

Agent-Based Modelling of an Airport's Ground Surface Movement Operation

T.E.H. Noortman

Agent-Based Modelling of an Airport's Ground Surface Movement Operation

Understanding the principles and mechanisms of decentralised control

by

T.E.H. Noortman

to obtain the degree of Master of Science in Aerospace Engineering
at Delft University of Technology,
to be defended publicly on May 25, 2018.

Student number: 4211766
Project duration: September 2016 – May 2018
Thesis Committee: Prof. Dr. R. Curran, TU Delft, ATO, chair
Dr. O.A. Sharpanskykh, TU Delft, ATO, supervisor
Ir. M. Tielrooij, To70, supervisor
Ir. O. Stroosma, TU Delft, C&S, external examiner

An electronic version of this dissertation is available at <http://repository.tudelft.nl/>.



Preface

Not only am I writing these final words to conclude my master's thesis, they are also the words that end a chapter of my life. Over the past five and a half years, Delft University of technology has given me the opportunity to develop both the engineering and personal skills that will act as the foundation of my future steps in life. I would like to thank everyone who supported me along this journey.

First of all, I would like to thank my supervisor Dr. Alexei Sharpanskykh, who introduced me to the intriguing concept of Agent-Based Modelling. His door was always open for some interesting discussions regarding the thesis, or the promising future of Agent-Based Modelling. I also would like to thank my supervisor ir. Maarten Tielrooij, whose research mindset and operational knowledge have been of great help for the research approach that I have taken. I am grateful for their advice and research guidance.

Another thanks goes to Heiko and Leo, who have developed the Open Source Simulator for ATM Research (OSSAR) tool that acted as a basis for this research. They were always willing to share insights regarding the implementation, and think along with further expansions of the model.

I would like to thank To70 for their collaboration. All colleagues in the office were pleased to share their operational expertise, and their input has been very valuable.

Over the past few years, I have got to know many great people. I am also blessed to have met Jan on my first day in Delft. No matter whether he was nearby or abroad, he found the time to have great conversations and to fantasise about future opportunities in life. I also would like to thank Marta, who was always available for a coffee or a chat, to share some thoughts and to set my mind at rest.

My gratitude goes towards my parents and sister, for their never ending support and trust. They had my back, and gave me that extra push when I needed one.

And above all, a special thanks to my girlfriend Melissa, who has always been there to cheer me up. Since the day I have met her, she stood by my side, encouraging me to fulfil my ambitions. Her endless support has given me the feeling that I did not make this journey alone.

*T.E.H. Noortman
Delft, May 2018*

List of Tables

2.1	Minimum separation distance between aircraft	6
5.1	Specification of the general parameters	54
5.2	Specification of the aircraft dynamics	54
5.3	Specification of the Air Traffic Control agents	54
6.1	Characteristics of selected days	56
6.2	Statistical analysis of the simulated operation	60
6.3	Sample Entropy value per day	68
6.4	Values of operational parameters as tested by the sensitivity analysis	71
6.5	Values of the aircraft dynamics as tested by sensitivity analysis	72
6.6	Speed profiles for sensitivity analysis	73
D.1	Create dummy graph from original graph	107
E.1	Results sensitivity analysis of operational parameters on taxi time	113
E.2	Results sensitivity analysis of operational parameters on taxi distance	114
E.3	Results sensitivity analysis of operational parameters on average taxi speed	114
E.4	Results sensitivity analysis of operational parameters on average density	114
E.5	Results Spearman's Rank Correlation of operational parameters on taxi time	114
E.6	Results Spearman's Rank Correlation of operational parameters on taxi distance	115
E.7	Results Spearman's Rank Correlation of operational parameters on average taxi speed	115
E.8	Results Spearman's Rank Correlation of operational parameters on average density	115
E.1	Results sensitivity analysis of aircraft dynamics on taxi time	118
E.2	Results sensitivity analysis of aircraft dynamics on taxi distance	118
E.3	Results sensitivity analysis of aircraft dynamics on average taxi speed	119
E.4	Results sensitivity analysis of aircraft dynamics on average density	119
E.5	Results Spearman's Rank Correlation of aircraft dynamics on taxi time	120
E.6	Results Spearman's Rank Correlation of aircraft dynamics on taxi distance	120
E.7	Results Spearman's Rank Correlation of aircraft dynamics on average taxi speed	120
E.8	Results Spearman's Rank Correlation of aircraft dynamics on average density	120

