aircraft in Scenario 5. 66

30 Closeup of the altitude profile of Scenario 5. 66

31 Exemplary altitude profile of the aircraft in Scenario 6. 67

32 Closeup of the altitude profile of Scenario 6. 67

33 Scenarios that appeared under the preconditions given in SAL. 69

34 Altitude profile of the SAL scenario shifted to a higher altitude and the actions and events in the scenario. 70

35 Iterative method to connect WMC and SAL. 70

36 Iterative method to combine SAL and WMC. 75

SUMMARY

Automation surprises are examples of poor Human-Machine Interaction (HMI) where pilots were surprised by actions of the automation, which lead to dangerous situations during which pilots had to counteract the autopilot. To be able to identify problems that may arise between pilots and automation before implementation, methods are needed that can uncover potentially dangerous HMI early in the design process.

In this work, two such methods, simulation and model checking, have been combined and compared to leverage the benefits of both. In the past, model checking has been successful at uncovering known automation surprises. Simulation, on the other, hand has been successful in the aviation domain and human factor issues.

To be able to compare these two approaches, this work focused on a common case study involving a known automation surprise. The automation surprise that was examined, is linked to the former Airbus speed protection logic that caused aircraft on approach to change the flight mode, resulting in a sudden climb. The results provided by the model checking with SAL (Symbolic Analysis Laboratory) in a previous work, have been used to provide input for simulation. In this work, this automation surprise was simulated with the simulation platform WMC (Work Models that Compute) and compared to the corresponding results from SAL.

By using the case study, this work provides a method to examine system behavior, such as automation surprises, using model checking and simulation in conjunction to leverage the benefits of both.

CHAPTER I

INTRODUCTION

The increasing complexity of cockpit automation demands new non-traditional analysis approaches and new procedures to guarantee its smooth collaboration with pilots. One implication of more sophisticated automation is that the role of pilots has shifted from active control to a supervisory role. This shift is justified to some extent by the claim that the human is the largest source of variability in the system. However, under off-nominal conditions the human's problem solving skills and inherent variability are seen as an asset, not a nuisance. Thus, human pilots remain the responsible agents, which makes it necessary for the pilots to be aware of the situation in the cockpit. This responsibility is challenged when the automation behaves unexpectedly and the pilots lose track of the situation, i.e. if the pilots do not anticipate or comprehend the aircraft maneuvers commanded by the automation. The resulting effect is called "automation surprise" from the pilots perspective. An example is when the automation changes the flight mode of the aircraft and the pilot is neither aware of the change, nor able to comprehend the behavior resulting from the new flight mode. Such an automation surprise is called "mode confusion" because the pilot does not understand what mode the autoflight system is currently in.

To investigate system behavior and predict potential problems, such as automation surprises new, more powerful analysis methods are needed. Further, such issues between pilots and automation need to be detected early in the design process.

Model checking and simulation are two promising methods, both having distinct advantages and limitations. Simulation, in the context of this work, is digital, continuous time, agent-based and including a high-fidelity aircraft model. Model checking

originates from computer science, where properties of software or hardware systems were checked. In model checking a simple model of a system and transition paths between states (also called actions) are fed into a model checker and then evaluated exhaustively as to whether it meets a given specification. For example, a model could describe an internet firewall and the objective could be to prove its robustness to a single failure mode. Due to its origins, model checking has had success with problems related to computer science where Boolean outcomes for operations are expected. Using model checking, a discrete safety requirement can either be proved to be true or false, which by implication indicates whether the system is considered safe or unsafe. In the domain of aviation, one challenge in proving that safe properties hold is that human behavior is rarely limited to binary options. Further, describing human-automation interaction demands sophisticated human performance models that are more complex than model checking algorithms can currently handle. Other challenges are that model checking lacks explicit representation of time and continuously evolving dynamics. Therefore, it is difficult to apply the results to real world aircraft.

Hybrid time simulation frameworks such as WMC (Work Models that Compute) have historically been used successfully in the aviation domain for evaluation of human-automation interaction. Simulations can provide more information about the events leading to problems in human-automation interaction as they utilize sophisticated and realistic models of humans, automated agents and their environment. However, simulations, due to their inherent complexity, involving aircraft dynamics and sophisticated models of agent behavior, cannot exhaustively explore the full range of outcomes for pragmatic reasons. The complexity and continuity given in such simulations provides an infinite amount of state-space to explore. Instead simulation analysis must choose a small subset of conditions to vary. Such conditions can describe varied agent behavior, as it can describe varied machine behavior.

The objective of this work is to evaluate a common scenario using both methods,

compare and contrast outputs and inputs, and examine assumptions and requirements for capturing automation surprises. Furthermore, this work will briefly discuss how these two approaches can be combined for future examination of potential automation surprises linked to more automated cockpits. More generally, such approaches can be used to study future autonomy and authority assignments between pilots and automation, the risks such new assignments bear, and ultimately how to mitigate these risks. Finally, being able to simulate a known automation surprise helps validate the capability to simulate potential automation surprises of future aircraft systems.

The automation surprise that is used as case study in this work is an automatic mode change linked to the Airbus automatic speed protection that led to numerous automation surprise incidents in the 1990's, and was subsequently changed. This thesis will create the simulations necessary to predict this automation surprise and to compare and contrast the insights provided here with those of Rushby and colleagues using model checking as described by Crow et al. [8].

CHAPTER II

BACKGROUND

Over 30 years ago, Wiener and Curry [31] examined flight-deck automation and were questioning whether functions previously performed manually should be automated due to human factors issues. Even today this issue is current with the further progression of automation in the cockpit and the increasing number of tasks performed by the automation. In accordance to the concerns about pilots' problems understanding the automation expressed by the human factors community, Wiener [30] presented the result of a three-year study on the impact of new automation in a Boeing 757, at that time considered highly automated. The study revealed a flight crew that was surprised by the plane's automated system. Sarter and Woods [24, 25] also evaluated the pilot-automation interaction in the cockpit and examined pilots' difficulties understanding the Flight Management System (FMS), especially problems with tracking its status. Subsequently, Sarter and colleagues [27] summarized their extensive findings on pilot interaction with modern flight deck automation in a paper entitled "Automation Surprise." They concluded that automation surprises are not simply the result of over-automation, but is one result of design failures driven by the assumption that new technology necessarily improves system performance.

The lack of feedback from the automation system is one of the problems that was identified as leading to automation surprises. Sarter and colleagues [26] examined the Airbus A320, a highly automated aircraft equipped with a glass cockpit, and its role in automation surprises. Their analysis showed that most pilots experienced automation surprises independent of their previous glass cockpit experience and are a prevelant issue when it comes to the progression in cockpit automation. The authors classified

automation surprises into two categories: 1) the automation fails to do something the pilots expect, and 2) the automation does something the pilots did not expect. The pilots especially raised concern about the lack of feedback from flight controls in the new fly-by-wire A320, including the nonmoving throttles. The authors came to the conclusion that feedback systems are important for pilots to stay aware of what is going on in the cockpit, which remarks a major challenge for the development of highly automated cockpits. Palmer [16] came to the same conclusion in two case studies of Human-In-The-Loop (HITL) experiments. In the experiments the pilots were exposed to high workload which led to automation surprises. He concluded that pilots should monitor the Flight Mode Annunciator displays more vigilantly, since these were often the only indication of what the automation was doing. Furthermore, he suggested that autoflight designers should provide more salient displays. Finally, Leveson [14] analyzed flight deck automation and proposes ways to mitigate "Technology-Induced Human Error" by making changes to the system while still in the conceptual phase.

Automation surprises are based on the premise that the pilots' *mental model* of the aircraft automation differs from the actual way the aircraft automation works. Norman [15] has done fundamental work in the human-computer interaction domain of mental models, which is of interest in this work describing the interaction between pilots and aircraft (automation and autopilot). Through his observations he came to the conclusions that mental models are incomplete, unstable, unscientific and parsimonious. Doyle and Ford [10] have reviewed the literature for mental models, especially for the domain of mental models in dynamical systems, such as an aircraft. Their description of a mental model states that "*a mental model of a dynamic system is a relatively enduring and accessible, but limited, internal conceptual representation of an external system whose structure is analogous to the perceived structure of that system*". Doyle et al. [9] and Romera [19] investigated how to measure these mental models.

2.1 Mental Models

There are two types of mental models. One describes the mental model of the system dynamics and can be characterized as the "long term mental model". This mental model reflects the pilots' understanding of the airplane as a whole, including the cockpit system. It is the long term mental model that allows a pilot to know given the current state of the aircraft. Formation of this mental model about a particular aircraft starts the first time the pilot learns about the airplane in training. The information about a system contained in this mental model changes slowly and is stable (when, for instance, the pilot finds new or forgets functionalities of the aircraft). The mental model can be changed purposefully through training or implicitly through experience. The second type of mental model describes the current state of the environment, and is characterized as the "belief" of the pilot, and is of central interest in this work. It comprises all variables describing the current state of the aircraft, including speed, altitude and heading. Whenever the pilot takes a look at the flight instruments the belief is updated. Another part of this type of mental model is the expectation of the pilot about the future state of the aircraft. If the belief and the expectation do not match an automation surprise occurs as shown in Figure 1.

Javaux [11, 12] did work on the first type of mental model that describe the system dynamics. He used state models to explain Sarter and Woods' work on automation surprises, placing special emphasis on mode confusion. He came to the conclusion that mode confusion appears because pilots are not aware of all of the preconditions that have to be fulfilled for a given mode change, and that the reason humans are not fully aware is caused by (1) frequential and (2) interferential simplification. Javaux's work explained how pilots simplify or reduce the number of preconditions that trigger a certain mode change to the preconditions that are most salient. Preconditions for intended mode changes that are not salient, but are almost always satisfied, are forgotten. Circumstances in which the forgotten precondition is not satisfied, often

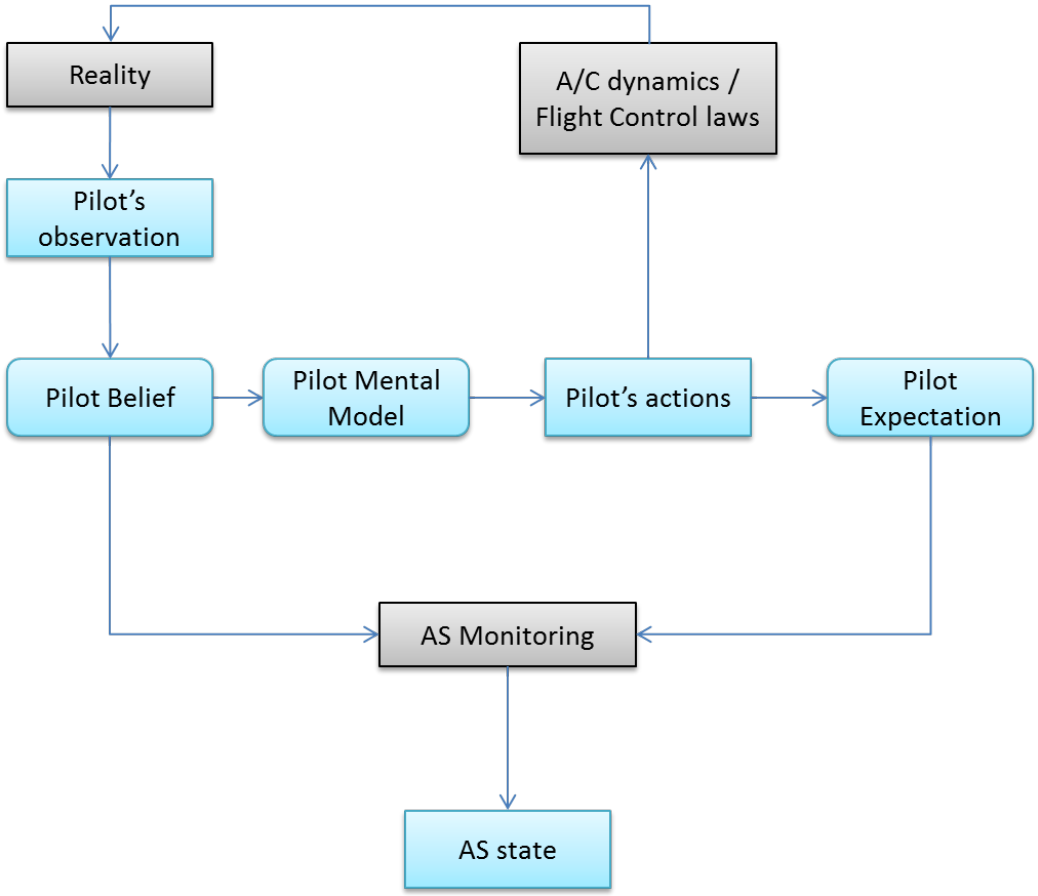

Figure 1: This figure shows the way automation surprises occur.

leads to mode confusion.

Work on *"short-term " mental models*, in this case the beliefs, about the system state in human agents has been done extensively by Corker and colleagues [7] in their human performance model MIDAS (Man-Machine Integration Design and Analysis System). They simulated the pilot's belief through a construct called Updateable World Representation (UWR). The UWR is a subset of the state of the world, which is instantiated with the current state of the world. Subsequently, when the simulation runs, the UWR may begin to deviate depending on the percepts the human agent is capturing and the state of the memory. The UWR accounts for the fact that humans have an internal representation of the external world that usually differs from the actual state of the world. Their belief of the state of the aircraft may not correspond

to the real state, which takes part in explaining automation surprises in the cockpit. Depending on how inaccurate the data in the mental model is that the agent acts upon, this may lead to undesired results. For instance, if the pilot acts on the airplane thinking one particular flight mode is engaged, while the flight mode has been changed by the automation, this may lead to aircraft maneuvers that were not desired by the pilot. Furthermore, memory does not store everything that is perceived for an infinite amount of time. Memory decays in the UWR, which is modeled by an exponential function that leads to forgetting of information stored in the UWR over time. The MIDAS UWR does not capture how the mental model would evolve over time to simplify its explanation of the way the system operates, as Javaux did. Instead it only relies on the state of the world and the update of the copy of this state in the UWR depending on the percepts captured.

Corker and colleagues' [7] human performance model MIDAS was integrated in an accident risk assessment methodology called TOPAZ [3, 28] (Traffic Organization and Perturbation Analyzer) by Blom and Corker and colleagues [4]. Their work demonstrated the feasibility of such an integration. In their paper, they describe how to do this integration to describe the risk of collision between two aircraft.

In 1998, Butler et al. recognized [5] the power and benefit of *model checking* in assisting human factors through "a deeper scrutiny of the internals of the automation" and at describing and avoiding mode confusion. Rushby and colleagues continued on this path and described system behavior and mental models as finite state transition systems to make them viable for model checking. In particular they examined automation surprises involving the Airbus A320 [8], Boeing 737 [20] and MD-88 [21]. Rushby's [1] instantiation of a mental model for use in his model checker SAL (Symbolic Analysis Laboratory) is simpler than the UWR. A mental model for inclusion in SAL only represents the pilot's expectation about the state of the aircraft: descending, leveled or climbing. This expectation is updated upon the pilots performing actions

that lead to the change of this state. Rushby's work on the automation surprise in [8] is described in detail in section 3.1.2.

2.2 Model Checking

Model Checking originates from computer science and is still used particularly in this domain for property checking of system models. A model is checked for a specified property by an exhaustive state space exploration (see Figure 3).

A classical example for model checking is a fault tolerance algorithm examining the error robustness of a signal transmission system. A source sends out a signal to one or several receivers that is passed through relays. The specification to be checked is how many faulty relays the system can handle before a wrong signal is passed to the receivers. Figure 2 shows such a system comprising a source, three relays and two receivers. One specification to check in this system can be: is the correct signal going to reach the receivers if one relay is faulty. Obviously, when three relays are faulty there is no possibility for the correct signal to reach the receivers.

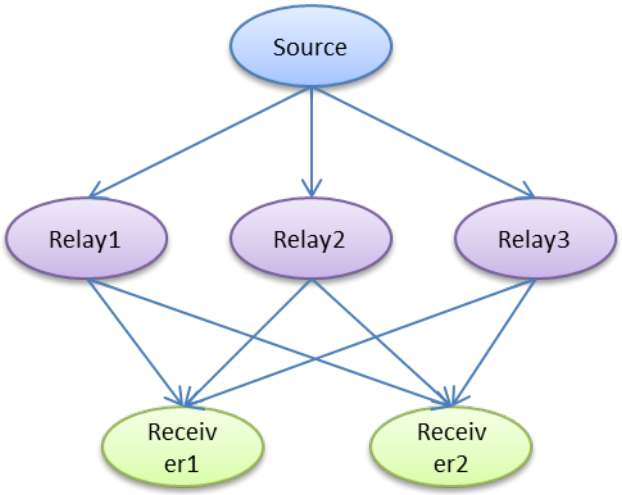

Figure 2: Example of a fault tolerance algorithm model.

A model comprises *an initial state*, *state transitions* and a *specification* that is checked. Rushby describes in [22] and [23] how to create a model with SAL (Symbolic

Analysis Laboratory), the model checker described in this work. All variables of the model are set to an initial value which describes the initial state from which the model checker begins the state exploration. *State transitions* describe changes to variables that occur in between states. Transitions can require a precondition to be true to occur or may occur without fulfilling preconditions. In state transitions, one or more variables are set to a new value. Last, the *specification* to be checked is defined, which is characterized by a state in which the specification no longer holds. For instance, a specification can be that a variable can never take a certain value, no matter which state transitions occur.

Model checkers take the initial system state and exhaustively search the state space associated with the prescribed transitions looking for violations in the specification. If the specification is fulfilled, then the model checker notifies the user that the undesired state (in Figure 3 the state with the 'State NOT OK' sign) is not reachable. If the specification is not fulfilled the model checker demonstrates how the undesired state can be reached by showing the state transitions that lead from the initial state to the undesired state (see Figure 4).

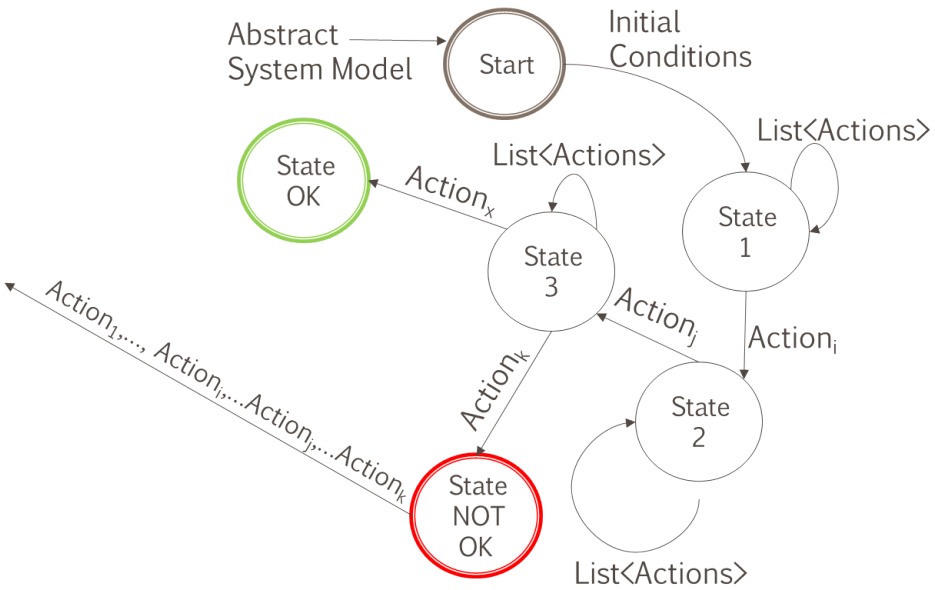

Figure 3: Example of the exploration of the state space.

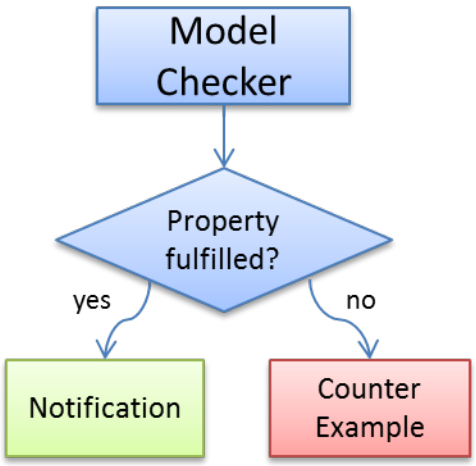

Figure 4: Architecture of a model checker.