List of Figures

2.1	Layout of Amsterdam Airport Schiphol	4
2.2	Important taxiways of Amsterdam Airport Schiphol	4
2.3	Work breakdown of the ground operation	5
2.4	Possible types of taxiway conflicts	6
2.5	Diagram of interactions between the agents involved in a ground operation	10
3.1	Research framework	19
3.2	Iterative process to develop the simulation model	19
4.1	Effect of replacing an apron by an entry/exit node	22
4.2	Effect of removing nodes in between a long segment and grouping complex corners to one node	23
4.3	Effect of merging a cluster of nodes to one node	23
4.4	Effect of simplifying the airport layout	23
4.5	Links between database tables	25
4.6	Heat map based on ADS-B data points	25
4.7	Steps required to filter track data	27
4.8	Steps required to clean and process track data	27
4.9	Airport boundary	27
4.10	Merged arriving segment and ground segment	28
4.11	Points outside the airport boundary	28
4.12	Corrections made to path of flight 1	29
4.13	Corrections made to path of flight 2	29
4.14	Corrections made to path of flight 3	30
4.15	Strange behaviour of flights	32
4.16	Restrictions of a simplified layout	32
5.1	Decentralised Air Traffic Control	35
5.2	Structure of Air Traffic Control agents	36
5.3	Decision making process of a source agent	36
5.4	Decision making process of a runway agent	37
5.5	Decision making process of an intersection agent	38
5.6	Decision making process of a stopbar agent part I	39
5.7	Decision making process of an intersection agent part II	39
5.8	Example of relation between Air Traffic Control agents	40
5.9	Model interface to closely monitor the simulated and actual operation	42
5.10	Procedure to estimate the future usage of a runway	44
5.11	Location of stopbars surrounding runway 18C – 36C	46
5.12	Example of operation involving short segments	49
5.13	Auction process around surrounding runway 18C – 36C	49
5.14	Sharp turns that are present in the airport's taxiway layout	51
5.15	Aircraft willing to use each other's link	51
6.1	Box plot of taxi time	59
6.2	Box plot of taxi distance	59
6.3	Box plot of average taxi speed	59
6.4	Box plot of average density	59
6.5	Travelling along a bypass, instead of taking the shortest route	60
6.6	Most frequently used routes part 1	61
6.7	Most frequently used routes part 2	62

6.8	Taxi time and taxi distance plot of 9 th of May	63
6.9	Taxi time and taxi distance plot of 3 rd of May	63
6.10	Bad runway queue due to missing global goal	64
6.11	Unknown insights into future operation	64
6.12	Loop in aircraft's ground path	65
6.13	Layout used for complexity measure Sample Entropy	66
6.14	Rolling hour of Sample Entropy on 9 th of May	67
6.15	Rolling hour of Sample Entropy on 1 st of May	67
6.16	Rolling hour of Sample Entropy on 2 nd of May	68
6.17	Deviation from procedures of Air Traffic Control	68
6.18	Location of gridlocks	73
6.19	Operational gridlock caused by insufficient information of aircraft positions	74
6.20	Operational gridlock caused by missing forced outpath	75
6.21	Adding extra layer of auctions	77
6.22	Closest point of approach	78
6.23	Separation between slow moving vehicles	79
A.1	Process to determine the closest segment	92
A.2	Procedure to shift data points with similar time stamps	93
A.3	Procedure to remove data points when going forward in time but backwards in distance	93
B.1	Decentralised Air Traffic Control	96
B.2	Initial airport layout	96
B.3	Decision making process of an aircraft agent's acceleration	97
B.4	Decision making process of an intersection agent's hand off	98
B.5	Diagram with interaction between agents	99
C.1	Establishing of the link_times dictionary	104
D.1	Concept of the dummy graph	106
D.2	Route update requiring aircraft to turn around at the apron	110
D.3	Original weights per dummy segment	111
D.4	Updated weights per dummy segment	111
D.5	Complexities to determine the weight of a dummy segment	111
D.6	Aircraft speed profile along a 100 metres long segment that requires two turns	111

List of Acronyms

AAS Amsterdam Airport Schiphol.

ABM Agent-Based Modelling.

ACDM Airport Collaborative Decision Management.

ADS-B Automatic Dependent Surveillance-Broadcast.

AFTM Air Traffic Flow Management.

ApEn Approximate Entropy.

ATC Air Traffic Control.

ATM Air Traffic Management.

CPDSP Collaborative Pre-Departure Sequence Planning.

CTOT Calculated Take Off Time.

FCFS First Come, First Served.

GNSS Global Navigation Satellite System.

ICAO International Civil Aviation Organization.

ILS Instrument Landing System.

KPI Key Performance Indicator.

LP Linear Programming.

MTOW Maximum Take Off Weight.

OD Origin/Destination.

OSSAR Open Source Simulator for ATM Research.

RMO Runway Mode of Operation.

RSA Runway Safety Area.

SA Search Algorithm.

SampEn Sample Entropy.

TAT Turnaround Time.

WTC Wake Turbulence Category.

List of Symbols

Symbol	Description	Unit
acc	Acceleration of an aircraft	[kts/s]
acc_{com}	Comfort acceleration	[kts/s]
B	Butget of an Air Traffic Control agent	[s]
B_s	Bid on segment s	[s]
d_{res}	Reservation distance	[m]
d_{sep}	Separation distance	[m]
dec_{com}	Comfort deceleration	[kts/s]
dec_{max}	Maximum deceleration	[kts/s]
dt	Simulation time step	[s]
$f_{penalty}$	Penalty factor used to increase weight of stop bar segment in advance of a runway becoming active	[-]
h	Heading of an aircraft	[°]
min_speed_frac	Fraction of the minimum speed that is used to determine the weight of an segment	[-]
N	Number of items	[-]
no_turn	Turn angle for which aircraft do not have to slow down	[°]
P	P-value	[-]
p_j	Percentage of the maximum radius that is under investigation to find closest segment	[%]
r	Effect size	[-]
$r_{interest}$	Maximum radius within which the closest segment should be found	[m]
s_i	Length of segment i	[m]
s_{acc}	Acceleration distance to go from speed1 to speed2	[m]
s_{dec}	Deceleration distance to go from speed1 to speed2	[m]
s_{next}	Length of next segment	[m]
t	Simulation time	[s]
t_{arr}	Arrival time	[s]
t_{dep}	Departure time	[s]
t_{reconf}	Time of runway reconfiguration	[s]
t_{reset_bids}	Time after which lost bids are removed	[s]
t_{apron}^{arr}	Apron occupancy time for arrivals	[s]
t_{apron}^{dep}	Apron occupancy time for departures	[s]
t_{runway}^{arr}	Runway occupancy time for arrivals	[s]
$t_{stop\ bar}^{margin}$	Safety margin used by stop bars to determine whether departing aircraft can enter the runway	[s]
$t_{runway}^{between_modes}$	Defines the maximum time allowed between two flights of same runway mode of operation to consider it as one block	[s]
t_{runway}^{cross}	Runway occupancy time for crossings	[s]
t_{runway}^{dep}	Runway occupancy time for departures	[s]
t_{runway}^{hor}	Time horizon within which a runway includes departing aircraft in its schedule of future modes of operation	[s]
t_{runway}^{margin}	Safety margin added before and after a flight's expected time of runway usage	[s]

V	Value of an aircraft, set to its taxi time	[s]
v	Speed of an aircraft	[kts]
\bar{v}	The average taxi speed along a segment during the last minute	[kts]
v_i	The applicable taxi speed for a segment	[kts]
v_{\max}	Maximum taxi speed	[kts]
v_{turn}	Maximum turn speed	[kts]
w_i	Weight of segment i	[-]
w_{original}	Original weight of a segment i	[-]
w_{new}	Updated weight of a segment i	[-]
w_{next}	Weight of next segment	[-]
w_{total}	Total weight of a segment, which combines the weight on the current segment and next segment	[-]
$x_{\text{closest},i}$	Number of times an Automatic Dependent Surveillance-Broadcast (ADS-B) point is closest to segment i	[-]
x_{coord}	Scope of coordination	[-]
$x_{\text{correction}}$	Fraction of segment's length that should be taken into account	[-]
z	Z-value	[-]