In model checking time is not explicitly modeled but implicitly represented by state transitions. A state is characterized by properties and the transitions are characterized by the change of these properties. A state transition that leads the model from one state to the next can change one or several state variables at the same time. The change of state variables can be either very large or small depending on how the state transition is characterized. The time such a state transition represents in reality can therefore be large or small, depending on how long such a state transition would take in reality. For example if a state transition can change the altitude of the aircraft by 500 feet, then such a transition would take several seconds in reality, depending on the pitch, altitude and speed of the aircraft.

Due to the exhaustive nature of model checking, depending on how complex the model is, the number of potential states may be prohibitive. To reduce the number of states that need to be explored, it is reduced using algorithms. For instance, SAL (Symbolic Analysis Laboratory) uses a symbolic algorithm to reduce the number of states. Nevertheless, a poor definition of the model can lead to a state space explosion that a model checker cannot examine in an appropriate time frame anymore. For instance, a state explosion occurs when representing continuous variables as each

having an infinitely large number of discrete values. Therefore, all variables have to be discrete or discretized into a bounded set of values. Unfortunately, such an architecture makes it impossible to fully model aircraft dynamics.

CounterExample-Guided Abstraction Refinement (CEGAR) [6] is a method in model checking in which a simple model is constructed, which then is iteratively refined. The model is checked for the desired specification; if a counterexample is found that does not appear realistic, the model is refined implementing new state transitions or refining unrealistic state transitions. This process is continued until a counterexample is found which appears feasible.

2.3 Simulation

Simulations aim to reproduce real world behavior at a useful level of fidelity. Flight simulations have to comprise flight dynamics of the aircraft to be simulated and imitate a real world flight leading to a high level of fidelity. Ideally, behavior observed in simulations can be reproduced in reality and vice versa. The benefit of simulations is that they are less cost intensive than performing hardware or human-in-the-loop (HITL) tests and allow a broader spectrum of tests than real world agents or systems would allow.

In the context of this work, the degree of fidelity models continuous aircraft flight dynamics in contrast to highly abstracted flight dynamics given in model checking, namely in the SAL model provided by Rushby and colleagues [1]. Furthermore, simulations, such as in this work, explicitly model time. Every aircraft state and action is linked to a specific time at which it occurs. Model checkers do not provide time explicitly, instead rely only on the sequence of actions, which also lacks the description of the full aircraft state. In the simulation model given in this work, the actions of the involved agents: pilot, air traffic controllers and automation are decomposed into the manipulation of levers and buttons. For instance actions for the

pilots include deploying flaps, dial and push heading selector. Decomposing further into human agent body movements is not necessary, nor helpful.

WMC, the *simulation* platform that was used for the simulations in this work, was developed to predict human-automation interaction in operational contexts [17, 18]. WMC comprises agent models that are generic and autonomous from any description of work. The knowledge of work is encapsulated outside the agent models. Work particular to a system, such as the cockpit system including the pilots and the automation in the cockpit or the controllers working in Air Traffic Control are such a system. These models describing the work of a particular sub-system situated in the simulation are called workmodels. Each workmodel is built hierarchically similar to an Abstraction Hierarchy (AH) [29] at which bottom the functions are decomposed of actions that are passed to the agent models that execute the actions.

In WMC, there are different types of agents. Perfect agents that execute each action exactly as is without having any internal representation of a task structure. This kind of agent is used for any machine, such as the flight deck automation. The other type is a human agent model that has a limit of tasks it can execute at a time (usually 3, 7 or 50, which corresponds to an unlimited amount). If the human agent is completely occupied, meaning executing the maximum number of actions, new actions with same or lower priority are delayed and new actions with higher priority interrupt active actions. To keep track of these tasks the human agent model has a list for active, delayed and interrupted actions. If an action is delayed or interrupted for too long, the pilot may forget them altogether.

Each action that is part of a workmodel contains information about what resource variables it manipulates (gets information from or sets to a new value) and how frequently it has to be executed. Resources describe the state of the environment and the aircraft and are the only possibility for an agent to perform changes in the environment within an action. Each action has to specify which resources it is allowed

to retrieve ("get") and which it is allowed to change ("set").

Actions are executed in the order they are specified by the agent they are assigned to. A simulation clock keeps track of which action is scheduled next and after an action is executed it is either scheduled for another time in the future or to the a very long time later, which is a time that is practically never reached by the simulation. When an action is set for a very long time away and another action needs to schedule this action, its next scheduled time is updated to a closer time. For more information regarding WMC, see [17] and [18].

CHAPTER III

AUTOMATION SURPRISES

3.1 Case Study Description: Airbus Automatic Speed Protection

This work uses an automation surprise found in previous versions of Airbus models A300, A310 and A320 as a case study. An unexpected automatic flight mode change during approach is initiated by the automatic speed protection, resulting in a mode change from vertical speed / flight path angle (V/S FPA) mode in which the aircraft maintains either vertical speed or flight path angle, to an open mode which gives priority to the airspeed. Two open modes are available, open climb (OPEN CLB) and open descend (OPEN DES). They ignore any FPA or VS previously commanded and rely on the Flight Control Unit (FCU) target altitude (FCU for Airbus and MCP, Mode Control Panel, for Boeing). If the FCU target altitude is set to an altitude above the current altitude, the automation switches into the OPEN CLB flight mode and starts climbing regardless of flight phase. In descent this mode change, called "mode reversion," can be an automation surprise to the pilots who are attempting to land.

Two NASA Aviation Safety Reporting System (ASRS) reports describe automation surprises that likely involved the automatic speed protection. In the first report from 1994 (ACN: 268726) the aircraft, an Airbus A320, was on approach to Minneapolis St. Paul International Airport. The aircraft was instructed to level off at 4000ft by ATC (Air Traffic Control). After resuming approach and descending a few hundred feet mode reversion caused the aircraft to climb automatically to the Missed Approach altitude that has been entered earlier into the FCU. In the second report, also in 1994 (ACN: 262473), an Airbus A310 was at 2000ft on approach when the

aircraft initiated a climb with Go Around thrust. The aircraft climbed up to 4000ft even though the Missed Approach altitude was set to 3000ft.

To model this automation logic, three flight modes are of interest. The first is the V/S FPA mode that holds either a constant vertical speed or a constant flight path angle commanded by the pilot. The vertical speed can be set between ±6000ft per minute, and the flight path angle can be set between ±9.9°. If the pilot commands an FPA of -3°, the aircraft will keep this angle on descent, with the engines tracking speed. Most automation surprises occur on the ILS (Instrument Landing System) glideslope (G/S) when the aircraft descends with an FPA of 3°. If the aircraft descends at such a low FPA, the airspeed stays below the maximum airspeed. However, even with the throttles at idle the airspeed can approach a dangerous value when the aircraft descends with higher FPAs.

One such scenario where the aircraft is forced to descend with a higher FPA is when the pilots are told by ATC to level off while on the ILS glideslope, as was the case in the 1994 Minneapolis incident. ATC may issue such an instruction if the airspace is congested or the runway is still occupied. When the aircraft is then allowed to descend again, it is necessary to dive down to recapture the ILS G/S. The longer the aircraft is commanded to hold its altitude, the steeper the FPA required to regain the G/S. The steeper the descent path, the more the aircraft accelerates. As soon as the aircraft reaches the maximum airspeed, the automation switches from the V/S FPA flight mode to one of the open modes to protect the airframe by reducing speed. The open mode depends on the altitude entered into the FCU: if this value is above the current altitude, the OPEN CLB mode is engaged; if below, the OPEN DES mode is engaged. The mode change to OPEN CLB is the change of interest of this work, since it is the mode change that causes the automation surprise during landing. The pilots expect the aircraft to descend since they are on final approach but the aircraft interrupts the descent and initiates a climb often with the aircraft in

a "dirty" configuration with flaps and possibly gear extended.

The time when the FCU altitude is set to the Missed Approach altitude and the time when the flaps are extended are both essential to whether the automation surprise occurs. Usually, the FCU altitude is set to the Missed Approach altitude when the aircraft is on final approach. As described, the automation surprise may only take place when the aircraft descends below this altitude. Another factor that makes the incident more likely is if flaps are extended prematurely. Deployed flaps decrease the maximum airspeed, which increases the risk of reaching it.

Table 1: Flight modes relevant for automatic speed protection.

Flight Mode	**Characteristic**	**Airspeed**
V/S FPA	VS = ± 6000ft/min or FPA = ± 9.9° (idle)	No consideration to airspeed
OPEN CLB	Full thrust, climb to FCU altitude	Below max airspeed
OPEN DES	Idle thrust, adjust attitude, descend to FCU altitude	Below max airspeed

In current Airbus aircraft, the problem of the automatic mode change leading to the automation surprise has been fixed and, as soon as the maximum velocity is reached, the flight path angle is adjusted to a less steep angle while remaining in the V/S FPA flight mode. However, given the good intentions of automation designers when creating this automatic mode change, creating tools that can detect issues such as these will help prevent similar issues with automation in the future.

Figure 5 illustrates events leading to the automation surprise incident described in the 1994 Minneapolis incident. In Step 1, the aircraft is on the ILS glideslope, and the FPA V/S flight mode is engaged with an FPA of -3°. In Step 2, ATC tells the pilots of the aircraft to level off and maintain a certain altitude until further notice. Subsequently, ATC allows the aircraft to resume descent and in Step 3, the aircraft tries to recapture the ILS glideslope with a higher FPA. The steeper approach in Step 3 results in a higher airspeed until the maximum airspeed is exceeded. In Step 4 the

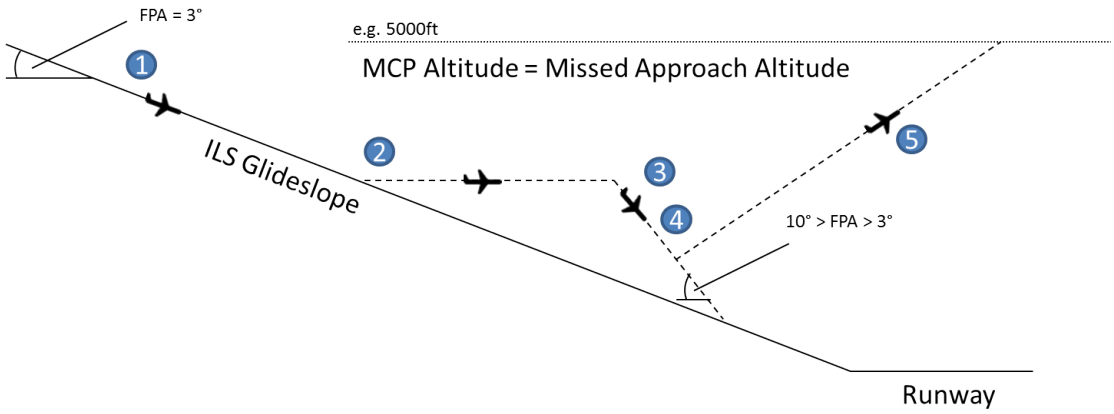

Figure 5: Sequence of the automation surprise.

flight mode changes to OPEN CLB because maximum airspeed is reached and FCU altitude is above the current altitude, which are the preconditions for the flight mode change.

3.1.1 Flight 381 incident involving Automation Surprise

The most notable incident involving this automation surprise was Tarom flight 381 involving an Airbus A310 from Bucharest to Paris Orly in 1994. The aircraft did not capture the ILS glideslope automatically and the captain disconnected the autopilot to capture the glideslope manually. However, he did not disconnect the auto-throttle. Additionally, prior to disengaging the autopilot the captain had set the Go Around altitude. When the captain extended the flaps at 1700ft to the next configuration, the airspeed was the same as the maximum airspeed with flaps extended at this configuration, i.e. 195kt. Subsequently, the automatic speed protection engaged and attempted to slow the aircraft down by changing the mode to an open mode. Since the MCP altitude was 2000ft, the aircraft changed to OPEN CLB and started climbing to this altitude, and the engine thrust increased automatically. The resulting nose up effect was countered by the crew with the pitch controls. The trimmable horizontal stabilizer (THS) at its maximum nose-up value could not be sufficiently countered by the crew with the elevators. The aircraft started to climb with a pitch attitude

of 60°. Shortly hereafter the aircraft stalled. Luckily, the pilots managed to recover without any damage to the structure of the aircraft or any injured occupants.

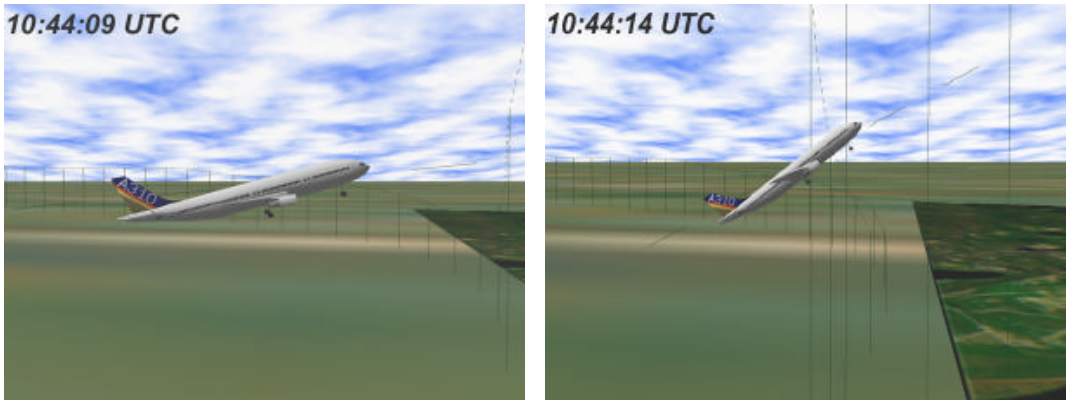

Figure 6: Reconstitution of the aircraft's path [2].

3.1.2 Model Checking the Airbus Automatic Speed Protection

The automation surprise resulting from the automatic mode change described above has been formally proved by Rushby and colleagues [1] using the model checker SAL. Rushby has formally shown that, with automatic speed protection logic active, there exists at least one scenario where the pilots' mental model of the aircraft state is at odds with the FMS control logic, resulting in an automation surprise. In Rushby's formal specification the interaction of pilots, aircraft and automation is modeled by discrete states. The entire state-space of a simplified model of an aircraft approach was exhaustively explored to uncover action sequences that would result in an automation surprise. Due to the limitations of this formal specification, actions were modeled sparsely so that only a few actions were included in the specification. Figure 12 shows the sequence of states and actions that SAL analysis revealed probably led to the state transitions resulting in the automation surprise. The commands signify the actions performed by the agents: aircraft, automation, and pilot. The pilots' mental model (expectation about future state of the aircraft) is in descend even in step 6 where the aircraft initiates a climb. In Figure 12 this can be seen by the

altitude changing in step 5 from 2990ft to 3291ft in Step 6. The altitude to which the aircraft climbs, 3291ft, is the FCU altitude. This indicates the moment where the aircraft state and the mental model of the pilots start to deviate, which marks the automation surprise event. The output shows what actions have to take place so that the automatic speed protection engages leading the aircraft to climb, resulting in the automation surprise.

Step	Flight Mode	Airspeed	Altitude	FCU Altitude	FCU FPA	FCU Mode	Flaps	Max Speed	Mental Model	Pitch
1	Other	200	3000	3291	-1/50	Other	Retracted	400	Level	-1/100
	Commands: Aircraft flies, Pilot dials descend									
2	V/S FPA	201	2989	3291	-1/100	V/S FPA	Retracted	400	Descend	-1/100
	Commands: Aircraft flies, V/S FPA mode actie, Pilot extends flaps									
3	V/S FPA	200	2988	3291	-1/100	V/S FPA	Extended	180	Descend	0
	Commands: Aircraft flies with flaps, Mode Reversion to OPEN CLB,									
4	OPEN CLB	201	2989	3291	-1/100	V/S FPA	Extended	180	Descend	0
	Commands: Aircraft flies with flaps, OPEN CLB mode active,									
5	OPEN CLB	200	2990	3291	-1/100	V/S FPA	Extended	180	Descend	1/50
	Commands: Aircraft flies with flaps									
6	OPEN CLB	190	3291	3291	-1/100	V/S FPA	Extended	180	Descend	3/100

Figure 7: Sequence of events leading to the automation surprise in SAL [1].

3.2 Definition of Automation Surprise

Palmer [16] defines automation surprise as "An automation surprise occurs when the automation behaves in a manner that is different from what the operator is expecting." Here, it is applied to the aircraft as a system, with the pilot as the operator and the aircraft's autoflight system as automation, which includes both the autopilot and flight management system. The pilot's expectations about the behavior of the aircraft is based on the pilot's belief about the current flight mode the aircraft is in and his expectation of the aircraft behavior resulting from this mode. When the pilot knowingly commands changes in the cockpit, then these changes are expected and anticipated.

Expectation for a certain aspect of dynamics of aircraft behavior can be decomposed into three basic components:

- Direction

- Magnitude
- Time

An example of such an expectation is an expected pitch change (aspect of dynamics of aircraft behavior) in which the pilot expects a certain direction (climb, descent, level), at a given rate (vertical rate) and at a certain time (e.g. now, in 2 minutes or at a certain waypoint). More concretely, when the pilot knowingly commands a descent with the vertical rate of 1000ft/min and engages the proper vertical pitch mode, then the expectations are:

- Direction: descent
- Magnitude: 1000ft/min
- Time: now

In this example the pilot purposefully commanded the change by setting vertical speed and engaging the vertical pitch mode. An automation surprise in such a scenario would be if the altitude suddenly starts increasing and the pilot did not intend to command a climb.

An automation surprise does not necessarily involve the automation acting in an unexpected way resulting from system changes. It may also appear through changes initiated by the operator of which he does not understand the consequences. If the pilot pushes a button and the aircraft behaves in different way than he is expecting, this can also be designated as an automation surprise. Another possibility of a non-traditional automation surprise is when the pilot forgets to perform a required action and unexpected actions are performed by the system. An example of such an automation surprise is a One-Two-GO Airlines flight in 2007 involving an MD82. In this flight the pilots forgot to push the TO/GO button and were surprised by

the subsequent actions of the aircraft. Obviously, this does not depict a typical automation surprise, nevertheless the pilots were surprised by the performance of the automation which resulted in an automation surprise (the pilots being surprised by the automation).

3.3 Detecting and Modeling Automation Surprises in SAL

To detect an automation surprise in SAL a state needs to be specified that is desired or not desired to be attained. In a network protection software such an undesired state can be a security breach. In the setup of this work the undesired state is an automation surprise. SAL like most model checkers works by examining state-space model associated transitions exhaustively to determine certain states are ever reached or if certain transitions occur that lead to this state.

SAL models, when built from scratch, need to be simple and are given more complexity progressively, this process is called CEGAR (as described in Section 2.2). First, when models are created SAL produces scenarios that do not violate any constraints in the models but are unrealistic, such as state transitions that make changes in variables that are excessively large. An example of such a change is if the aircraft descends from one state to the next state 1000ft or more. Second, constraints are added to make these state transitions more realistic. Third, if the undesired state is attained and the state transitions leading to that state are realistic, more complexity is added to the model, by adding more models and state transitions and making state transitions more complex. This means more actions are added or more agents are added that are involved in the process that is modeled. Starting from a model where an aircraft descends, pilots are added to make changes to the aircraft state and other actions, such as flaps extension, are added. In between any step, but especially when adding constraints, deadlocks may appear that make it impossible for SAL to transition from certain states because no state transition fulfills the preconditions of

a certain state transition. This means that the state in which the model deadlocks is described by state variables that are different from all state transition preconditions. State transition preconditions are values of certain state variables. In this case constraints have to be relaxed or state transitions have to be changed or added.

A SAL model intended to identify a state with an automation surprise needs to include the pilot's mental model that corresponds to the expectation the pilot has about the future state of the aircraft. As it is necessary to begin with as simple a model as possible, it is customary to create an input model based on a single state of the aircraft, such as pitch. Since SAL works with discrete states, expectation is codified as the expectation of the state following a state transition, i.e. the next state. Therefore, within the state transition the pilot's expectation about the next state is set in the mental model construct. If the expectation of the previous step does not match the actual state of the aircraft in the next step, SAL identifies it as an automation surprise. A state involving an automation surprise is the state that SAL attempts to prove is attainable (or not attainable). An analysis of the states SAL reaches can identify whether the state transitions are realistic.