Contents

1	Introduction	1
2	Background	3
2.1	Ground Operations	3
2.1.1	Background of Amsterdam Airport Schiphol	3
2.1.2	Insights in the Ground Operation	4
2.1.3	Operational Constraints	6
2.1.4	Operational Uncertainties	7
2.1.5	Agents involved in the Ground Operation	8
2.1.6	Interaction between Agents	9
2.2	Modelling of the Ground Operation	10
2.2.1	Modelling Techniques	10
2.2.2	Research Gap	13
3	Research Framework	15
3.1	Problem Statement	15
3.2	Research Objective and Questions	16
3.2.1	Research Objective	16
3.2.2	Research Questions	17
3.3	Research Scope	17
3.4	Methodological Steps	17
4	Data Processing	21
4.1	Data Sources	21
4.2	Create Airport Layout	22
4.3	Process ADS-B Data	24
4.3.1	Filtering and Cleaning	24
4.3.2	Determine ground path from location points	27
4.3.3	Interpolation to obtain missing data	29
4.3.4	Derive Flight Schedule	30
4.3.5	Limitations of the ADS-B Tool	30
5	Agent-Based Model for Ground Operations	33
5.1	Assumptions	33
5.2	Baseline Model	34
5.2.1	Summary Baseline Model	34
5.2.2	Limitations Baseline Model	34
5.3	Model Specification	35
5.3.1	Implementing Decentralised Control in an Agent-Based Model	35
5.3.2	Agent Specification	35
5.3.3	Environment Specification	40
5.3.4	Interaction among Agents	40
5.3.5	Interaction between Agents and Environment	41
5.4	Model Implementation	41
5.4.1	Model Interface	41
5.4.2	Reservation Mechanism	42
5.4.3	Runway Schedule	43
5.4.4	Entering or leaving the Taxiway System	43
5.4.5	Stopbar Agent	44
5.4.6	Taxiway Closure	46

5.5	Model Analysis and Improvement.	47
5.5.1	Improvements Reservation Mechanism	47
5.5.2	Improvements Runway Schedule	48
5.5.3	Improvements Auction Protocol	49
5.5.4	Improvements Routing Algorithm	50
5.5.5	Improvements Conflict Solving	51
5.5.6	Improvements Taxiway (Dummy) Graph.	52
5.5.7	Additional Adjustments	53
5.6	Simulation Parameters	53
6	Model Analysis and Validation	55
6.1	Experiment Set-up	55
6.2	Results	57
6.2.1	Performance at System Level.	57
6.2.2	Routing Strategies	60
6.3	Sensitivity Analysis	68
6.3.1	Process Sensitivity Analysis	69
6.3.2	Operational Parameters	69
6.3.3	Aircraft Dynamics	71
6.4	Discussion of Results	72
6.4.1	Operational Gridlock.	73
6.4.2	Overall Findings	75
6.4.3	Model Limitations and.	77
6.5	Areas of Improvements	79
7	Conclusions and Recommendations	81
7.1	Contribution	82
7.2	Main Findings.	82
7.3	Recommendations	84
	Bibliography	85
A	Deriving the travelled Routes from ADS-B Data	91
A.1	Determine nearest Segment.	91
A.2	Shift ADS-B Points with similar Time Stamps	92
A.3	Prohibit Aircraft from going Backwards	93
B	In-depth Explanation Baseline Model	95
B.1	Model Operation and Main Findings	95
B.2	Model Environment.	96
B.3	Types of Agents	96
B.4	Interaction among Agents and with Environment.	98
C	Coordination Mechanism	101
C.1	Auction Concept	101
C.2	Determining Bid Price	102
C.3	Design Flaws	103
D	Setting up directed Graph Dijkstra's Algorithm	105
D.1	Initialisation of Taxiway Graph	105
D.2	Initialisation of Dummy Graph	105
D.3	Calculation of a Segment's Weight.	107
D.4	Design Flaws	107
E	Sensitivity Analysis Results of Operational Parameters	113
F	Sensitivity Analysis Results of Aircraft Dynamics	117
G	Statistical Hypothesis Testing	121
G.1	Baseline Scenario	121
G.2	Sensitivity Analysis	122



Introduction

Analyses from EUROCONTROL show that airports were responsible for 12% of the flight delay during October 2017 [16]. In addition, 14% of the, by EUROCONTROL issued, delays were due to airport capacity problems [17]. These delays are highly inconvenient for passengers, as they will arrive late at their destination, and might miss their connecting flight or personal event.

Air Traffic Control (ATC) is responsible for conducting an airport's ground operation in a safe manner, with the aim to do this as efficiently as possible in order to minimise the delay. As the human capability is limited in the amount of information it can absorb and process, air traffic controllers mainly base their route decisions on a minimal amount of global information. These decisions are supported by procedures and fixed taxiway directions, to ease ATC's workload. As these policies restrict ATC's option space, and thereby airport capacity, the consequences of an operational disturbance are long lasting in terms of flight delay.

Amsterdam Airport Schiphol (AAS) is one of the airports that suffers from capacity problems, resulting in Europe's highest share of airport Air Traffic Flow Management (AFTM) delay during the first three months of 2018 [19]. AAS's efficient use of airport capacity is influenced by its unique characteristic to perform 14 runway reconfigurations on average per day. This high number is required to match its wave-like schedule of consecutive arrival and departure peaks, to adapt the set of active runways to the weather conditions, and to satisfy noise constraints issued by the environment council. Even in a transient state of operation caused by a runway reconfiguration, it maintains the goal of ATC to maximise the efficient use of runway capacity. However, during such reconfigurations, multiple, potentially conflicting, flows of ground traffic exist on the manoeuvring area simultaneously, making it hard to utilise the available runway capacity efficiently. Additionally, a runway reconfiguration creates an operational disturbance as the new traffic streams have to be integrated in the already existing traffic patterns. ATC lacks the means and flexibility to solve this disturbance quickly. Consequently, the effects of the operational disturbance are long lasting in terms of flight delay, urging the need for operational improvements.

Decentralised control

Since expanding infrastructure entails high cost, alternative options are explored to improve the ground operation within the available capacity. The main focus is on decreasing the ATC's workload by means of team expansion, reducing tasks due to automation or support of systems to enrich operational knowledge [43, 64]. However, the operation still relies on a centralised controller, who is responsible for many aircraft and tasks in a large area of the airport.

Decentralised control is an alternative approach in which the decision making process is shifted towards a more local level. In the context of ground operations, this could mean a local controller is placed on each of the taxiway intersections. Since each of these agents is responsible for a small region of the airport, they can include detailed information to solve conflicts locally. It is suggested that local controllers enable a more resilient operation due to their decision making at a local level [68]. Consequently, decentralised control could lead to a more efficient use of scarce resources, and thereby an improvement in airport capacity. So far, little understanding is available on the potential and mechanisms of a decentralised control at an actual airport.

Research approach

This research aims at demonstrating the feasibility of implementing decentralised control, by developing a simulation model using the Agent-Based Modelling (ABM) technique. A case study analysis at AAS is performed, using the actual airport layout, an actual flight schedule and historic track data. The analysis is carried out from a validation perspective, by continuously assessing the simulated operation with respect to the actual operation as conducted by ATC. This approach is two-fold: differences are examined to derive and incorporate further requirements of decentralised control. Additionally, the global behaviour that emerges from the local decisions is compared with the decisions made by ATC, to gain a better understanding of the principles and mechanisms of decentralised control. Overall, the simulation model is developed in an iterative manner, based on the assessment with respect to the actual operation. It should be noted that it is not the objective to match the actual operation, but to gather practical knowledge of the requirements and performance of decentralised control.