In the SAL model created by Rushby (see Figure 12), the expectation is explicitly denoted as the three different pitch positions of the aircraft: climb, level, descent. The corresponding variable in reality is the pitch of the aircraft: when positive the aircraft is climbing, when zero the aircraft is level and when the pitch is negative the aircraft is descending. When expectation and reality do not correspond the automation surprise is flagged. In the initial state of the SAL model the aircraft is in cruise and the pitch is level. In the first state transition the pilots dials descent, which means that his expectation is that the aircraft will descend. The aircraft proceeds with the descent and the pilot issues no further vertical control commands. Consequently, the pilot's expectation is that the aircraft is descending as originally commanded. The expectation of the pilot is violated when the automatic speed protection engages and

the aircraft initiates a climb in a subsequent state. When the automation commands the climb, the mental model and expectation about future states of the aircraft of the pilot remains descent, since the change was not commanded by him. The difference between reality (positive pitch = climb) and mental model (descent) leads to the automation surprise.

3.4 Automation Surprises in WMC

Similar to model checking, a mental model construct for human agents is required in WMC to simulate an automation surprise. In contrast to SAL, WMC includes more and more detailed actions and they are defined in a workmodel that is separate from the agent models. The workmodel contains all the knowledge of how to perform the work in the form of actions, and passes the actions to the agent model to be executed. This architecture makes the creation of a mental beliefs construct in WMC straightforward as the beliefs construct can update itself in the background every time the human agent acts / initiates actions which cause state changes in its environment. Note that a beliefs construct about the current state of the aircraft does not exist in SAL and the expectations are directly compared to the real variables describing the state of the aircraft. In contrast, in WMC the expectations are compared to the pilot's beliefs about the state of the aircraft.

The other aspect of a mental model, the expectation of the pilot about the future state of the aircraft, as given in SAL, is also implemented in WMC. The expectations construct formulates the agent's expectation based on the actions he performs that cause changes in aircraft behavior. In this work, four basic aircraft variables are used to create the pilot's expectation: altitude, airspeed, vertical speed and heading. Automation surprises result from the pilot's expectations being violated about these variables. Additionally, there may be a surprise in regards to variables which are closely linked to these 4 states. Specifically, the latitude and longitude of the aircraft

can cause an automation surprise, however an unexpected latitude and longitude is the result of an unexpected heading. Also, the the throttle may lead to an automation surprise but is closely aligned with increased speed. Since the basic four values capture the behavior of these additional variables, they are are not implemented. These four values are sufficient to model all types of automation surprises that can lead to dangerous aircraft behavior that can appear in a current cockpit.

To formulate expectations, the aircraft variables that cause the changes to the basic aircraft variables need to be defined. This depends on the logic of the flight mode that is engaged. Different flight modes follow different guidance logic. Depending on the flight mode there are different target values the aircraft tries to maintain or achieve, which includes airspeed, vertical speed, altitude, latitude, longitude, throttle, heading, flight path angle and timing values. Depending on which guidance logic a flight mode follows it is possible to formulate expectations. A negative flight path angle leads to the pilot's expectation that the aircraft will descend, while a positive angle is leading to the expectation of a climb. All rules formulated for the expectations in this work can be found in the appendix Section A.1.

3.5 Implementing Automation Surprise in WMC

This leads to three additions that were required to implement automation surprise in WMC:

1. Mental belief resources
2. Pilot expectation resources
3. Actions to update and compare belief resources to expectation resources and flag an automation surprise if there is a discrepancy

The mental beliefs construct that was created in the extent of this work in WMC is a C++ construct, called map. The map is a table made of two columns. The

first column is the key value, and the second column is the mapped value. In the mental beliefs construct, the key value is the real resource, and the mapped value is the belief resource. The beliefs construct creates a copy of the current state of environmental variables every time a monitoring action takes place or when the pilot performs an action that leads to changes of the environment. Depending on how often a monitoring action is called, determines the latency of the values in the beliefs construct. Another feature of the beliefs construct is a history trace. To be able to monitor the change of beliefs over time, the past ten values of each belief is stored in the beliefs construct. This way it is possible to monitor the changes in beliefs over time, which allows the agent to track the trend of real resource change and draw conclusions about the aircraft.

The expectation resources are independent from the beliefs construct and are encoded as either positive, negative or neutral. For each of the four basic aircraft variables the direction is interpreted differently. For the altitude and airspeed this means whether it is increasing, decreasing or level / constant. For the vertical speed this means whether it is positive, negative or neutral. Lastly, for the heading the current heading (or the belief about the current heading) is set to 0, while an increasing heading (turn to the right in clockwise direction) means positive, a decreasing heading (turn to the left in counterclockwise direction) means negative and a steady heading means neutral.

Also expectations were implemented about flight mode logic. Some flight modes take the MCP values into account to construct the flight path plan, such as VNAV modes. When such a flight mode is engaged and the pilot is aware that it is engaged, his expectations about the altitude and the vertical speed are formulated depending on the MCP selected altitude and vertical speed. Other flight modes, such as G/S, which capture the glideslope of the ILS, do not depend on any of the MCP selected values. Final approach modes that capture the glideslope are supposed to follow the

glideslope until reaching the point at which the pilot takes over from the autopilot to perform the landing. In such modes, the pilot expects the aircraft to continue to descend until the runway is reached, independently from the values entered in the MCP. For example, when the G/S mode is engaged, it does not matter what altitude is selected in the MCP. This is why, the altitude can be set above the current altitude, to the go around altitude. Since the aircraft is in G/S mode in which MCP values are ignored, the pilots expectation about the altitude is decrease even when the MCP selected altitude is above the current altitude. However, in other cases the aircraft is in a flight mode that takes MCP values into consideration, such as VNAV modes. When the MCP selected altitude value is manipulated it will lead to a change in the real altitude subsequently. Depending on the new value, the pilot forms expectations about the following behavior of the aircraft. In concrete, this means that if the pilot sets the MCP selected altitude lower than the current altitude, then he will expect that the real altitude will start decreasing.

An automation surprise monitoring action compares expectations and beliefs at regular intervals of 20 seconds. In the next iteration of the WMC workmodel this will be changed to be linked directly to the times the pilot agent performs monitoring actions. To be able to compare beliefs and expectations it is necessary to transform the belief to make it comparable to the expectation. This means the belief has to be transformed to positive, negative or neutral depending on which of the four basic aircraft variables is considered. For example the belief about altitude has to be transformed to increasing, decreasing or stable depending on the direction the last ten values show. If the pilot's expectation does not correspond to his belief then an automation surprise is flagged. A threshold is set to make sure that the changes are not just within normal fluctuation. The exact algorithms and thresholds are shown in the appendix in Section A.2.

3.6 Comparison of WMC to SAL

This section compares the models used in WMC and SAL and the functionalities of simulation and model checking in general. Table 2 summarizes the differences between WMC and SAL and the difference in the respective models used for this work.

Table 2: Comparison between WMC and SAL.

WMC	**SAL**
Sophisticated models	Simple models, few actions
Limited to scenarios	Exhaustive state space search
Slow (one simulation takes time)	Fast (millions of runs in seconds)
Time modeled	No explicit modeling of time
High-Fidelity aircraft dynamics	Cannot handle continuity (state explosion)

WMC uses sophisticated models of the aircraft used in the simulation, the work performed by the agents and the human agents themselves. Aircraft dynamics are modeled in high-fidelity and work to be performed by pilot, air traffic controllers and automation is modeled with a high degree of detail. SAL is limited to the use of simple models with significantly less actions. This also limits SAL to a highly abstracted representation of aircraft dynamics, since continuity, such as given by aircraft dynamics leads to a state space explosion, which cannot be exhaustively checked by a model checker. At the same time this means that SAL can run through many more scenarios in a shorter time than WMC. With symbolic algorithms, SAL reduces the states to explore for an exhaustive state space exploration and can perform the search in a short time. In WMC, each scenario needs to be defined that is explored one at a time. Due to the high complexity of the models an exhaustive exploration of the state space is impossible in WMC.

Another key difference between WMC and SAL, but also for model checking and simulation in general, is the representation of time that SAL lacks. In SAL only the sequence of actions is shown by the action trace, while WMC can provide the aircraft state and actions performed by agents and the respective time.

The mental model of the pilot expectations in SAL are rudimentary, only comprising the pitch of the aircraft. This means that for each state transition where mental models change, the change needs to be explicitly defined by the modeler. When extending the mental model to comprise more variables than the pitch, all the mental model changes have to be explicitly stated. The mental model of the pilot expectations are formed depending on dynamic changes in the simulation. Also, WMC comprises a beliefs construct which is created and updated automatically. This means that the beliefs and expectations construct can be used to detect a wider range of automation surprises than model checking can. While in SAL a difference between expectations and state of the aircraft (in this case pitch) flags an automation surprise, in WMC it is the difference between pilot's expectations and beliefs. In the following chapter Methods, the way these two approaches work together and how the combination of both can lead to a more robust method to uncover automation surprises given the shortcomings of both methods, simulation and model checking, used is described.

CHAPTER IV

METHOD

To be able to compare WMC and SAL approaches modeling an automation surprise, the common scenario that has been successfully modeled in SAL previously was implemented in WMC. The mechanism to compare these two methods to model automation surprise is to incorporate a mental model including a beliefs and expectations construct, into WMC. Violated expectations result in an automation surprise. The WMC version that was available, had the capability to simulate several aircraft performing an approach guided by air traffic control. To be able to simulate the automation surprise resulting from the common scenario, several additional capabilities have been incorporated into WMC:

- Creation and integration of a beliefs construct as an Updateable World Representation and an expectation construct and the capability to detect automation surprises (as described in chapter 3).
- Addition of additional flight mode logic and associated actions.
- Scenario creation and experimental design.

In order to provoke the automation surprise, SAL analysis revealed that two events have to come together:

- The FCU altitude needs to be above the current altitude.
- The airspeed needs to equal or exceed the maximum airspeed.

The altitude and airspeed are the two dependent variables that are manipulated through the independent variables in the simulation to achieve the two events described above.

The independent variables that can be manipulated directly through the construction of scenarios in WMC are:

- The Go Around altitude.
- The speed at which flaps are extended.
- The altitude at which a level off is commanded.
- The duration for which the level off is commanded.

Scenarios comprising different combinations of these four independent variables enable to provoke the automation surprise. The experimental design section (Section 4.4) describes how the scenarios have been construed.

4.1 Method to translate between Frameworks

One objective of this work is to simulate a common scenario with SAL and WMC using the case study that involves the known automation surprise. The common scenario is used to illustrate how the two analysis methods can be used in combination. Since SAL can explore all possible states in a simple and highly abstracted model specification, it is used first to find interesting scenarios worth exploring further. Interesting scenarios in this case are where system behavior differs from the expectations about the state of aircraft of the pilot. This has been the circumstance in the present case study in which SAL provides a basic scenario where an automatic mode reversion causes an automation surprise.

Figure 8 illustrates the process, beginning by abstracting the scenario narrative of Tarom Flight 381 into a model specification for SAL, which then is exhaustively explored. The scenario is abstracted by breaking it down into a simpler scenario with few actions and preconditions. The preconditions given in the scenario narrative construct an initial state from which the model checker starts off. Next a property is defined and SAL tries to find a counterexample, a state in which the property does

not hold anymore. In this case the property is that expectations always correspond to the pilot's belief about the state of the aircraft. When SAL finds a counterexample, an action trace from the initial state to the state in which the property does not hold anymore is shown. A counterexample in this case is an automation surprise in which the behavior of the aircraft deviates from the expectations of the pilots.

In the next step, as shown in Figure 8, the event sequences or preconditions that the SAL counterexample trace reveals are the necessary preconditions for the examined scenario to occur (such as an automation surprise), are given to WMC. The preconditions describe the states and transitions that eventually lead to the specification violation. The requirements that are necessary to recreate the SAL preconditions are then modeled in WMC as a scenario. Scenario creation in WMC requires consideration of a significantly larger number of variables (because the SAL models are heavily abstracted), and comparable scenarios must be carefully designed. Specifically, the initial conditions and procedures need to be defined prior to running the simulation. In most cases, simulations are created that are both normative and deterministic so that they may be checked for errors. Once properly verified, then stochastic elements or off-nominal occurrences or behaviors can be interjected. The simulation then must be run enough times to capture the variability of interest often using either a full factorial design of experiments or a Monte Carlo approach, both of which are likely to require hundreds or thousands of runs. Depending on the complexity of the simulation and speed of the computational hardware this can take several minutes to several days.

In the subsequent step, the results of the scenario runs in WMC provide a more concise description of the preconditions that lead the examined scenario to appear, if the scenario appears at all. These results will provide feedback on whether the scenario in SAL is realistic and under which conditions it is likely to occur.

Figure 9 shows the method from another perspective describing how the two frameworks, SAL and WMC, fit together and what operation is needed to translate models between the frameworks.

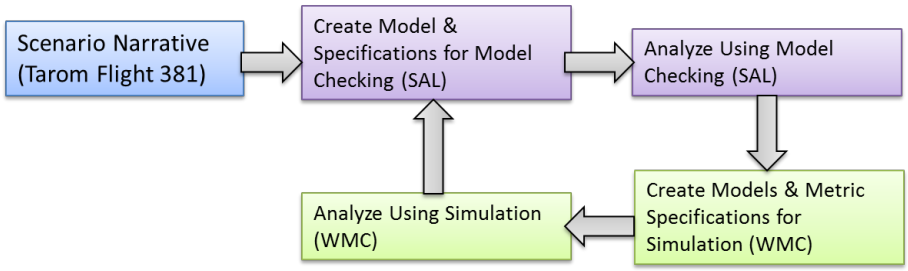

Figure 8: Iterative method starting from an initial scenario for SAL.

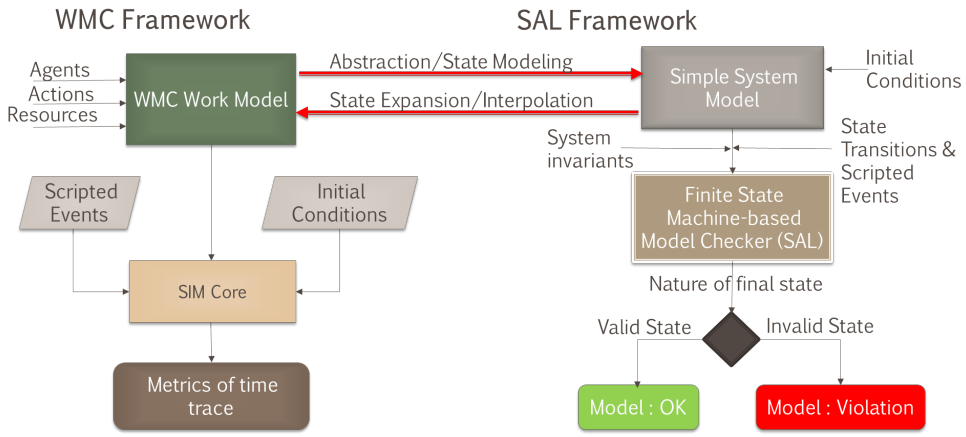

Figure 9: The two methods, SAL and WMC, and how they are connected in the method.

Building on SAL's finding, WMC provides a realistic scenario if it appears under the preconditions provided by SAL. This scenario includes a more precise description of the preconditions under which the specification violation (here an automation surprise) occurs and an outline of the sequence and times of events in the scenario.

After completing one round of analysis (Figure 8), one or the combination of several outcomes is possible:

1. The problems identified by the model checker are validated by the simulation.

2. The analysis suggests scenario variants based on the examined system behavior, meaning more cases are identified than the model checker initially predicted.

3. The model checker's findings are proved spurious.

In Case 1, the issue can be referred to a higher fidelity simulation facility, such as a cockpit flight simulator, for further validation. If system behavior that potentially leads to dangerous situations (such as automation surprise) is uncovered, then a solution should be sought to prevent the problem from occurring. In Case 2, the simulation may not only verify the issue found by the model checker, but may also find additional issues. In this case, the model checking input model should be expanded to verify the additional issues. In Case 3, it is discovered that a scenario leads to a dead end, meaning that such a scenario is an artifact of the model checking abstraction and is unlikely to appear in reality. These spurious scenarios are not worth exploring further and the abstraction used in model checking should be refined to eliminate them.

Additional iterations of the loop in the case of 2 or 3 can serve to further refine the specifics with the specification violation or expand the search space to find additional preconditions leading to the specification violation. Model checking refinement or expansion requires making the state transitions more complex and realistic or extending the model by adding states variables, state transitions, or extending the values of state variables.

This method is similar to CEGAR as described in Section 2.2. However, in contrast, the simulation results provide details for the model in model checking. Each major refinement of the model checking model goes through an iteration of simulation. In CEGAR iterations are only performed by refining model checking models. Obviously, when building the model in model checking CounterExample-Guided Abstraction Refinement is used to create and refine the model and the resulting action

trace that is realistic. Only then inputs and outputs of model checking are implemented in simulation.

4.2 Make SAL scenario appear in WMC

To make the SAL scenario appear in WMC as it does in SAL, changes in the WMC model is required. Implementing the case study in WMC, it was shown that the SAL scenario is not reproducible using a Boeing aircraft model on a the ILS approach as it is modeled in WMC. Therefore, it is not possible to achieve the same results in WMC as in SAL under the conditions given for an approach as modeled in WMC. One reason is that the aircraft model used, a Boeing 747 model, does not have the same maximum speeds as the aircraft model used in SAL and the real incident. The aircraft model used in WMC has a higher maximum speed with flaps extended, which means that at lower altitudes the mode reversion to OPEN CLB leading to the automation surprise is less likely to appear. Another part of the reason are the waypoints used for the approach. These waypoints model the average trajectory flown into LAX airport retrieved from statistical data. These data points do not permit the SAL mode reversion appear. The aircraft speeds at the altitudes in which the SAL model operates are lower in WMC and therefore lower than they have to be to overspeed and make the mode reversion emerge. What is necessary to make the same scenario appear in WMC as in SAL, is to create approach conditions that would allow the automation surprise to emerge in the same form in WMC.

Modifications that are required that will exactly replicate the undertying SAL model comprise:

- Using an aircraft model that corresponds to the aircraft that was involved in the Tarom 381 incident, the Airbus A320. This would provide lower maximum speeds at the altitude at which the mode reversion occurred in reality.

- Instead of using the average waypoint crossing speeds, model conditions in which

the aircraft approaches at a faster airspeed. This makes sense since in the case study the approach was not an average approach but the pilots were forced to recapture the ILS glideslope from above at a higher FPA. Therefore, the aircraft and the pilots did not have a target airspeed that was needed to be achieved at a certain waypoint.

- Alternatively, model a different airport where the average waypoint crossing speeds are lower than they are for LAX.

- Model a non-precision approach, instead of an ILS approach at an FPA of -3°, in which at low altitudes the aircraft is faster.

4.3 Changes in WMC

4.3.1 Aircraft Performance Model

The aircraft performance model used in WMC is a high-fidelity, 6 Degrees of Freedom (DoF) model of a Boeing 747. It uses a model of autoflight behavior that was used and validated in previous human-in-the-loop studies [13]. Since this is a model of a Boeing aircraft, the automatic speed protection of Airbus aircraft, was not included and needed to be implemented to simulate the automation surprise scenario. The provocation of the automation surprise in the WMC simulation requires aircraft dynamics for the aircraft model. However, since SAL uses no dynamics at all, it is sufficient to validate the results produced by SAL with any high-fidelity aircraft model. If it is possible to show that the results can be simulated using an aircraft model that comprises dynamics, then it can be shown that it is possible to go from a simple model, as given in SAL, to a more complex model that can then be used for simulations. The sequence of actions and the behavior do not change based on the type of dynamics that are modeled, but only the timing of the events leading up to the automation surprise is affected. Any set of "representative aircraft dynamics" are adequate to show and simulate the principle of this automation surprise.

Since a Boeing aircraft model is used as the baseline, no open modes, as they exist in Airbus aircraft, were available. The necessary changes in the scope of this work were to introduce a G/S capturing flight mode and the OPEN DES (Open Descent) and OPEN CLB (Open Climb) mode (see Table 3 for flight mode description). After modification of the aircraft model, the aircraft still performs the approach in a vertical mode (e.g. VNAV_PTH, Vertical Navigation Path), but then switches to the G/S mode when intercepting the ILS G/S. When reaching the maximum allowable airspeed, the automatic speed protection engages and the flight mode switches into an open mode (see Figure 10). The maximum allowable airspeed depends on the flaps extension degree. Besides the OPEN CLB mode that leads to the automation surprise, the OPEN DES mode was also implemented to be able to describe how often the automatic speed protection engages, and in which of these cases it leads to an automation surprise involving the OPEN CLB mode and in which the OPEN DES mode.

Table 3: Flight mode logic description.

Flight Mode	**Name**	**Logic**
VNAV_PTH	Vertical Navigation / Path	Follows a computed flight path
G/S	Glideslope	Follows the glideslope that is captured (e.g. ILS glideslope)
OPEN DES	Open Descent	Descends to the FCU altitude on engine idle at an attitude that allows maximum speed
OPEN CLB	Open Climb	Climbs to the FCU altitude at full throttle

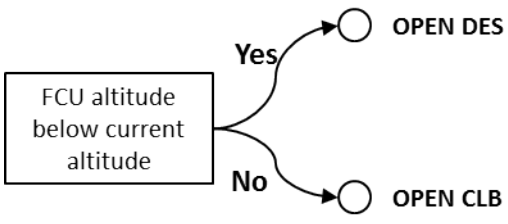

Figure 10: Mode transition logic when automatic speed protection is engaged.

4.3.2 Workmodels

The actions necessary to simulate the scenario of interest were implemented in the air traffic control and aircraft workmodels. These changes are low level and only concern the scenario at hand and, primarily concern actions necessary to transform a nominal descent into one similar to Tarom 381. To simulate the steep dive late in the arrival, it is needed to simulate ATC issuing a level off clearance and then release the aircraft back on its original path. The ATC workmodel needs to include actions to issue a level off, including all the actions that are required to do so. The first action is to issue the level off and the second action is to clear the aircraft to resume approach. In the aircraft workmodel, the error action of extending the flaps prior to the airspeed falling below maximum airspeed with flaps extended, was implemented.

4.4 Experimental Design

In contrast to model checking where millions of runs can be made within seconds and the entire model can be exhaustively evaluated, WMC like any other simulation platform is limited to only thousands of runs and only a partial investigation of the system. Thus, WMC scenarios have to be organized for the mode transition to appear.

The goal of the WMC simulation is to:

1. Verify that the action sequence predicted by SAL to be problematic continues to be problematic.

2. Refine SAL's prediction to include specific temporal relationships between events.

Scenarios have been created that involve different combinations of values for the variables that define the scenario (see Table 1), that serve as independent variables. All of these variables impact the occurrence of an automation surprise linked to the automatic speed protection logic. To provoke an automation surprise, adequate yet realistic values have been selected. The selection of the variables is described in the

following sections. Different combinations of independent variable values have been selected to form a scenario. Also the exact preconditions that were given in SAL that made the automation surprise appear were used for scenarios. These were values for the go around altitude to which the aircraft climbed after the mode reversion to OPEN CLB, and the speed at which the pilot agent extended the flaps prematurely. With the preconditions given from SAL it was possible to see whether these preconditions that lead to the automation surprise in SAL also lead to an automation surprise in WMC.

4.4.1 Variables for Scenarios

The table below (Table 4) summarizes the actions that impact the occurrence of the automation surprise and to what variable the action is linked. The Go Around altitude and the altitude at which the level off is commanded are defined by an altitude, the duration for which the level off is commanded is defined by a time interval and the speed at which the flaps are extended is defined by a speed. Different altitudes, speeds and times at which these actions were performed define a scenario.

Table 4: Variables used to define the simulation scenarios.

Action / Variable	**Variable Set**	**Description**
Go Around Altitude	Altitude	What is the go around altitude that is set in the MCP
Extend Flaps	Speed based	What is the speed of the aircraft when pilots extends flaps
When Level Off	Altitude based	At what altitude is the level off commanded by ATC
Level Off Duration	Time interval	For how many seconds does the aircraft level off

4.4.2 Go Around / Missed Approach altitude

One scenario variable is the go around altitude that is programmed into the MCP. A Go Around or a Missed Approach is a procedure that the pilot performs if he cannot guarantee a safe landing. On a go around procedure the pilot aborts the landing,

climbs to the specified go around altitude circles around and attempts another landing following ATC instructions. Usually, approach procedures specify the go around altitude and route to be taken for a go around. A go around may be necessary if the pilot cannot see the runway at a certain altitude called the decision altitude. The decision altitude is where the pilot has to decide whether he can guarantee a safe landing and if he cannot, the landing has to be aborted and the go around procedure has to be performed. The go around altitude can be found in the ILS procedure document for each runway. Since this altitude plays an important role in the appearance of the examined automation surprise it is a variable for the scenarios.

Considerations for the limits of the go around altitude:

- A level off must make sense from ATC perspective, if the aircraft's altitude is too low a level off poses problems to pilots and would not normally be issued.
- Needs to be at an altitude at which a level off and the subsequent steeper descent can still make the aircraft exceed the maximum speed to provoke the mode reversion.
- Should be set lower than the highest altitude an airport actually decides to set the missed altitude, since the missed approach altitude cannot be higher than an airport actually decides to set it. Usually the missed approach altitude is between 500 feet and 6000 feet.
- Should be set lower than the ILS intercept at 10000 feet.

This results in the following thresholds in between which the scenario variables for the go around altitude have to be situated.

The intervals are set to 500ft. This leads to the following seven altitudes (in feet): 3000, 3500, 4000, 4500, 5000, 5500 and 6000.

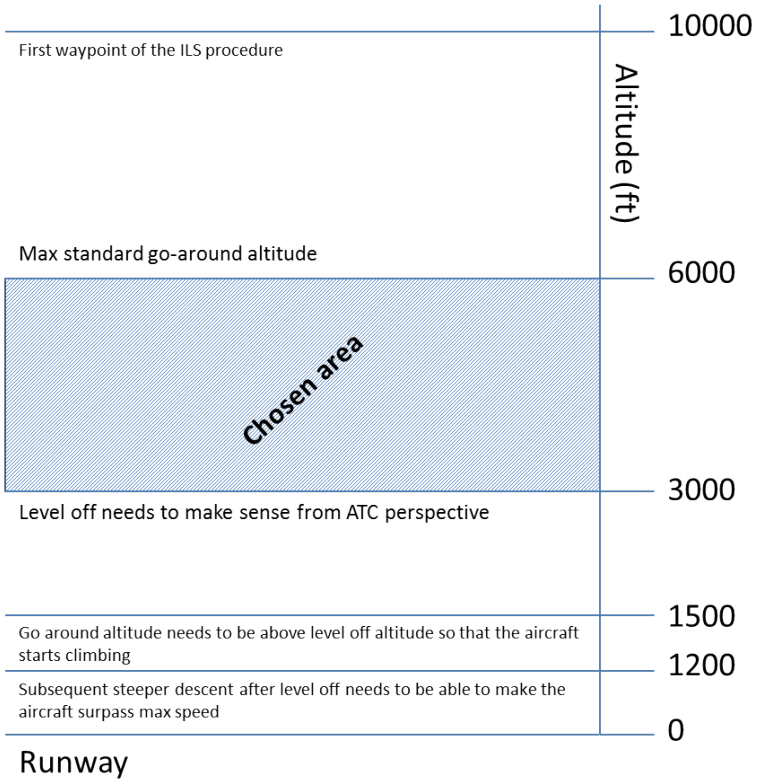

Figure 11: Determining the lower and upper limit for the go around altitude.

4.4.3 When is Go Around altitude set

Another scenario variable that may need to be considered is the time or altitude at which the go around altitude is set in the MCP. This altitude can be above the go around altitude or below it. This is due to the reason that the go around altitude is the last altitude that is set in the MCP. The altitude is defined in the ILS procedure and is set after the final clearance to the runway. However the altitude must be set in the MCP before the decision altitude so that when the aircraft reaches this altitude, in case a go-around is necessary the pilot can flip the TO/GO switch and the aircraft can automatically climb to the go around altitude. The absolute earliest time to set the go around altitude is when the last clearance to the runway is given and the aircraft is in the G/S mode in which MCP selected values are ignored.

The pilot is supposed to set the go around altitude in the MCP only after the

aircraft is established on the ILS glideslope. In the reference scenario based on Tarom 381, however the aircraft was never established on the glideslope since the instruments could not capture the glideslope, which is why the pilots disconnected the autopilot. In such a case pilots are not supposed to enter the go around altitude into the FCU (since it was an Airbus) yet.

This leads to the conclusion that the time when the go around altitude is set in the MCP should not be varied. It is instantiated in the simulation that the pilot sets the go around altitude in the MCP after he receives the last clearance and the aircraft is established on the ILS glideslope.

4.4.4 Time at which Flaps are extended

The extension of flaps is necessary on approach to reduce the speed at which the aircraft can be safely flown and to decrease the angle of descent. The flaps lower the stall speed and increase drag. When flaps are extended the maximum allowable speed for the aircraft decreases, due to structural limits. They should only be extended to the next setting after the speed has fallen below the maximum airspeed on that setting. The more the flaps are extended, the lower the maximum airspeed becomes. In the case of Tarom 381 the flaps were extended above the appropriate speed threshold causing an overspeed (195 knots). Therefore, flaps are set to 20 degrees by the pilot prematurely.

There are three general times that the flaps can be extended (to the described flap setting degree of 20):

- Before the level off is initiated, this is mostly necessary if the level off is commanded late (on a low altitude).
- During the level off, this is possible but less likely since generally the pilot will not slow down during a level off or will slow down at a lower rate than when descending. Therefore, reaching the speed for flaps extension is unlikely.

- After the level off, if the pilot did not extend the flaps beforehand then the pilot is going to extend them soon after finishing the level off.

To make the early extension of the flaps an actual error, the pilot should extend the flaps while the current speed is the same or still above the maximum speed of the flap setting. This way it is assured that the automatic speed protection engages.

As mentioned the most important consideration for the lower limit is that the current speed must be above the maximum allowable speed with flaps extended to 20 degrees. If this condition is not fulfilled the aircraft will not overspeed based on early flap extension as it has been the case in the reference scenario with Tarom 381. Since the pilot does not monitor the airspeed at all times this leads in some scenarios to the pilot extending the flaps when below the maximum speed. Therefore, the lower limit for speed at which flaps are extended to 20 degrees is 205 knots, which is the maximum speed at the flaps extension setting.

Since the pilot monitors the airspeed from as low as every 3 seconds to every 60 seconds, the upper limit for speed of flaps extension should be sufficiently large that if the check is performed late the aircraft still overspeeds when flaps are extended. However, it is not realistic that the pilot extends flaps while the current speed is more than 20 knots above the maximum speed of the aircraft with extended flaps. Therefore, the upper limit for speed at which flaps are extended to 20 degrees is 225 knots.

This leads to thresholds of 205 knots and 225 knots. The intervals are set to 5 knots. This leads to the following five speeds (in knots): 205, 210, 215, 220 and 225.

4.4.5 Altitude at which Level Off is commanded

Another scenario variable is when the level off is commanded by ATC. A level off means that the pilot aborts the descent and flies at a constant altitude. This may be necessary if two descending aircraft come too close and therefore the trailing aircraft

has to level off to maintain separation distances. Other reasons can be general traffic in the area. A level off while on a STAR (Standard Terminal Arrival Route) path, which is the first phase of the descent, is usually not problematic since the aircraft is still at a high altitude and has sufficient time to perform adjustments needed to return to the correct path. Also, the requirements for a STAR path are very relaxed, waypoints are far apart and there are few trajectory and speed constraints. It becomes more problematic if the aircraft is on the ILS path in the final approach phase. On the ILS path waypoints are close, and the aircraft has to follow the glideslope which means that if the aircraft levels off, it has to recapture the glideslope when descent is resumed. Also there is less time since the aircraft is already close to the runway and adjustments to the trajectory and speed have to be performed quickly.

The lower limit for the altitude for the level off is similar to the altitude at which the go around altitude is set. Low level offs are unlikely since at low altitudes it may not be possible for the aircraft to recapture the glideslope before the decision altitude. If at low altitude an action is required to separate the descending aircraft from other traffic, it is more likely that ATC will vector the aircraft to avoid the traffic. Alternatively, ATC may command the pilot to perform a missed approach following the missed approach procedure.

If the altitude is set too high then the pilot will have sufficient time to recapture the glideslope and problems are unlikely (only if the level off is commanded for a considerably long time which is usually not observed in reality). However, since the time necessary for the aircraft to recapture the glideslope depends on how long the level off takes the upper limit is set to the altitude of the first ILS waypoint, which is usually around 10000 feet.

This leads to thresholds of 3000 feet and 10000 feet, lowest go around altitude and highest ILS altitude. The intervals will be set to 1000 feet. This leads to the following eight altitudes (in feet): 3000, 4000, 5000, 6000, 7000, 8000, 9000, 10000.

4.4.6 Duration of the level off

Another variable is the duration for which the level off is commanded. The longer the level off, the steeper and the longer the aircraft has to fly to recapture the ILS glideslope. A steeper descent leads to a larger speed gain which makes overspeed and subsequently the appearance of the automation surprise more likely.

Little information is available on the impact a level off in the terminal area has on the speeds in the subsequent dive to recapture the glideslope. Furthermore, it takes the aircraft some time to level off. Therefore, it was decided to set the level off duration between 30 and 100 seconds at intervals of 10 seconds. This leads to the following eight level off durations (in seconds): 30, 40, 50, 60, 70, 80, 90, 100.

4.4.7 Inputs from SAL

In contrast to WMC, SAL runs through all scenarios that can emerge under all modeled circumstances. WMC is a simulation platform that needs a significantly higher demand of computational resources to run a scenario simulation, while SAL only needs a fraction of a second since the models that make up a SAL specification are much simpler and abstract. WMC's contribution is that the output of the simulations is more specific and possibly insightful, due to a more expanded set of action space to trace, real aircraft dynamics, and time modeled explicitly including the sequence of the events and the time of their occurrence. SAL only shows the state transitions that lead to the automation surprise since the objective of model checking is to prove the occurrence of an event and show the critical sequence of state transitions that leads to the undesired state (or prove that a state cannot be reached).

To compare methods and determine how they can be used in a complementary nature, the same conditions that have been given in SAL leading to the automation surprise have to arise at some point in the WMC simulation. The necessary conditions / preconditions in the SAL model that lead to the automation surprise are:

1. The aircraft is on descent
2. The pilot's expectation about future states (mental model) is descent
3. The FCU altitude is above the current altitude
4. Flaps are extended
5. Airspeed is higher than maximum airspeed

When these conditions are given, the mode reversion and subsequently the automation surprise should appear in WMC as it does in SAL. If these conditions indeed lead to the automation surprise in the simulation, then model checking can serve to WMC simulations by uncovering potentially dangerous scenarios. If, for a scenario, the same conditions lead to the same result in both SAL and WMC, then it is safe to assume that this scenario may be problematic for a real aircraft. WMC, with its extensive outputs, successfully coupled with SAL's scenarios, can provide information about the events and their respective times that lead to the problem.

In SAL there is a simplified set of variables and actions that are involved in the automation surprise. In the SAL specification there is no level off involved that leads to the overspeed because SAL models can be easily tailored to invoke any sequence of actions beginning at any point in the approach. The action leading to the automation surprise is extending the flaps before the speed of the aircraft is below the maximum speed with flaps extended. SAL cannot provide scenario variables for the level off altitude and duration since the level off is not modeled. Furthermore, the go around altitude is assumed to be set in the SAL model. This leaves two variables that are provided through the SAL model: the go around altitude and the speed at which flaps are extended. These are the values given by the SAL model. To verify these values they were used as the input variables for a set of scenarios.

Table 5: Inputs from SAL for WMC

Action / Variable	**Variable Set**	**Value**
Go Around Altitude	Altitude	3291 feet
Extend Flaps	Speed-based	201 knots

However, there are two points that have to be considered when translating these results from SAL to WMC:

- Flaps are not extended at a given speed in SAL but depend on two conditions: the pilots expectation about future states is descend (in SAL called mental model) and flaps are retracted. The second condition is needed so that the pilot can extend the flaps only once.

- In the SAL model an Airbus A320 aircraft is assumed that has a maximum speed of 180 knots with flaps extended in comparison to the Boeing 747 that has a maximum speed of 205 knots with flaps extended. Therefore, there are two scenarios that have to be run to verify these results: once where flaps are extended at 201 knots as in SAL, which is unlikely to lead to the mode reversion since it is below the maximum airspeed of 205 knots. The second one where flaps are extended at an airspeed that is 21 knots above maximum speed (226 knots) as it is given in SAL.

Step	Flight Mode	Airspeed	Altitude	FCU Altitude	FCU FPA	FCU Mode	Flaps	Max Speed	Mental Model	Pitch
1	Other	200	3000	3291	-1/50	Other	Retracted	400	Level	-1/100
	Commands: Aircraft flies, Pilot dials descend									
2	V/S FPA	201	2989	3291	-1/100	V/S FPA	Retracted	400	Descend	-1/100
	Commands: Aircraft flies, V/S FPA mode actie, Pilot extends flaps									
3	V/S FPA	200	2988	3291	-1/100	V/S FPA	Extended	180	Descend	0
	Commands: Aircraft flies with flaps, Mode Reversion to OPEN CLB,									
4	OPEN CLB	201	2989	3291	-1/100	V/S FPA	Extended	180	Descend	0
	Commands: Aircraft flies with flaps, OPEN CLB mode active,									
5	OPEN CLB	200	2990	3291	-1/100	V/S FPA	Extended	180	Descend	1/50
	Commands: Aircraft flies with flaps									
6	OPEN CLB	190	3291	3291	-1/100	V/S FPA	Extended	180	Descend	3/100

Figure 12: Sequence of events leading to the automation surprise in SAL [1].

4.4.8 Total number of scenarios

The four scenario variables of interest in full factorial combination lead to 8*7*8*7 = 3136 scenarios.

Table 6: Variable values for scenarios

GA Altitude	**Speed Flaps**	**Level Off Altitude**	**Level Off Time**
3000 feet	201 knots	4000 feet	30 sec
3291 feet	205 knots	5000 feet	40 sec
3500 feet	210 knots	6000 feet	50 sec
4000 feet	215 knots	7000 feet	60 sec
4500 feet	220 knots	8000 feet	70 sec
5000 feet	225 knots	9000 feet	80 sec
5500 feet	226 knots	10000 feet	90 sec
6000 feet			100 sec

CHAPTER V

RESULTS

The results of the simulations demonstrate how WMC can provide details that SAL cannot provide due to its simple models. Not only are the SAL results shown to be valid, also several other potentially dangerous scenarios linked to the automatic speed protection were uncovered. The results revealed that if the automatic speed protection were still implemented in its original form it could lead to many different types of potentially dangerous scenarios surprising the pilots beyond the automation surprise in the Tarom 381 incident.

5.1 SAL matching case

As described in the Methods chapter 4, changes to the WMC model are required to make the SAL scenario appear with the same or a similar altitude profile. To be able to reproduce the SAL scenario the existing WMC model was changed to match the SAL model and conditions. The Boeing 747 model was modified to match the maximum speed of 180 knots at 20 degrees flap extension. The target speeds for waypoints below 4000 feet were adjusted to be greater than 180 knots. The level off was commanded at about 3000 feet and flaps extension was forced to be performed after the level off. When these preconditions are given the SAL scenario appears in WMC. Figure 13 shows the altitude profile of the WMC scenario and Figure 14 shows the altitude profile of the original SAL scenario. Since SAL only models discrete states, altitude points connected by straight lines are drawn.

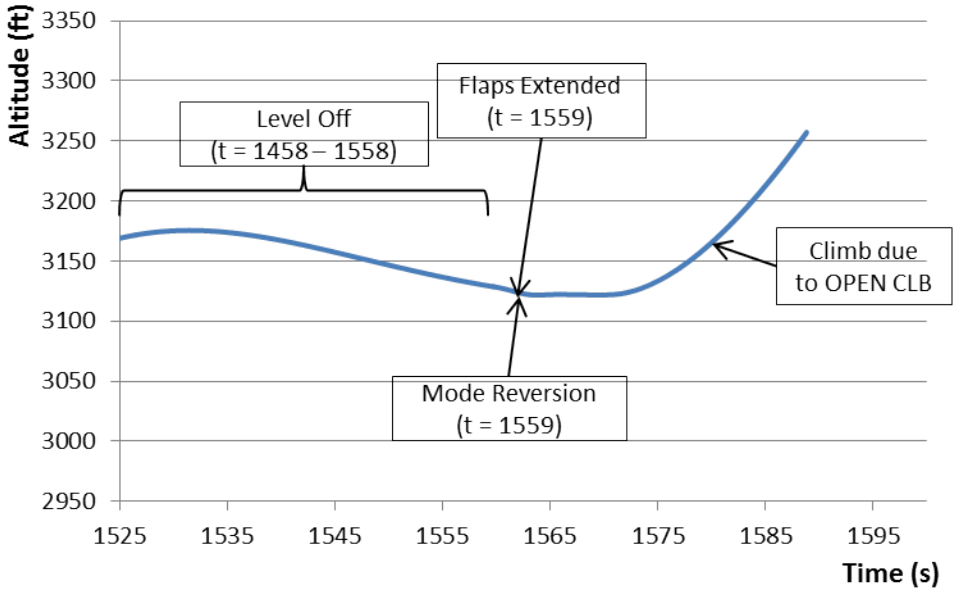

Figure 13: Altitude profile when SAL preconditions are applied to the WMC model. The aircraft levels off slightly above 3000 feet and after resuming descent, flaps are extended which leads the aircraft to overspeed and change modes to OPEN CLB. Subsequently, the aircraft climbs to the MCP altitude of 3291 feet causing an automation surprise.