Main findings

Decentralised control is a feasible approach to manage the ground operations of an actual airport layout. Comparing the simulated and actual operation, it is found that the performance in terms of taxi time and distance show similar patterns. The flexibility of decentralised control is clearly visible in the routing strategies applied, allowing decentralised control to cope with higher route complexities. This research emphasises the need for future operational information and proactive behaviour, in order to be able to manage the dynamics of a ground operation.

Report structure

This report starts with a ground operation background in Chapter 2, covering both the ground operation at AAS and the performed literature review. Based on the acquired understanding of the operation and state-of-the-art in literature, the research framework has been drawn and can be found in Chapter 3. Historical operational data of AAS is processed to obtain the data required to compare the simulated operation with the actual operation. The steps that have been taken to perform the extensive data processing are explained in Chapter 4. The development of the agent-based simulation model is covered in Chapter 5. Chapter 6 describes the setup and findings of a case study at AAS. This case study is used to validate the model and to discuss the results in terms of requirements to apply decentralised control to a full scale and complex operation. Finally, the conclusions and recommendations for future research are found in Chapter 7.

2

Background

In 2001, EUROCONTROL identified that airport congestion is one of the main constraints to enable future growth in aviation [15]. To obtain insights in the causes of airport congestion, as well as the complexities involved in the operation, the ground operations at Amsterdam Airport Schiphol (AAS) have been observed. A summary of these insights is given in Section 2.1. Since EUROCONTROL's announcement, different elements of the airport's ground operation have been investigated to improve the operational efficiency. The state of the art on how to model an airport's ground operation has been analysed. An overview of the reviewed literature is found in Section 2.2.

2.1. Ground Operations

AAS is selected as case study, as the type of challenges faced by Air Traffic Control (ATC) in terms of airport layout and operation are representative for many airports. Subsection 2.1.1 discusses the background of AAS and explains based on what features this airport is selected as case study. To understand the dynamics, complexities and aspects that are involved in an airport's ground operation, the actual ground operation at AAS is analysed. By observing the ground movements via *Flightradar24*, analysing the radio communication with ATC via *LiveATC*, and speaking with experts and (former) air traffic controllers, a detailed understanding is acquired. A summary of the insights is provided in Subsection 2.1.2, followed by an identification of the operational constraints as listed in Subsection 2.1.3.

2.1.1. Background of Amsterdam Airport Schiphol

AAS is used as case study to obtain an understanding of the actual ground operation. Additionally, it serves as input for validating the simulation model, that is to be designed to evaluate the feasibility of applying decentralised control to an ground operation of an actual airport. This subsection provides a general overview of the airport layout and reasoning why AAS is selected as case study.

Airport layout

During its 100th anniversary year in 2016, a total of 63.6 million passengers travelled via AAS to one of its 322 destinations [26]. Over these past 100 years, the infrastructure of AAS has grown to a layout consisting of 6 different runways.

As shown in Figure 2.1, the airport layout possesses some important characteristics. First of all, there is a set of main taxiways surrounding the terminals and piers of AAS. Figure 2.2 zooms in on the main taxiways of AAS. Two parallel taxiways can be identified, namely taxiways Alpha (A) and Bravo (B). In normal operations, Alpha is used in clockwise direction, while Bravo is used anti-clockwise. There is one single taxiway highlighted, Quebec (Q) which has no parallel taxiway. For the remainder of this report, these taxiways are referred to as "single lane taxiways". ATC follows the procedure that taxiway Quebec (Q) is used in one direction only, and the direction of this taxiway depends on the runway configuration in use.

Many of AAS's runways (including 09 – 27, 06 – 24 and 18L – 36R, see Figure 2.1) have their entry or exit directly linked to taxiway Bravo. This means that a long departure queue waiting for a runway could block the taxiway system for arriving aircraft, as well as taxiing aircraft that want to leave or enter the apron area.