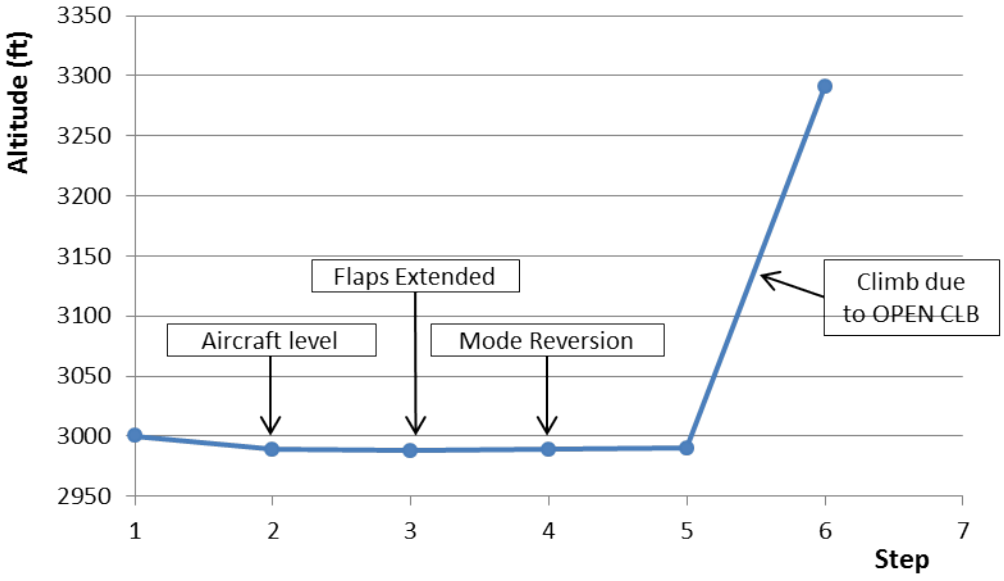

Figure 14: Altitude profile of the SAL scenario itself. Since SAL does not model time, the x-axis is showing the steps, representing time implicitly. The distance between steps is arbitrary and is not supposed to represent time. The altitudes are connected by straight lines.

Step	Flight Mode	Airspeed	Altitude	FCU Altitude	FCU FPA	FCU Mode	Flaps	Max Speed	Mental Model	Pitch
1	Other	200	3000	3291	-1/50	Other	Retracted	400	Level	-1/100
	Commands: Aircraft flies, Pilot dials descend									
2	V/S FPA	201	2989	3291	-1/100	V/S FPA	Retracted	400	Descend	-1/100
	Commands: Aircraft flies, V/S FPA mode actie, Pilot extends flaps									
3	V/S FPA	200	2988	3291	-1/100	V/S FPA	Extended	180	Descend	0
	Commands: Aircraft flies with flaps, Mode Reversion to OPEN CLB,									
4	OPEN CLB	201	2989	3291	-1/100	V/S FPA	Extended	180	Descend	0
	Commands: Aircraft flies with flaps, OPEN CLB mode active,									
5	OPEN CLB	200	2990	3291	-1/100	V/S FPA	Extended	180	Descend	1/50
	Commands: Aircraft flies with flaps									
6	OPEN CLB	190	3291	3291	-1/100	V/S FPA	Extended	180	Descend	3/100

Figure 15: Shows the states and state transitions in the SAL model.

5.2 Overspeed Causes

For the exploration of the state-space with the four independent variables described earlier, however, the original Boeing 747 dynamics and operational procedures modeling, waypoints and the maximum speeds were used in their original form, as implemented in WMC.

In the case study there are two overspeed causes. The first is extending flaps before the speed falls below the maximum speed of the next flaps setting. The second is when the aircraft is above the ILS glideslope and has to dive to capture or recapture the glideslope similar to the Minnesota incident. In the Tarom incident both causes were combined, the aircraft did not capture the glideslope properly requiring a dive to recapture it. During the dive the pilots extended the flaps to the next setting prematurely, reducing the maximum airspeed below that of the aircraft's actual speed.

In this work, not capturing the glideslope was simulated by a level off requested by ATC after which the aircraft dives to recapture the glideslope. In the WMC simulations, overspeeds generated by both causes appeared, as well as an overspeed as a result of both causes at the same time. The combination of both causes means the aircraft dove to recapture the glideslope, extending the flaps decreased the maximum speed and the dive made the aircraft reach this new maximum speed.

5.3 Simulation Artifacts

Automation surprises also appear in the simulation which are not related to the speed protection logic. This is due to the simulation guidance logic that sets the target altitude to the altitude of the next waypoint, be it below or above the current altitude. This procedure is not a problem in itself, but waypoints are usually calculated by the FMS dynamically, rather than being fixed at the beginning of the flight. In this work, it becomes a problem in some cases where the aircraft levels off or climbs to reach the waypoint altitude while the pilot's expectation is that the aircraft will descend. Since, these automation surprises are not part of the behavior of interest and artifacts of the simulation guidance logic rather than real aircraft behavior, these are excluded from the results.

5.4 Grouping Results into meaningful Scenarios

In contrast to SAL which sought to prove or disprove a specific mode transition could occur and under what preconditions it would do so, simulations are rather open ended. With simulation we were searching for a more generic state of automation surprise arising out of the combination of the independent variables manipulated. The simulations showed three different outcomes, automation surprises involving the OPEN CLB or OPEN DES mode and scenarios that did not lead to a mode reversion. These outcomes were grouped based on their causes and are referred to as scenarios. Given the number of independent variables and the complexity of an actual simulation comprising flight dynamics, six such scenarios appeared in the results. Each scenario has its counterpart for OPEN CLB and OPEN DES, representing three stages or reasons for which a mode reversion occurs for each of the modes. The first mode reversion (Scenarios 1 and 2) appears due to early flaps extension before the level off, the overspeed occurs because flaps are extended early which sets the actual airspeed instantaneously to be above maximum. The second mode reversion (Scenarios 3

and 4) occurs after the level off during the dive for the same reason. The third mode reversion (Scenarios 5 and 6) occurs during the dive, the pilot extends the flaps at the correct time, however the speed gain through the dive makes the aircraft overspeed.

The following subsections describe the types of scenarios that appeared due to the automatic speed protection. Not all of the scenarios resulted in an automation surprise but some describe particular aircraft behavior linked to the automatic speed protection. Table 7 gives an overview of the scenarios that appeared.

Table 7: Overview of the scenarios that appeared. AS stands for automation surprise, (*) means an automation surprise occurred and it is unclear whether this is due to the simulation guidance logic or whether this is real behavior that has to be examined further and (**) means that even though an automation surprise occurred, this is due to the simulation guidance logic and unlikely to appear in reality.

Scenario	**Mode**	**AS**	**Description**
1	DES	No	Mode Reversion before level off, early flaps extension leads to overspeed
2	CLB	Yes	Mode Reversion before level off, early flaps extension leads to overspeed
3	DES	Yes*	Mode Reversion after level off, early flaps extension leads to overspeed
4	CLB	Yes	Mode Reversion after level off, early flaps extension leads to overspeed
5	DES	Yes**	Mode Reversion after level off, dive leads to overspeed on current flaps configuration
6	CLB	Yes	Mode Reversion after level off, dive leads to overspeed on current flaps configuration

For comparison Figure 16 shows the altitude profile of a nominal approach and Figure 17 shows the altitude profile of the Tarom 381 incident.

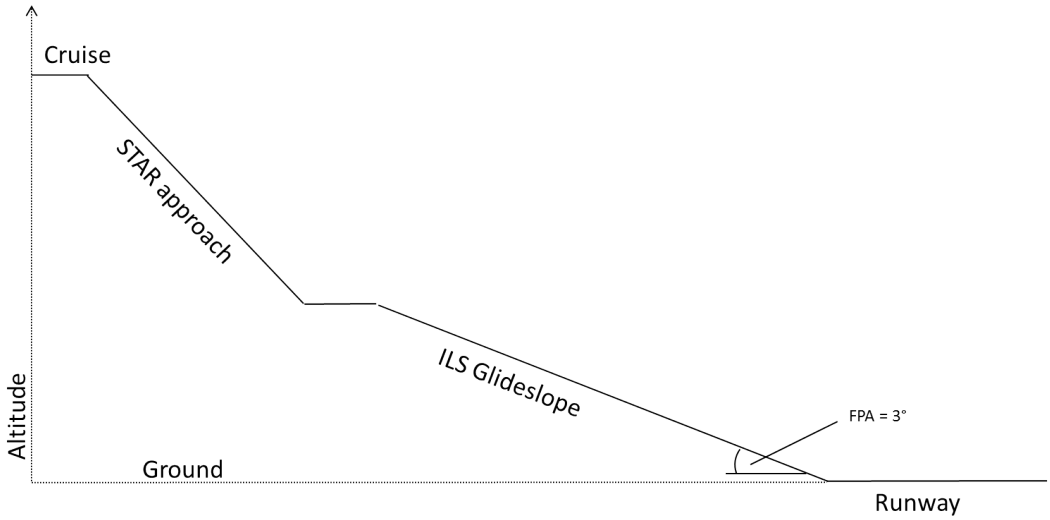

Figure 16: Altitude profile of a nominal approach.

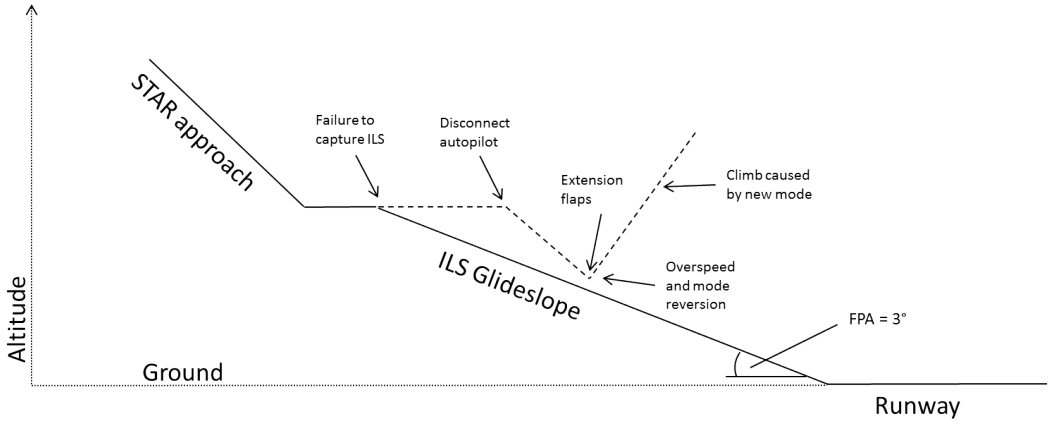

Figure 17: Altitude profile of the Tarom 381 approach.

5.4.1 Scenario 1: OPEN DES, No Automation Surprise, Flaps Extension before Level Off causes Mode Reversion

In this scenario, flaps are extended early before the aircraft is commanded to level off. The go around altitude is below the altitude at which the mode reversion occurs. The aircraft switches into OPEN DES, a change not noticed by the pilots because the mode switch does not change the pitch significantly. Subsequently, ATC commands to level off, which makes the pilots change the flight mode to ALT HOLD. Either the aircraft already has slowed down sufficiently or it does so while leveled off, which means the pilots never notice the mode reversion caused by the overspeed. Figure 18 shows the altitude profile of one of such scenarios and the altitudes and times at which the events occur, Figure 19 show a closeup of the time and altitude around the mode reversion. It is clear in this figure why the pilots might never notice the change in mode.

Figure 20 shows a two-dimensional selection of simulation runs and the respective scenarios that arose, including the runs that did not lead to an automation surprise. Since four variables are involved (go around altitude, flaps extension speed, level off altitude, level off duration), to create a two-dimensional plot, two variables have to be held constant. In this case, the go around altitude and the flaps extension speed were held constant at 3291 feet and 226 knots (21 knots above maximum speed), the way the automation surprise occurs in SAL. The different symbols show what scenario was observed at the run or if no automation surprise occurred at all (notice that Scenario 1 also does not involve an automation surprise).

5.4.2 Scenario 2: OPEN CLB, Flaps Extension before Level Off causes Mode Reversion

In this scenario the pilot extends the flaps early before the level off which leads to overspeed and mode reversion to OPEN CLB. The subsequent climb to the go around altitude makes the automation surprise appear. For this scenario, the go

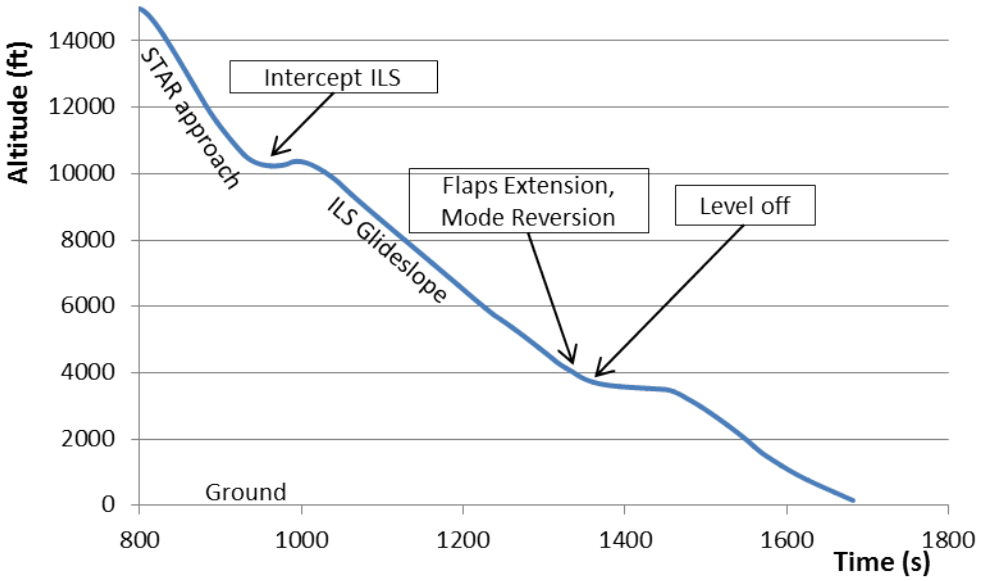

Figure 18: Exemplary altitude profile of the aircraft in Scenario 1 with altitudes in which the events occur. The scenario variables of this example are Go Around Altitude: 3000 feet; Flaps Extension Speed: 210 knots; Level Off Altitude: 4000 feet; Level Off Duration: 30 seconds.

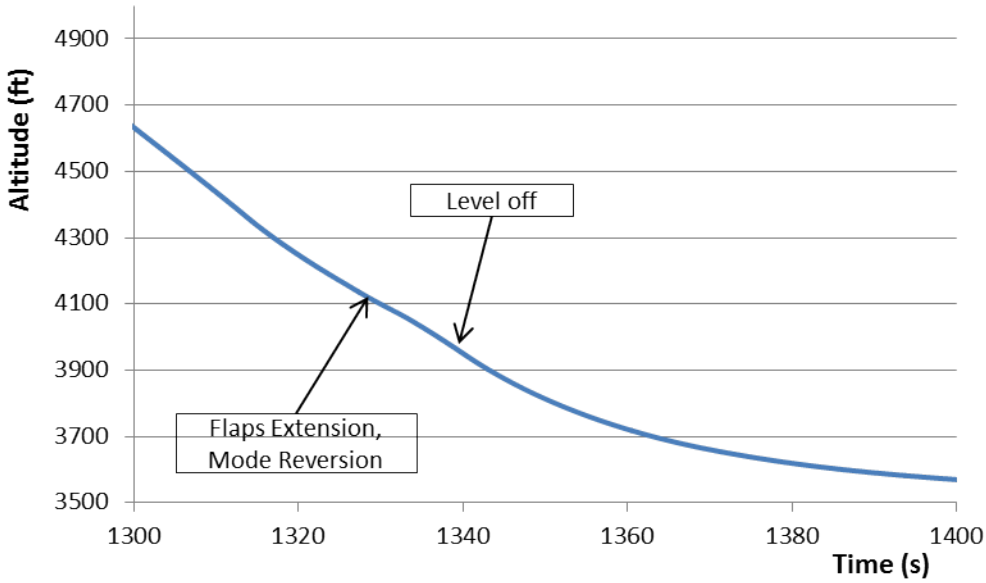

Figure 19: Closeup of the altitude profile of Scenario 1 in Figure 18 and the respective times and altitudes at which events and actions occurred. The pilot extends the flaps just before leveling off which causes an overspeed and a mode reversion to OPEN DES. When leveling off the pilot changes the flight mode to ALT HOLD and the automatic mode reversion is never noticed.

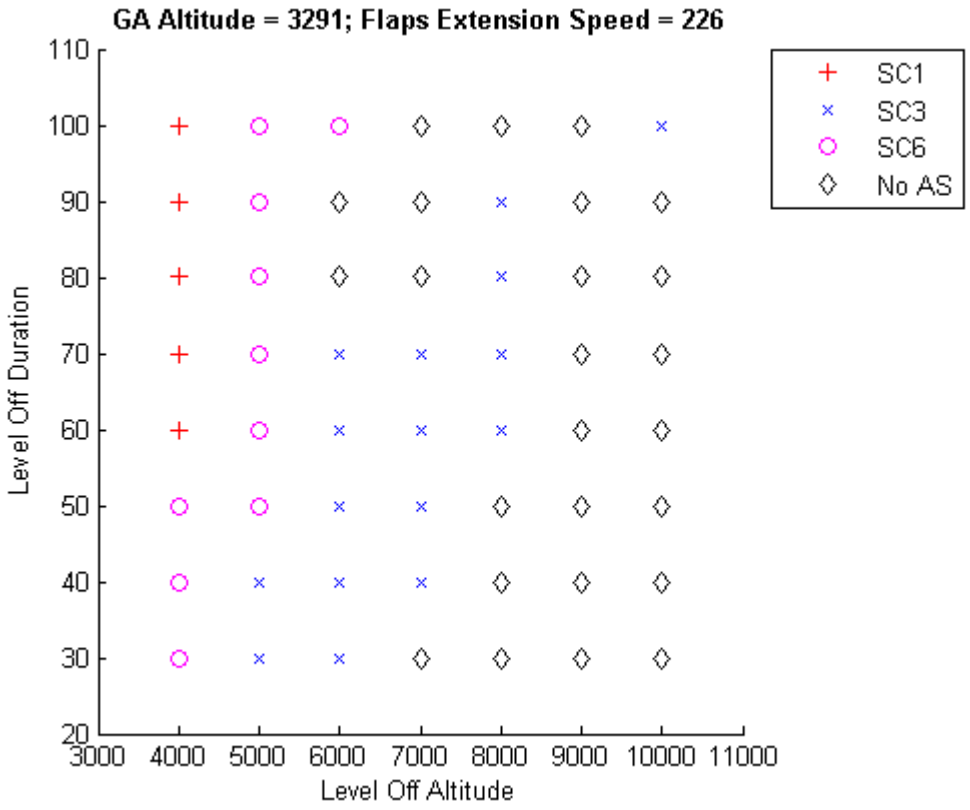

Figure 20: Conditions that make Scenario 1, Scenario 3 and Scenario 6 appear with the SAL preconditions Go Around Altitude: 3291 feet and Flaps Extension Speed = 226 knots. Note: Scenario 1 does not involve an automation surprise.

around altitude is above the level off altitude. Figure 21 shows the altitude profile of one of such scenarios and the altitudes and times at which the events occur. Figure 22 shows a range of preconditions that actually lead to this scenario or other scenarios. In this figure the go around altitude and the level off altitude are fixed. Figure 23 also shows a range of preconditions with the level off duration fixed instead of the level off altitude.

5.4.3 Scenario 3: OPEN DES, After Level Off, Extension of Flaps causes Mode Reversion, Guidance takes over after reaching MCP altitude

In this scenario, the aircraft overspeeds after the level off and switches into OPEN DES until it reaches the MCP altitude at which the aircraft levels off. Next, the guidance is activated which leads the aircraft to the following waypoint altitude; this

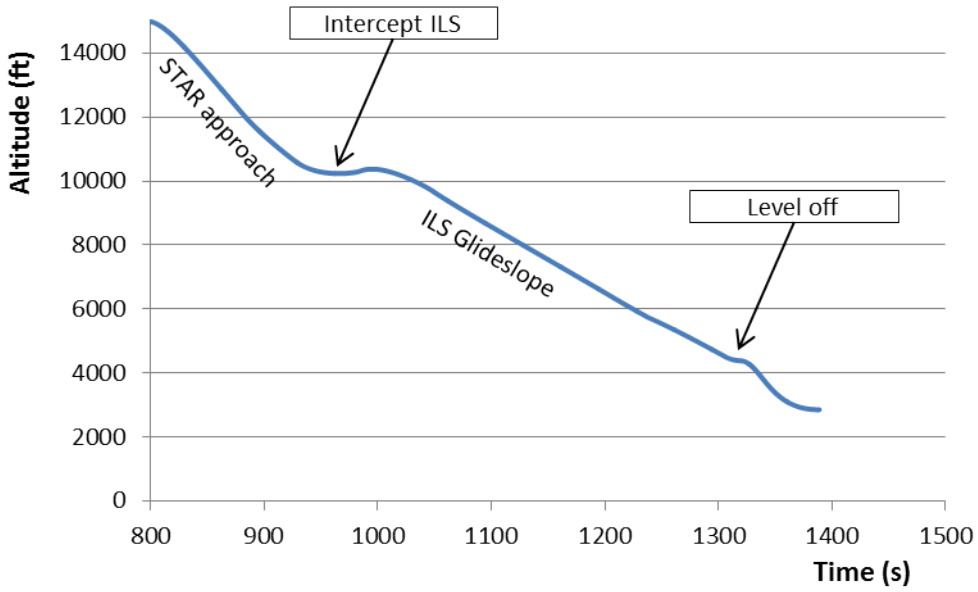

Figure 21: Exemplary altitude profile of the aircraft in Scenario 2 with altitudes at which the events occur. The scenario variables of this example are Go Around Altitude: 6000; Flaps Extension Speed: 210; Level Off Altitude: 4000; Level Off Duration: 50.

can be interpreted as a mode reversion to the VNAV mode. If the next waypoint altitude is below, the aircraft resumes its descent without the pilot noticing the two mode reversions. However, if the waypoint altitude is above, the aircraft climbs causing an automation surprise. Figure 24 shows the altitude profile of one of such scenarios and the altitudes and times at which the events occur, Figure 25 show a closeup of the time around the mode reversion. Figure 26 and Figure 20 show the preconditions that make Scenario 3 or other scenarios appear.

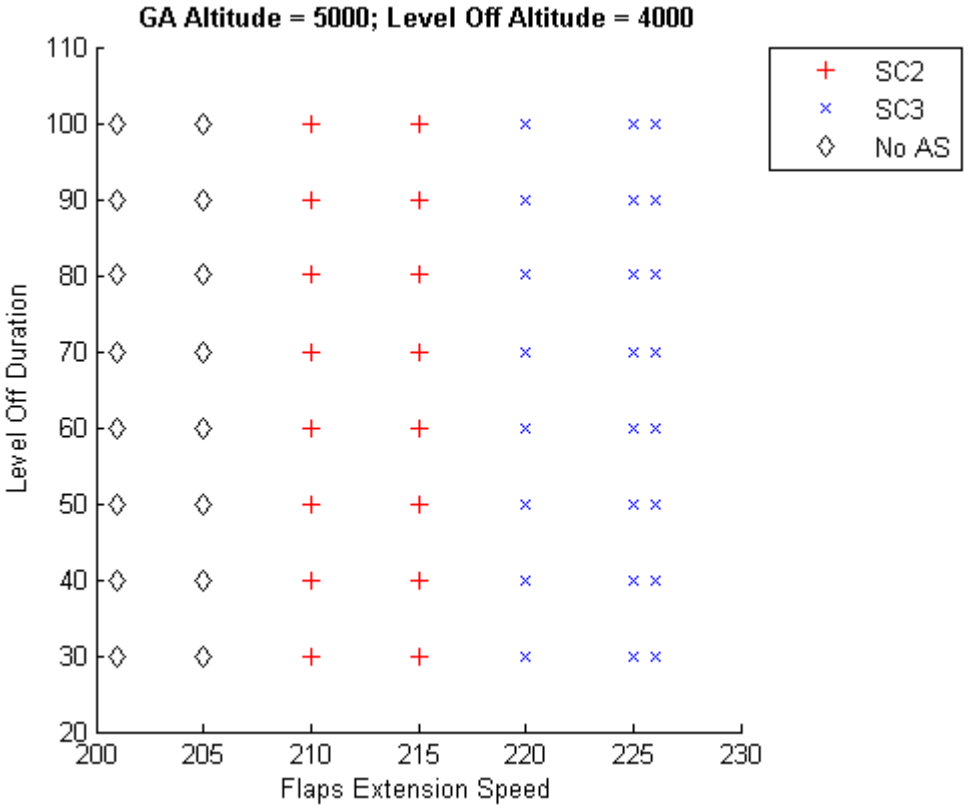

Figure 22: Conditions that make Scenario 2 and Scenario 3 appear with the preconditions Go Around Altitude: 5000 feet and Level Off Altitude: 4000 feet. Since flaps extension speeds of 225 knots and 226 knots (21 knots above maximum speed) are modeled, these data points are very close to each other.

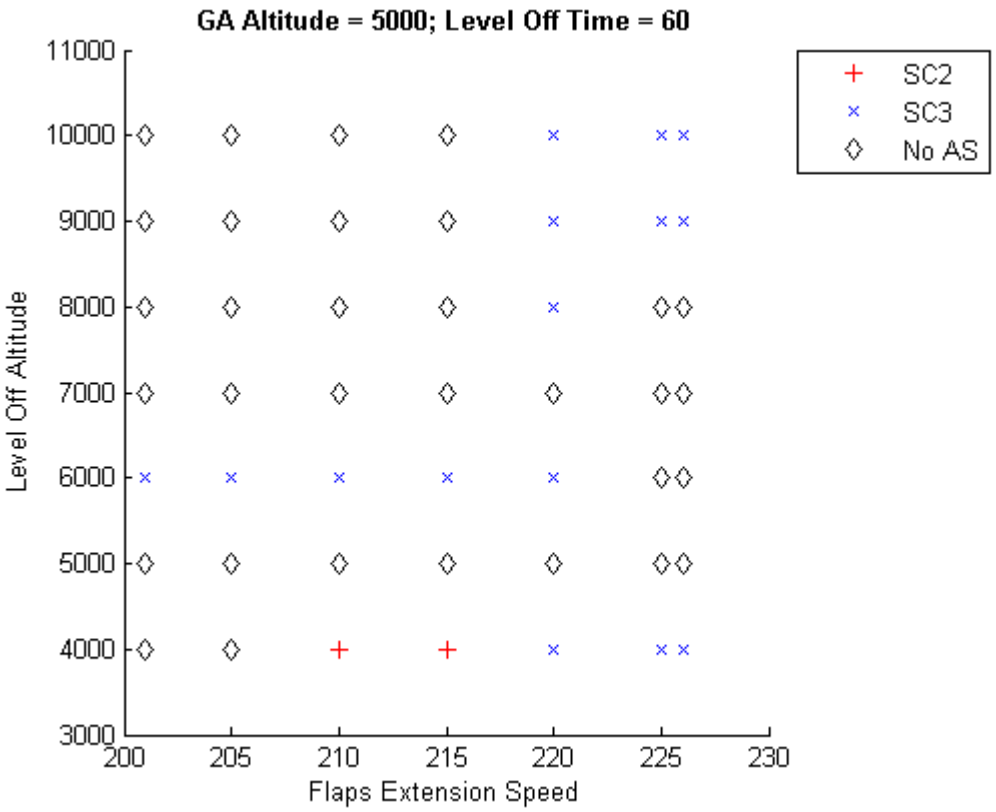

Figure 23: Conditions that make Scenario 2 and Scenario 3 appear with the preconditions Go Around Altitude: 5000 feet and Level Off Duration: 60s. This graph is similar to Figure 22, in this graph the level off duration is fixed at 60 seconds and the level off altitude is varied instead.

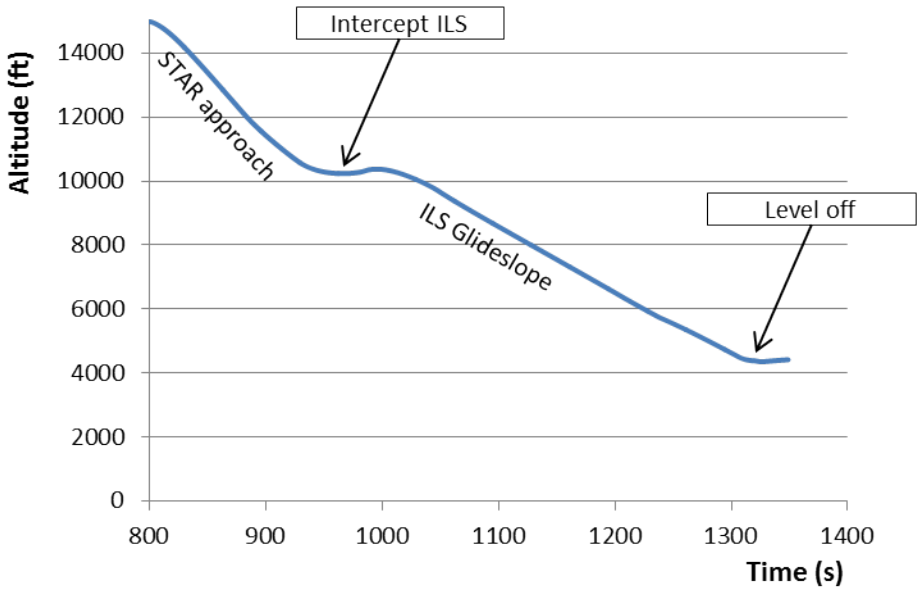

Figure 24: Exemplary altitude profile of the aircraft in Scenario 3 with altitudes in which the events occur. The scenario variables of this example are Go Around altitude: 3000 feet; Flaps Extension Speed: 210 knots; Level Off Altitude: 5000 feet; Level Off Duration: 40 seconds.

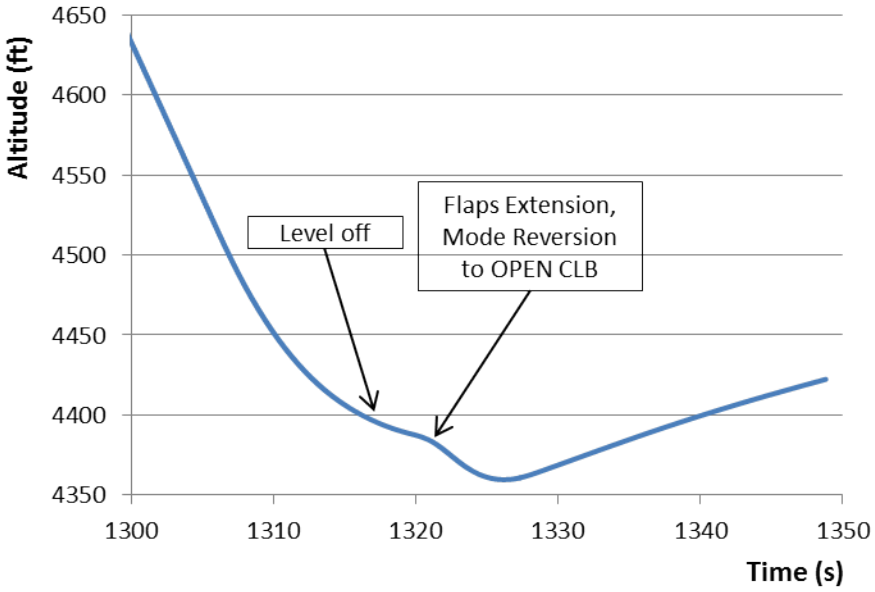

Figure 25: Closeup of the altitude profile of Scenario 3 in Figure 24 and the respective times and altitudes at which events and actions occurred. The level off occurs because the aircraft reaches the MCP altitude of 3000 feet. The subsequent mode reversion leads the aircraft to climb to the next waypoint altitude of 3500 feet. Since an automation surprise is already flagged after the level off the subsequent climb is not shown.

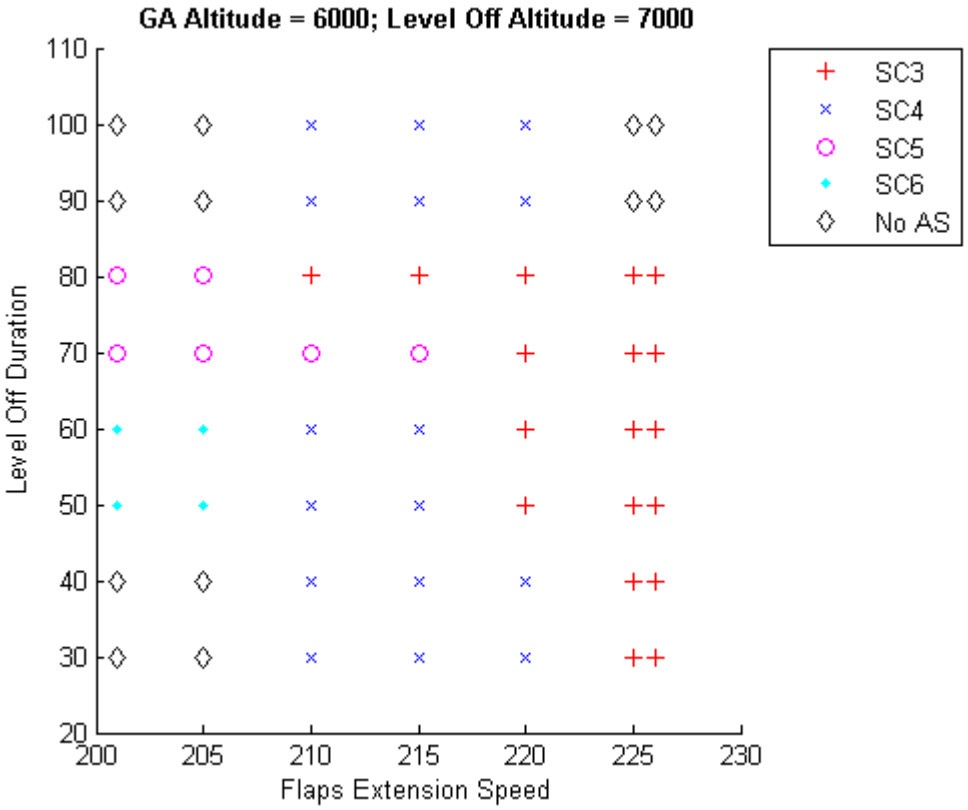

Figure 26: Conditions that make Scenario 3, Scenario 4, Scenario 5 and Scenario 6 appear with the preconditions Go Around Altitude: 6000 feet and Level Off Altitude: 7000 feet.

5.4.4 Scenario 4: OPEN CLB, After Level Off, Extension of Flaps causes Mode Reversion

In this scenario, the aircraft discontinues its level off and resumes the descent trying to recapture the glideslope. After a few seconds flaps are extended mistakenly early which leads the aircraft to exceed the speed limit with the new flaps setting. A mode reversion to OPEN CLB leads to a climb and an automation surprise is flagged. Figure 27 shows the altitude profile of one of such scenarios and the altitudes and times at which the events occur, Figure 28 show a closeup of the time and altitude around the mode reversion. Figure 26 shows the preconditions that lead to this scenario and other scenarios.

5.4.5 Scenario 5: OPEN DES, Dive causes Overspeed and subsequent Mode Reversion

In this scenario, the overspeed appears resulting from the dive and not due to flaps extended mistakenly early. The difference is that flaps are extended at the correct speed and the dive itself leads the aircraft to overspeed. The pilot extends the flaps after the level off at the correct speed to the next configuration (in this scenario lower than 20 degrees, since the flaps extension to 20 degrees is done early as commanded by the experimental design). However, the dive leads the aircraft to speed up and reaching the maximum speed with the new configuration and a mode reversion to OPEN DES. Since the new mode is OPEN DES and the aircraft continues descending this does not lead to an automation surprise. However, the leveling off due to the described guidance particularity leads to a level off and an automation surprise. In reality, this scenario with the change to OPEN DES would not lead to an automation surprise.

Another modeling problem in WMC is that thrust is increased after each flaps extension and this may be part of the reason that makes this scenario appear. It is possible that this scenario would not appear in reality the way it does in the

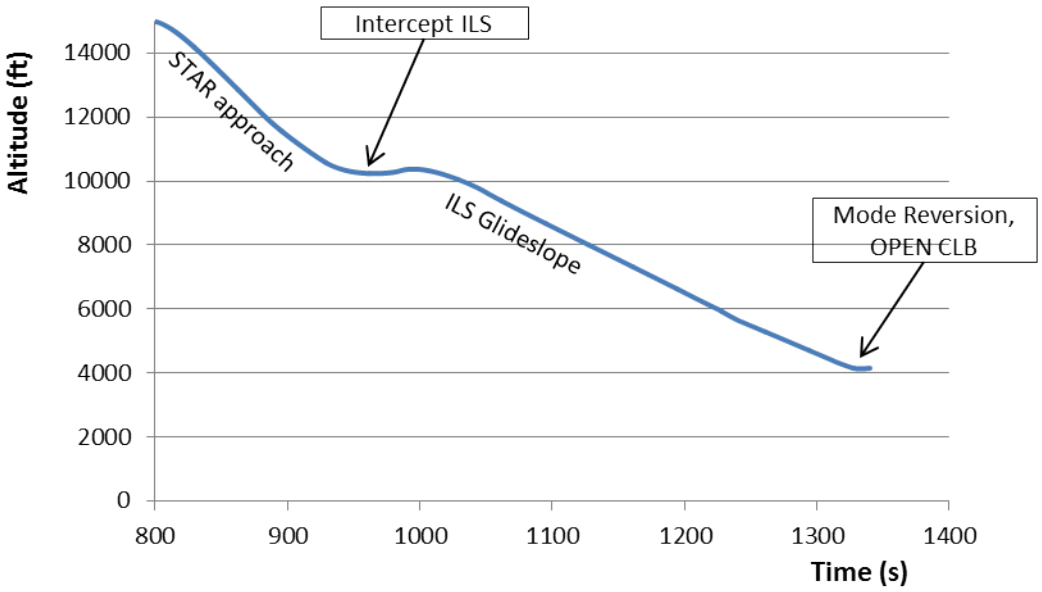

Figure 27: Exemplary altitude profile of the aircraft in Scenario 4 with altitudes in which the events occur. The scenario variables of this example are Go Around Altitude: 4500 feet; Flaps Extension Speed: 210 knots; Level Off Altitude: 5000 feet; Level Off Duration: 40 seconds.

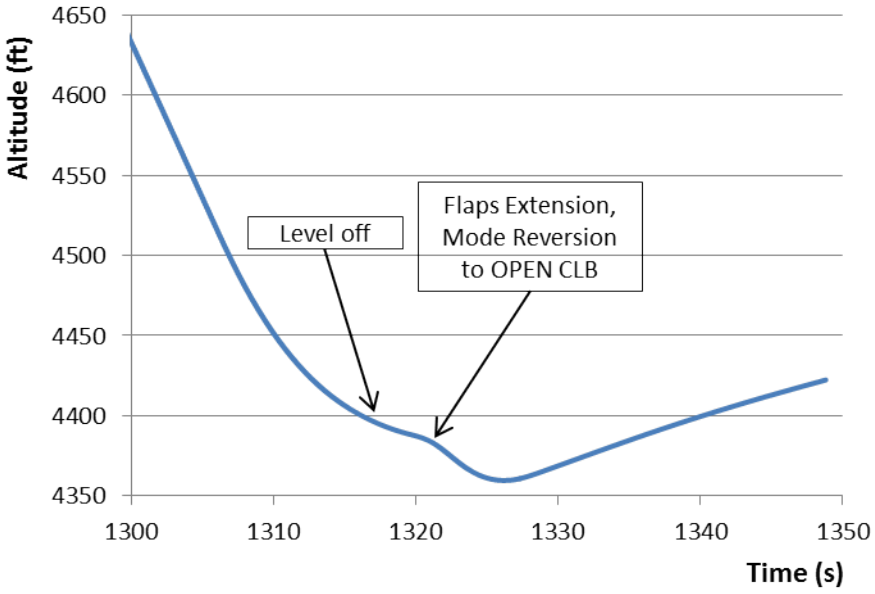

Figure 28: Closeup of the altitude profile of Scenario 4 in Figure 27 and the respective times and altitudes at which events and actions occurred. The level off occurs at 4400 feet after ATC commands it at 5000 feet. The aircraft only dives for a few seconds and then flaps are extended which leads to the mode reversion to OPEN CLB and the subsequent climb.

simulation.