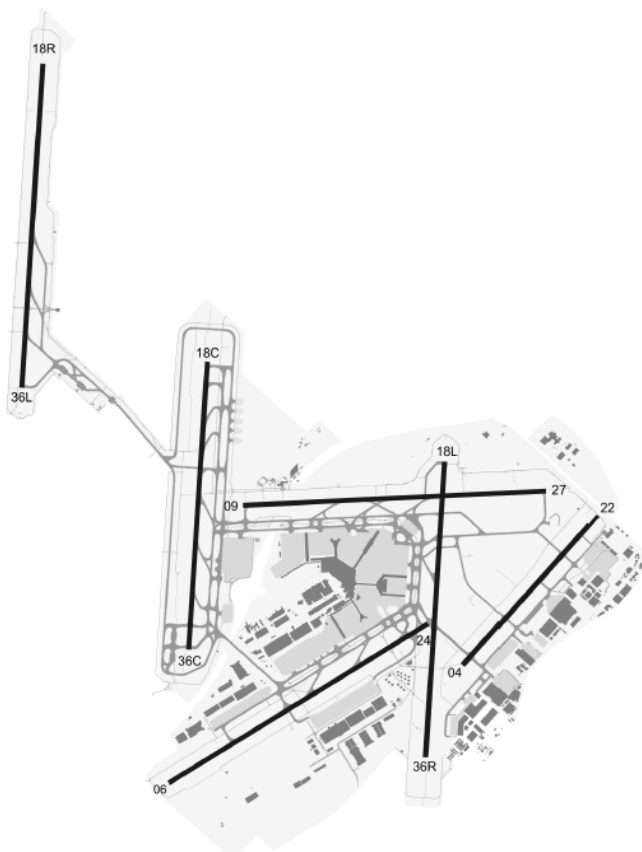


Figure 2.1: Layout of Amsterdam Airport Schiphol [47]

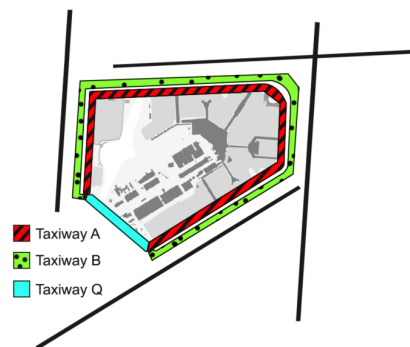


Figure 2.2: Important taxiways of Amsterdam Airport Schiphol [47]

Runway reconfiguration

During busy days, AAS handles over 1500 flights a day [70]. However, it is not necessarily this large number of flights that causes congestion or challenges for ATC, because the steady state of operation is a rather streamlined process in which the available runway capacity can be used efficiently. However, due to dynamics the operation can enter a transient state in which efficiency is likely to decrease. One possible cause of such a transient state, is a runway reconfiguration that is being executed to adapt the set of active runways to the circumstances at stake. This is done to obtain the maximum runway capacity within the set of available runways. One of AAS's characteristics is its wave-like schedule of consecutive arrival and departure peaks. It is not sufficient to alternate the amount of runways used, as runway usage also depends on weather conditions and regulations. More information on this matter can be found in Subsection 2.1.3. On average, AAS performs 14 runway reconfigurations per day. It maintains the goal of ATC that the use of runway capacity is maximised, even in the disturbed operation that is caused by the runway reconfiguration. For maximal use, it is required that the distance between two consecutive aircraft is minimised, while no aircraft are waiting in a runway queue. However, during such reconfigurations, multiple, potentially conflicting, flows of ground traffic exist on the manoeuvring area simultaneously. This makes it difficult for ATC to utilise the available runway capacity efficiently. Even during these runway reconfigurations, ATC has to control over a 100 aircraft per hour that either land or take off [70]. The complexities linked to this phase of the operation open an interesting case to gain an understanding of the challenges and constraints that are involved in an actual ground operation.

2.1.2. Insights in the Ground Operation

The intention of this subsection is to give a general overview of the ground operation process. This informative section is used as a starting point on which the following sections and chapters continue to build. This information is based on observing the ground movements and analysing radio communication between pilots and ATC, conducting interviews with experts and (former) air traffic controllers, and studying operating procedures manuals of air traffic controllers [47].