5.4.6 Scenario 6: OPEN CLB, Dive causes Overspeed and subsequent Mode Reversion

As in scenario 5, in this scenario, the overspeed appears resulting from the dive and not due to flaps extended mistakenly early. This scenario is the counterpart to Scenario 5 with the difference that at the time of the mode reversion, the go around altitude is above the current altitude. The pilot extends the flaps after the level off at the correct speed to the next configuration (lower than 20 degrees). However, the dive leads the aircraft to speed up and reaching the maximum speed with the new configuration and a mode reversion to OPEN CLB. Figure 31 shows the altitude profile of such a scenario and Figure 32 shows a closeup. Figures 26 and 20 show the preconditions that make Scenario 6 or other scenarios appear.

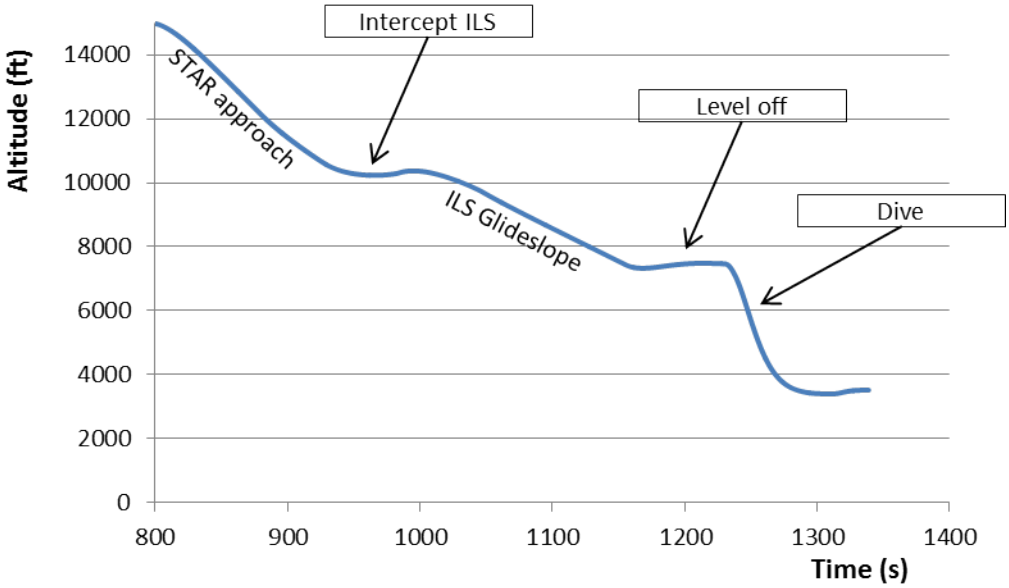

Figure 29: Exemplary altitude profile of the aircraft in Scenario 5 with altitudes in which the events occur. The scenario variables of this example are Go Around altitude: 3500; Flaps Extension Speed: 215; Level Off Altitude: 8000; Level Off Duration: 100.

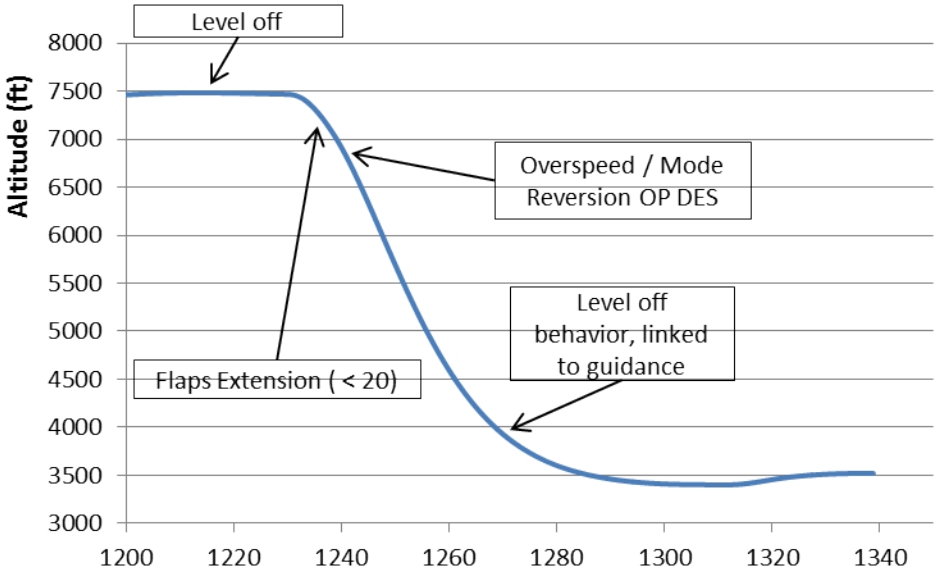

Figure 30: Closeup of the altitude profile of Scenario 5 in Figure 29 and the respective times and altitudes at which events and actions occurred. The level off occurs at about 7500 feet after being commanded at 8000 feet. The aircraft dives and a few seconds later flaps are extended to a lower configuration than 20 degrees, in this example to 5 degrees. The dive leads the aircraft to overspeed and the mode reversion to OPEN DES. The level off is a modeling particularity linked to the guidance and can be disregarded.

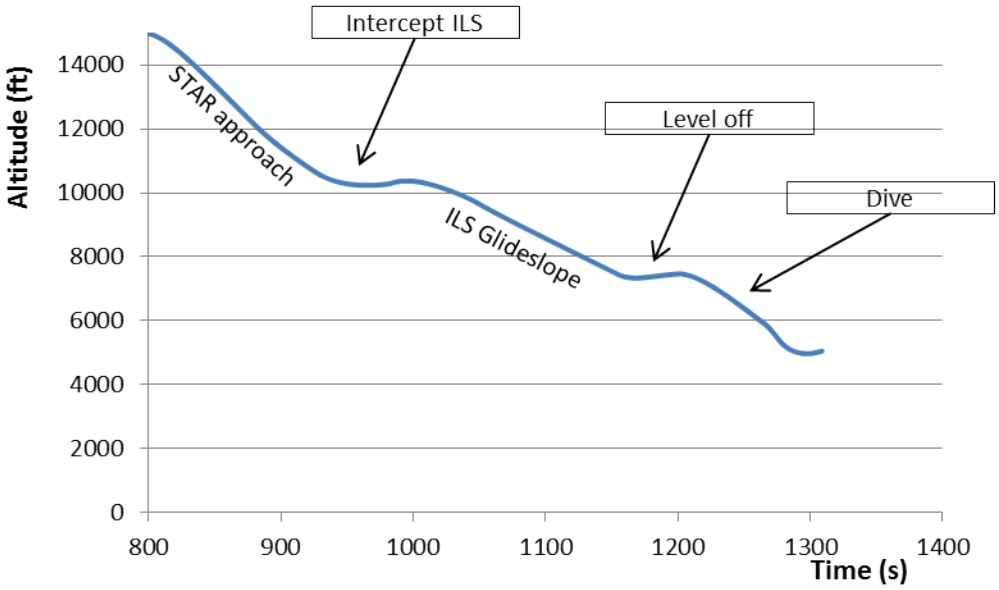

Figure 31: Exemplary altitude profile of the aircraft in Scenario 6 with altitudes in which the events occur. The scenario variables of this example are Go Around altitude: 6000; Flaps Extension Speed: 205; Level Off Altitude: 8000; Level Off Duration: 70.

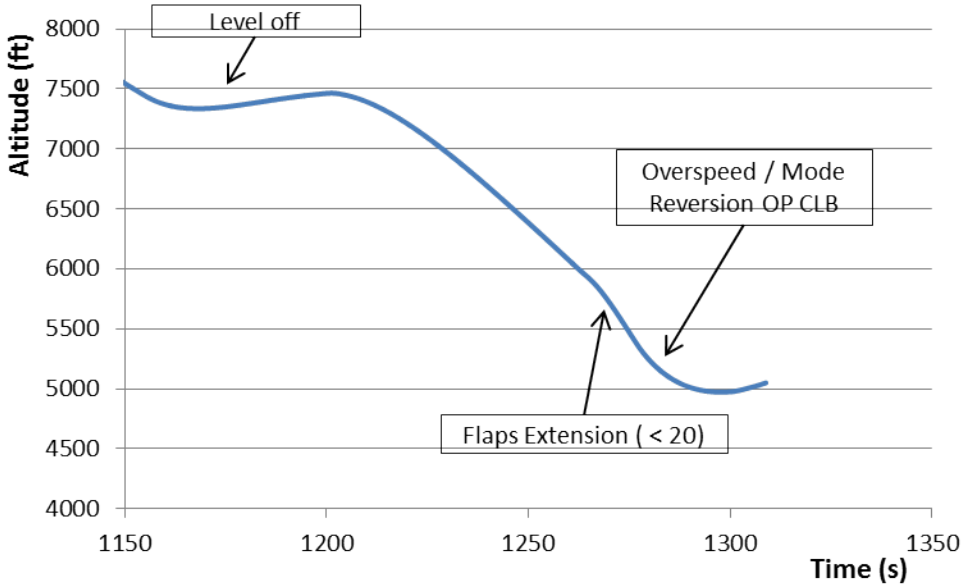

Figure 32: Closeup of the altitude profile of Scenario 6 in Figure 31 and the respective times and altitudes at which events and actions occurred. The level off occurs at about 7500 feet after being commanded at 8000 feet. The aircraft dives and some time later later flaps are extended to a lower configuration than 20 degrees, in this example to 10 degrees. The dive leads the aircraft to overspeed and the mode reversion to OPEN CLB and the aircraft starts climbing.

5.5 Analysis

5.5.1 SAL Equivalent Scenario

As described previously, simulation runs were performed that do not match the SAL scenario completely due to the use of a Boeing aircraft model and the way the waypoints are defined. At 3291 feet go around altitude, as given in the SAL scenario, the only automation surprise linked to the OPEN CLB that occurs is Scenario 6, where the pilot extends the flaps at the correct speed and the dive leads to the overspeed and mode reversion. However, in Scenario 6 the flaps extension is to a setting still below 20 degrees because the aircraft has not yet reached the speed at which flaps extension to 20 degrees is commanded. An automation surprise as it does occur in SAL, due to early flaps extension, does not appear at a go around altitude of 3291 feet and flaps extension speed of 226 knots because of the aircraft model used in WMC and the waypoint target speeds.

Nevertheless, it has been shown that if the preconditions provided by SAL are modified in WMC partially, to be more appropriate for Boeing 747 dynamics, then indeed an automation surprise in such a form appears. If the precondition that the go around altitude is 3291 feet is relaxed and the altitude is set higher, then the same automation surprise occurs. At a go around altitude of 4000 feet and below no automation surprise due to early flaps extension involving the OPEN CLB mode occurs. Scenario 6 is the exception, which is not caused by early flaps extension.

Figure 34 shows an example where the same automation surprise as in SAL is shown that appears at a go around altitude of 6000 feet and not 3291 feet as in SAL. In the scenario the aircraft levels off at 5000 feet and not at 3000 feet as in SAL. This fulfills the precondition given in SAL that the aircraft resumes descent after leveling off. Then the aircraft descends and flaps are extended early, which leads to the overspeed and the mode reversion to OPEN CLB. At the level off the expectation of the pilot is level, then when the aircraft descends it changes to descent and stays

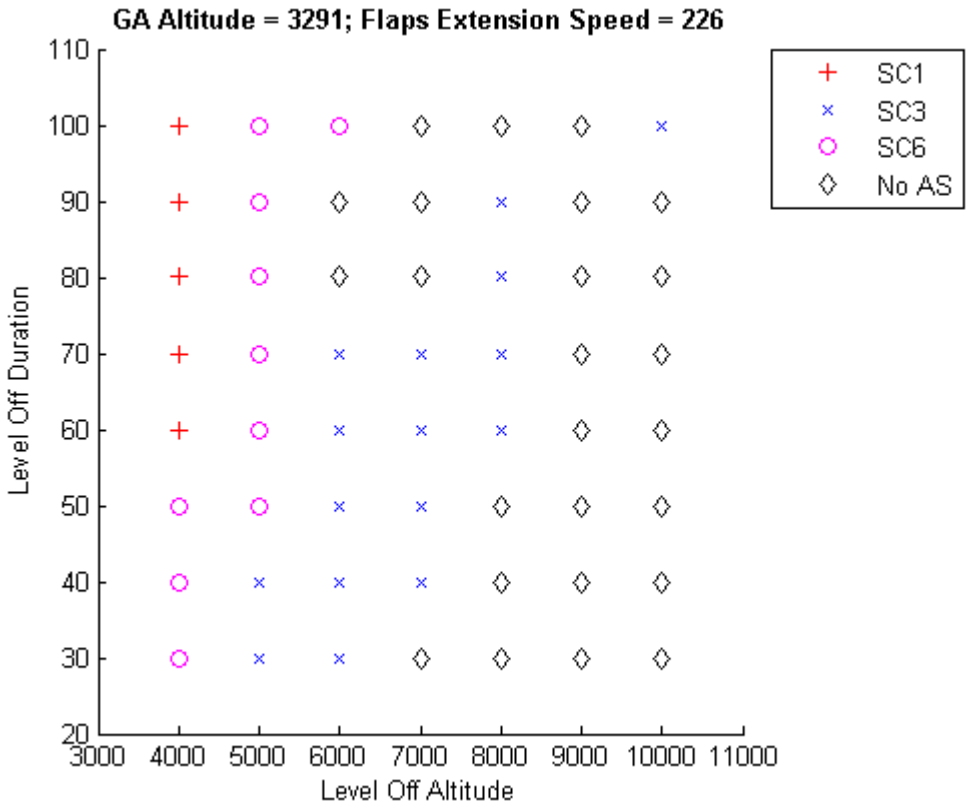

Figure 33: Scenarios that appeared under the preconditions given in SAL: Go Around Altitude of 3291 feet and Flaps Extension Speed 21 knots above the maximum speed (226 knots).

this way when the aircraft starts climbing which leads to an automation surprise flag.

5.5.2 Next Iteration: Modelcheck WMC Scenarios

Using the model and scenario given by SAL produces six scenarios in WMC that involve the automatic speed protection and are of interest. Of these six, four scenarios elicit an actual automation surprise that could occur in reality. As explained in the methods section in 4.1 and as is shown in Figure 35 the next step after obtaining the WMC results is to use these to make a more elaborate SAL model and to initiate a new iteration. Conceptually, the results from simulation can be used in two ways to improve model checking: 1) extending the models to include additional dynamics, 2) refining the model to be more precise and to include additional states.

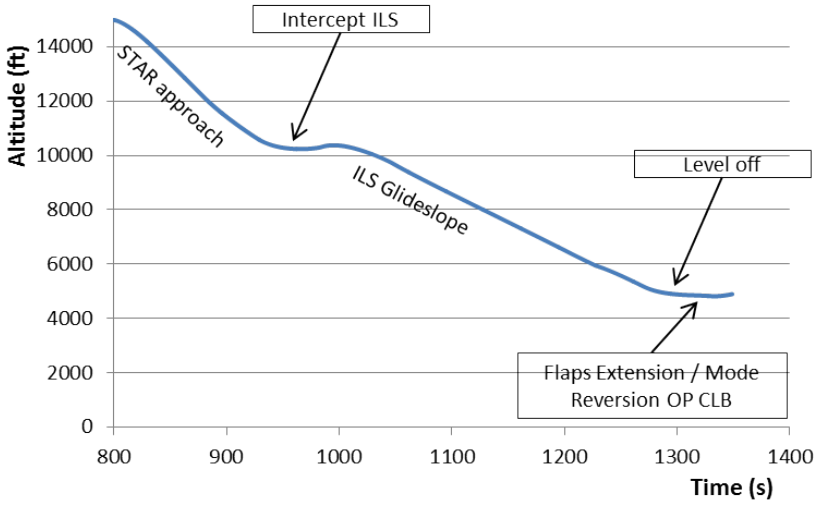

Figure 34: Altitude profile of the SAL scenario shifted to a higher altitude and the actions and events in the scenario.

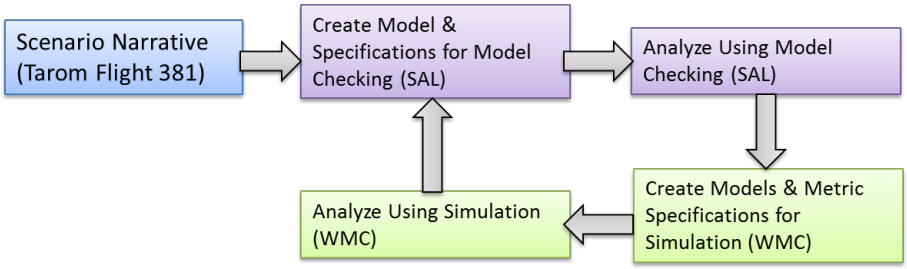

Figure 35: Iterative method to connect WMC and SAL.

There are several options for how to make the SAL model more extensive by including the findings of WMC. First, WMC has shown that there is more than one way for an automation surprise to appear that involves the OPEN CLB mode, this comprises scenario 2, 4 and 6. Second, WMC has shown that in addition to surprises caused by OPEN CLB, that automation surprises also involve the OPEN DES, such as in Scenario 3.

Another way that SAL can take WMC's findings into account is if WMC shows that the scenario appears under different conditions than the ones given in SAL. Since WMC replicated the SAL scenario by changing the assumptions of the aircraft model and the waypoints, there is no refinement in this direction necessary or possible.

Furthermore, the SAL model was constructed by abstracting real world data, which proves that an automation surprise can occur in reality. However, for other scenarios which are not based on a known (and fixed) cockpit problem, WMC can provide more realistic altitudes and speeds at which events occur.

CHAPTER VI

CONCLUSION

6.1 Summary

The goal of this thesis is to provide a method to use model checking (SAL) and simulation (WMC) in conjunction to explore potentially problematic aircraft system behavior. The method is demonstrated and validated through the use of a common scenario case study. In the case study an example of problematic aircraft system behavior called automation surprise is examined. In this era of extraordinary flight safety levels at which proactive rather than reactive approaches to accidents are needed, it is necessary to be able to examine aircraft system behavior and identify potential problems between pilots and automation before incidents linked to a problematic interaction arise.

The common scenario examined here as a case study is automation surprise resulting from the former Airbus overspeed protection flight logic. Automation surprises appear when the expectations of the pilot differ from the real state of the aircraft and / or the impact of the automation on the dynamic states of the aircraft. In the past it has been successfully shown that SAL can be used to formally prove an automation surprise can occur and reveal the sequence of precipitating events. The scenario modeled a single aircraft state comparing aircraft pitch and pilot's expectation about the pitch. In the model, when the speed protection logic was activated a discrepancy between reality and pilot expectation was shown to occur, eliciting an automation surprise. In this work it has been shown that the same can be achieved with a more detailed model simulation, using WMC. While SAL does not explicitly model time

nor includes flight dynamics, WMC is able to mitigate these shortcomings by providing time and realistic position and velocity data about the aircraft when events occur.

The benefits of using two such distinct methods (simulation and model checking) and frameworks (WMC and SAL) is particularly useful to mitigate the shortcomings of both methods separately and achieving synergy by leveraging the strengths of the two methods. Such a combination of methods yields: sophisticated models, including agent models and high-fidelity aircraft models, and the explicit modeling of time. These strengths are provided by WMC. On the other hand, SAL provides the possibility to exhaustively explore the state-space and uncover potential issues that may not be found by simulation on its own, since simulations are not able to explore the whole state-space. Such a combination of methods provides a deeper scrutiny for the exploration of all aspects of human-automation interaction and can therefore uncover potential problems in this interaction which other methods may not be able to uncover.

In conclusion, the study provides evidence that model checking and simulation can be used together to examine system behavior. Model checking provides a useful starting point in the state-space for simulation models to be initialized at. Simulation provides a much more detailed and realistic model to determine the accuracy of the model checking abstraction, to identify additional safety concerns in slightly different initial conditions or action sequences. In the future, the two modeling approaches can be used in conjunction to uncover potential problems in system behavior and provide conditions before issues can arise in real flight.

6.2 Contributions

This work provides a method how to use simulation and model checking in conjunction to look at potentially problematic aspects of system behavior. This includes

newly introduced automation functionality or changes in the authority and autonomy assignment of tasks between controllers and pilots.

Such a method has its advantages since it is not required to actually use expensive flight simulator time with pilots or perform flight tests to look at particular aspects of the system and the human-automation interaction. These instruments are only required when it is very likely that the new system can cause problems between automation and human. Likely problems can be uncovered with the use of the method presented in this work.

Also, this work provides a definition of mental models describing it with beliefs and expectations, and how the discrepancy of these two elicit an automation surprise for the pilot. Such a definition of mental models in the context of automation surprises, the beliefs of the human do not match the expectations, has not been defined previously in the context of simulating automation surprises in the aviation domain. Furthermore, the beliefs and expectations construct in WMC can be used for other work to be able to identify problems between the pilot's expectation about the state of the environment and the real state of the environment.

6.3 Future Work

A SAL model has been implemented and simulated in WMC. The method that is proposed involves iterations between SAL and WMC (see Figure 36). In SAL the first basic model is defined and then the same preconditions eliciting an interesting scenario, such as an automation surprise, are fed into WMC. In the next iteration step the models and results obtained from WMC are taken to make the SAL models more complex. These SAL models may led to other realistic and interesting scenarios. In the context of the case study presented, this means implementing the necessary models for the different conditions under which an automation surprise involving the OPEN CLB mode can occur. Also, potential automation surprises linked to OPEN

DES mode can be implemented and checked in SAL, since the SAL model lacks this mode transition. This is necessary since WMC suggests that there may be automation surprises linked to the OPEN DES mode. Since, this case study is taken from a real world incident, there is no more detail than can be provided than the actual incident scenario narrative. In case studies in which conditions are not provided by real world data, WMC can provide realistic temporal conditions and altitudes of problems.

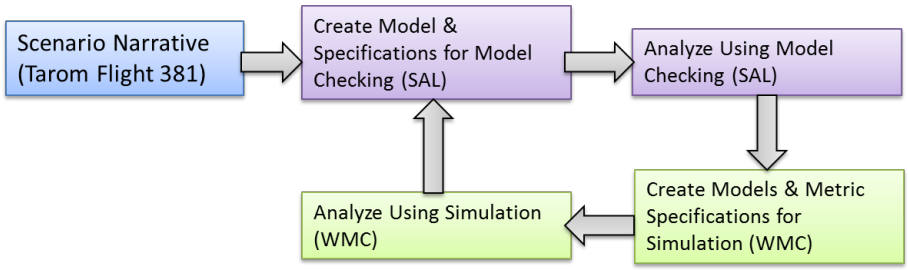

Figure 36: Iterative method to combine SAL and WMC.

The conditions given in the common case study previously modeled in SAL, have been implemented manually into WMC. Future methods need to translate between WMC and SAL in a more automatic fashion. Current work focuses on methods how to facilitate translation between these two platforms.

The goal of this method is to uncover problematic system behavior when implementing new cockpit automation or new procedures. It is necessary to make this method suitable for cockpit designers that do not have knowledge of SAL or WMC and can make sure that their system does not exhibit potentially dangerous Human-Automation Interaction.

APPENDIX A

ANCILLARY

A.1 Rules for Expecations

- Altitude: MCP selected altitude. Expectation is that the altitude will approach the MCP selected altitude.
- Airspeed: No integration yet.
- Vertical Speed: MCP selected altitude or selected vertical speed. If MCP selected altitude is below current altitude, expected vertical speed is negative and vice versa. If vertical speed is selected in MCP the pilots expect a vertical speed similar to it.
- Heading: Active waypoint latitude and longitude. Calculate expected heading from the latitude and longitude of the next waypoint.

A.2 Automation Surprise Algorithm

A.2.1 Algorithms to transform mental model to fit expectations

Vertical Speed (value computed from current value):

IF absolute(Mental_Resource_Value) <absolute(Threshold)

THEN Mental_Resource_Converted = 0;

ELSE Mental_Resource_Converted = Mental_Resource_Value / absolute (Mental_Resource_Value)

Others (value computed from past ten values to extract the trend):

1. Add all 10 Mental_Resource_Value and subtract 10 times the lowest value = Transformed_Mental_Resource_Value

2. Same procedure as with vertical speed

IF absolute(Transformed_Mental_Resource_Value) <absolute(Threshold)

THEN Mental_Resource_Converted = 0;

ELSE Mental_Resource_Converted = Transformed_Mental_Resource_Value / absolute (Transformed_Mental_Resource_Value)

The resulting value is -1, 0, 1 and can be compared to the expectations.

A.2.2 Thresholds

- Altitude: ±100 feet
- Airspeed: No integration yet.
- Vertical Speed: ±5 ft/s
- Heading: No definitive thresholds yet.

A.3 Description of Flight Modes

ACN: 268726

Time / Day

Date : 199404
Local Time Of Day : 0601-1200

Place

Locale Reference.Airport : MSP
State Reference : MN
Relative Position.Angle.Radial : 110
Relative Position.Distance.Nautical Miles : 15
Altitude.MSL.Single Value : 4000

Environment

Flight Conditions : VMC
Weather Elements / Visibility.Visibility : 20
Light : Daylight

Aircraft

ATC / Advisory.TRACON : MSP
Aircraft Operator : Air Carrier
Make Model Name : A320
Crew Size.Number Of Crew : 2
Operating Under FAR Part : Part 121
Flight Plan : IFR
Mission : Passenger
Flight Phase : Cruise
Flight Phase : Initial Approach
Route In Use : Vectors
Airspace.Class B : MSP

Component : 1

Aircraft Component : Unknown
Aircraft Reference : X
Problem : Design

Component : 2

Aircraft Component : FMS/FMC
Aircraft Reference : X
Problem : Design

Component : 3

Aircraft Component : Vertical Speed System
Aircraft Reference : X
Problem : Design

Component : 4

Aircraft Component : Unknown
Aircraft Reference : X
Problem : Design

Person : 1

Reference : 1
Reporter Organization : Air Carrier
Function.Flight Crew : Pilot Flying
Function.Flight Crew : First Officer
Qualification.Flight Crew : Air Transport Pilot (ATP)
Qualification.Flight Crew : Flight Engineer
Qualification.Flight Crew : Flight Instructor
Experience.Flight Crew.Total : 11500
Experience.Flight Crew.Last 90 Days : 150
Experience.Flight Crew.Type : 150
ASRS Report Number.Accession Number : 268726

Person : 2

Reference : 2
Reporter Organization : Air Carrier
Function.Flight Crew : First Officer
Qualification.Flight Crew : Air Transport Pilot (ATP)

Person : 3

Reference : 3
Reporter Organization : Government
Function.Air Traffic Control : Approach
Qualification.Air Traffic Control : Fully Certified

Events

Anomaly.Aircraft Equipment Problem : Less Severe
Anomaly.Deviation - Altitude : Excursion From Assigned Altitude
Anomaly.Deviation - Procedural : Clearance
Anomaly.Inflight Event / Encounter : Loss Of Aircraft Control
Anomaly.Other
Detector.Person : Flight Crew
Result.Flight Crew : Regained Aircraft Control
Result.Flight Crew : Overcame Equipment Problem

Assessments

Primary Problem : Aircraft

Narrative: 1

WHILE ON A HDG TO INTERCEPT THE ILS 29R AT MSP WE CALLED THE ARPT IN SIGHT AND WE WERE CLRED FOR THE VISUAL APCH. AT THIS TIME WE WERE LEVEL AT AN ASSIGNED ALT OF 4000 FT. PRIOR TO LOC INTERCEPT, THE GS CAME INTO VIEW AND I ARMED THE APCH FUNCTION OF THE AUTOPLT. (THE LATEST VERSION OF SOFTWARE ALLOWS GS CAPTURE PRIOR TO LOC INTERCEPT.) THE GS CAPTURED AND WE BEGAN A DSCNT FOR A COUPLE HUNDRED FT. FOR AN UNKNOWN REASON (POSSIBLY A FALSE GS) WE BEGAN TO CLB, WHILE STILL IN GS CAPTURE MODE. THIS TRANSITION TO A CLB WAS NOT NOTICED IMMEDIATELY DUE TO CONFIGN CHANGES, TURBULENT CONDITIONS AND A PWR

INCREASE THAT WAS NOT OBVIOUS, SINCE THE THRUST LEVERS DO NOT MOVE. I BECAME AWARE OF THE CLB AT ABOUT 4100 FT AND ATTEMPTED TO LEVEL OFF BY PULLING THE VERT SPD KNOB AND ROTATING IT TO A DSCNT. THE ACFT CONTINUED TO CLB ON THE AUTOPLT, WHICH I THEN DISCONNECTED. BY THIS TIME WE HAD BALLOONED THROUGH 4350 FT. I WAS SURPRISED TO FIND THAT THE VERT SPD SELECTOR WINDOW SHOWING A CLB. I DON'T KNOW WHY THIS WAS THE CASE -- POSSIBLY BECAUSE OF PULLING AND TURNING THE KNOB AT THE SAME TIME, AND THE SYS ACCEPTED THE ACFT CURRENT VERT SPD RATHER THAN THE REQUESTED VERT SPD. THERE IS NO ALT HOLD BUTTON. NO CONFLICT OCCURRED. THERE IS A LOT TO DISLIKE ABOUT THE DESIGN AND IMPLEMENTATION OF THIS AUTOMATED COCKPIT.

Synopsis

ALTDEV ALT EXCURSION DURING APCH ON AUTOPLT.

ACN: 262473

Time / Day

Date : 199401
Local Time Of Day : 0601-1200

Place

Locale Reference.Airport : CPH
State Reference : FO
Relative Position.Distance.Nautical Miles : 6
Altitude.MSL.Single Value : 2000

Environment

Flight Conditions : VMC
Weather Elements / Visibility.Visibility : 10
Light : Daylight
Ceiling.Single Value : 3000

Aircraft

ATC / Advisory.TRACON : CPH
Aircraft Operator : Air Carrier
Make Model Name : A310
Crew Size.Number Of Crew : 2
Flight Plan : IFR
Mission : Passenger
Flight Phase : Landing
Flight Phase : Initial Approach
Route In Use.Airway : CPH
Route In Use.Other

Person : 1

Reference : 1
Reporter Organization : Air Carrier
Function.Flight Crew : Captain
Qualification.Flight Crew : Air Transport Pilot (ATP)
Experience.Flight Crew.Total : 17000
Experience.Flight Crew.Last 90 Days : 175
Experience.Flight Crew.Type : 1500
ASRS Report Number.Accession Number : 262473
Analyst Callback : Attempted

Person : 2

Reference : 2
Reporter Organization : Air Carrier
Function.Flight Crew : First Officer
Function.Flight Crew : Pilot Flying
Qualification.Flight Crew : Air Transport Pilot (ATP)

Person : 3

Reference : 3
Reporter Organization : Air Carrier
Function.Flight Crew : First Officer
Qualification.Flight Crew : Air Transport Pilot (ATP)

Person : 4

Reference : 4
Reporter Organization : Air Carrier
Qualification.Other

Events

Anomaly.Aircraft Equipment Problem : Critical
Anomaly.Inflight Event / Encounter : Loss Of Aircraft Control
Detector.Person : Flight Crew
Result.Flight Crew : Executed Go Around / Missed Approach
Result.Flight Crew : Overcame Equipment Problem

Assessments

Primary Problem : Aircraft

Narrative: 1

AT ABOUT 2000 FT ON ILS APCH, WE EXPERIENCED AN UNSCHEDULED PITCH TRIM EVENT, WHICH RESULTED IN A PITCH UP ALT OF MORE THAN 20 DEGS WITH AUTO THROTTLES GOING TO GAR THRUST. BY THE TIME WE RECOVERED, WE WERE AT 4000 FT, BUT MISSED APCH ALT WAS 3000 FT. I DO NOT KNOW IF THERE WERE ANY TFC CONFLICTS. THIS WAS THE SECOND LEG OF A TRANSOCEANIC TRIP AND WE WERE ALL TIRED. THE TRIM IS SUPPOSED TO GIVE AN AURAL WARNING IF IT RUNS AWAY OR IF USED FOR MORE THAN 1 SECOND. NO ONE HEARD THIS, BUT WE SAW THE THROTTLES ADVANCING, WHICH CONTRIBUTED TO BOTH PITCH UP AND MOMENTARY CONFUSION ABOUT WHAT CAUSED THE PITCH UP. WE BELIEVE IT WAS CAUSED A STICKING TRIM SWITCH BUT ARE NOT SURE. MAINT FOUND NOTHING WRONG. PERHAPS THE ADDITION OF A WARNING LIGHT IN ADDITION TO THE AURAL WARNING (WHICH IS HARD TO HEAR ABOVE AMBIENT COCKPIT NOISE AND RADIO XMISSIONS) WOULD HAVE HELPED US TO RECOGNIZE THE SOURCE OF THE PROB MORE PROMPTLY WHEN VERY TIRED. AFTER AN ALL NIGHT OCEAN XING WE ALL NEED ALL THE HELP WE CAN GET. CALLBACK CONVERSATION WITH RPTR REVEALED THE FOLLOWING INFO: THE RPTR WAS FLYING AN AIRBUS A 310 FOR A U.S. BASED ACR. THE ACFT WAS BEING FLOWN MANUALLY BY THE FO WITH AUTO THROTTLE ENGAGED. THE AUTO THROTTLES WENT TWICE TO 'FULL HERMAN' FOR NO APPARENT REASON AND THE ELEVATOR TRIM WAS BEING DRIVEN TOWARDS FULL NOSE UP WITH NO COMMAND FROM EITHER PLT. THE RPTING CAPT RETARDED THE THROTTLES TWICE, THEN HIT HIS TRIM SWITCH FORWARD TO NOSE DOWN. AS THE ENGS ARE SLUNG UNDER THE WINGS, THE NOSE UP PITCH WAS ACCENTUATED BY THE ABRUPT THROTTLE MOVEMENT. ACR MAINT AT CPH WAS UNABLE TO FIND ANY PROB. ACR MAINT AT THE CREW'S HOME BASE CHANGED THE FO'S TRIM SWITCH THINKING THAT IT MIGHT HAVE MOMENTARILY STUCK IN THE NOSE UP POS. THE RPTR KNOWS OF NO OTHER INCIDENTS OF THIS TYPE, BUT THERE HAVE BEEN UNWANTED NOSE UP PITCHES WHEN THIS TYPE ACFT HAS BEEN MAKING A COUPLED APCH. THE CREW HAD NO FURTHER PROB CTLLING THE ACFT ON THE NEXT APCH AND LNDG.

Synopsis

AN ACR A310 CLBED THROUGH THE MISSED APCH ALT.

REFERENCES

[1] Bass, E. J., Feigh, K. M., Gunter, E., and Rushby, J., "Formal modeling and analysis for interactive hybrid systems," in *4th International Workshop on Formal Methods for Interactive Systems*, (Limerick, Ireland), June 2011.

[2] BEA, "Report on the incident on 24 september 1994 during approach to orly (94) to the airbus a 310 registered yr-lca operated by tarom," September 1994.

[3] Blom, H., Bakker, G., Blanker, P., Daams, J., Everdij, M., and Klompstra, M., "Accident risk assessment for advanced air traffic management," *Progress in Astronautics and Aeronautics*, vol. 193, pp. 463–480, 2001.

[4] Blom, H. A., Corker, K. M., and Stroeve, S., "On the integration of human performance and collision risk simulation models of runway operation," in *Proceedings of the 6th USA/Europe Air Traffic Management R&D Seminar*, pp. 27–30, 2005.

[5] Butler, R., Miller, S., Potts, J., and Carreno, V., "A formal methods approach to the analysis of mode confusion," in *Digital Avionics Systems Conference, 1998. Proceedings., 17th DASC. The AIAA/IEEE/SAE*, vol. 1, pp. C41–1, IEEE, 1998.

[6] Clarke, E., Grumberg, O., Jha, S., Lu, Y., and Veith, H., "Counterexample-guided abstraction refinement," in *Computer aided verification*, pp. 154–169, Springer, 2000.

[7] Corker, K. M. and Smith, B. R., "An architecture and model for cognitive engineering simulation analysis: Application to advanced aciation automation," in *AIAA Computing in Aerospace 9 Conference*, 1993.

[8] Crow, J., Javaux, D., and Rushby, J., "Models and mechanized methods that integrate human factors into automation design," in *Proceedings of HCI-Aero 2000: International Conference on Human-Computer Interaction in Aeronautics*, p. 163168, 2000.

[9] Doyle, J., Radzicki, M., and Trees, W., "Measuring change in mental models of complex dynamic systems," *Complex Decision Making*, pp. 269–294, 2008.

[10] Doyle, J. K. and Ford, D. N., "Mental models concepts for systems dynamics research," *System Dynamics Review*, vol. 14 (1), pp. 3–29, 1998.

[11] Javaux, D., "Explaining Sarter and Woods classical results. the cognitive complexity of pilot-autopilot interaction on the Boeing 737-EIS," in *Proceedings of Human Error, Safety, and System Development (HESSD98)*, (Seattle, Washington), April 1-2 1998.

[12] Javaux, D. and Polson, P. G., "A method for predicting errors when interacting with finite state machines: The impact of implicit learning on the user's model of the system.," in *Pre-Proceedings of Human Error, Safety, and System Development (HESSD '99)*, 1999.

[13] Kalambi, V., Pritchett, A., Bruneau, D., Endsley, M., and Kaber, D., "In-flight planning and intelligent pilot aids for emergencies and non-nominal flight conditions using automatically generated flight plans," in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 51, pp. 55–59, SAGE Publications, 2007.

[14] Leveson, N. G., Pinnel, L. D., Sandys, S. D., Koga, S., and Rees, J. D., "Analyzing software specifications for mode confusion potential," in *Proceedings of a Workshop on Human Error and System Development* (Johnson, C. W., ed.), (Glasgow, Scotland), pp. 132–146, Glasgow Accident Analysis Group, technical report GAAG-TR-97-2, Mar. 1997. Paper available at http://www.cs.washington.edu/research/projects/safety/www/papers/glascow.ps (sic).

[15] Norman, D., "Some observations on mental models," *Mental models*, vol. 7, 1983.

[16] Palmer, E., "Ooops, It Didnt Arm. A case study of two automation surprises," in *Proceedings of the Eighth International Symposium on Aviation Psychology*, (Columbus, OH), pp. 227–232, April 1995.

[17] Pritchett, A., Kim, S. Y., Kannan, S. K., and Feigh, K. M., "Simulating situated work," in *IEEE International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA)*, pp. 66–73, IEEE, 2011.

[18] Pritchett, A. R. and Feigh, K. M., "Simulating first-principles models of situated human performance," in *IEEE International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA)*, (Miami Beach, FL), IEEE, 2011.

[19] Romera, M., *Using finite automata to represent mental models.* PhD thesis, San Jose State University, 2000.

[20] Rushby, J., "Analyzing cockpit interfaces using formal methods," in *Proceedings of FM-Elsewhere* (Bowman, H., ed.), vol. 43 of *Electronic Notes in Theoretical Computer Science*, (Pisa, Italy), Elsevier, Oct. 2000.

[21] Rushby, J., "Using model checking to help discover mode confusions and other automation surprises," *Reliability Engineering and System Safety*, vol. 75, pp. 167–177, Feb. 2002.

[22] Rushby, J., "The needham-schroeder protocol in sal," *CSL Technical Note*, 2003.

[23] Rushby, J., "Sal tutorial: Analyzing the fault-tolerant algorithm om (1)," tech. rep., Technical Report CSL Technical Note, SRI International, 2004. Available at http://www. csl. sri. com/users/rushby/abstracts/om1, 2004.

[24] Sarter, N. B. and Woods, D. D., "Pilot interaction with cockpit automation: Operational experiences with the flight management system," *The International Journal of Aviation Psychology*, vol. 2, no. 4, pp. 303–321, 1992.

[25] Sarter, N. and Woods, D., "Pilot interaction with cockpit automation ii: An experimental study of pilots model and awareness of the flight management system," *The International Journal of Aviation Psychology*, vol. 4, no. 1, pp. 1–28, 1994. Hard copy.

[26] Sarter, N. and Woods, D., "Team play with a powerful and independent agent: Operational experiences and automation surprises on the airbus a-320," *Human Factors*, vol. 39, no. 4, 1997.

[27] Sarter, N., Woods, D., and Billings, C., "Automation surprises," *Handbook of human factors and ergonomics*, vol. 2, pp. 1926–1943, 1997.

[28] Stroeve, S., Blom, H., and Van der Park, M., "Multi-agent situation awareness error evolution in accident risk modelling," in *Proceedings of the 5th USA/Europe Seminar on Air Traffic Management Research and Development*, Citeseer, 2003.

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

[30] Wiener, E., "Human factors of advanced technology (glass cockpit) transport aircraft," 1989.

[31] Wiener, E. L. and Curry, R. E., "Flight-deck automation: promises and problems," *Ergonomics*, vol. 23, no. 10, pp. 995–1011, 1980.