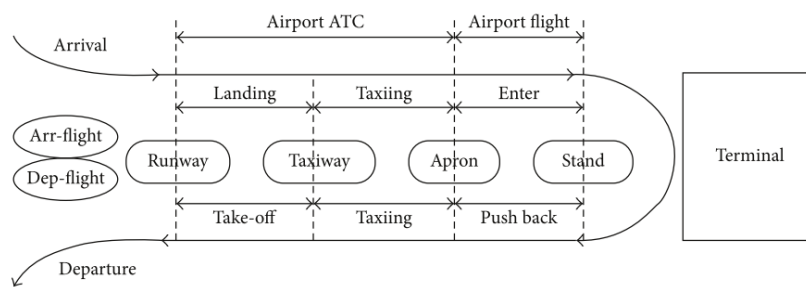


Figure 2.3: Work breakdown of the ground operation [35]

Breakdown of ground operation

The different steps that are part of the ground operations are displayed in Figure 2.3. Arriving aircraft receive a landing approval when they are on final approach. Once touched down, these aircraft should leave the runway as quickly as possible to make room for the next aircraft. A taxi route is allocated by ATC, which guides the aircraft towards its stand. In case the gate is not available (shortly), the aircraft is guided to a holding position such that it does not block taxiways for other aircraft. In air transportation the standard holds that aircraft coming from the right have priority [33].

Once an aircraft has arrived at its stand, ground handlers have to perform multiple actions to get the aircraft ready for its next flight. These operations include disembarking, refuelling, cleaning, catering and boarding. Although an aircraft has a scheduled time to be released from its stand, delays are likely because of e.g. the aircraft arriving late, limited availability of resources, severe weather conditions or restrictions from EUROCONTROL.

Once ready, the departing aircraft notifies ATC and, if the situation allows, a start-up approval is given, followed by a push back approval. This allowance depends on multiple conditions, including whether vehicles and personnel have left the stand and whether direct push-back would be possible.

After its push back, the aircraft receives a taxi route and approval to taxi. Depending on its route, approval might be required to cross specific points in the airport infrastructure, including a runway. When arriving at the runway, the departing aircraft once again waits for approval from ATC to line up and take off.

Controlling the operation

Although a safe operation is the main goal of an air traffic controller, its their secondary objective to run an efficient operation. The two main control parameters are the routes allocated to each aircraft, and the number of aircraft that are simultaneously present on the taxiway infrastructure.

The layout of AAS's infrastructure (Figure 2.1 and 2.2) plays an important factor. Air traffic controllers aim at obtaining stable streams of traffic using taxiway Alpha and Bravo in clockwise and anti-clockwise direction respectively. As noticed by the Dutch Safety Board [14], air traffic controllers may decide to deviate from this procedure. Reasons to do so can be that there is a queue in front of the aircraft, or when a taxiing aircraft has to get around an aircraft that is being pushed back to a location on one of the main taxiways. However, it has to be communicated clearly with the pilots that the aircraft should deviate from standard procedures.

It is stated in Subsection 2.1.1 that AAS's layout could cause runway queues of departing aircraft to build up on the main taxiways, blocking the taxiway system for other aircraft. It is therefore of importance that only limited aircraft receive a push back at the same time, in order to reduce potential conflicts on taxiways.

Taxiway conflicts

The mentioned procedures regarding taxiway direction and limited push backs are not sufficient to provide a safe operation. Aircraft still have to adapt speed or stop in order to prevent a conflict in which aircraft get too close, especially during a runway reconfiguration in which multiple flows of ground traffic exist on the manoeuvring area. Figure 2.4 shows three possible taxiway conflicts that can occur. The one on the left shows the situation where two aircraft arrive at (almost) the same time at an intersection and both want to use the same taxiway. In normal case, an aircraft approaching from the right has priority. However, there are two exceptions. Firstly, an air traffic controller can tell a pilot that it has, or has to give, priority. Secondly, an arriving aircraft leaving the runway via a runway exit has priority over other aircraft. That is because its speed is much higher, as well as the fact that the runway has to be emptied such that the next aircraft can land safely.



Figure 2.4: Possible types of taxiway conflicts [67]

Table 2.1: Minimum separation distance between aircraft [47]

	Trailing			
Leading	SUPER	HEAVY	MEDIUM	LIGHT
SUPER	-*	6 NM	7 NM	8 NM
HEAVY	-*	4 NM	5 NM	6 NM
MEDIUM	-	-	-	5 NM
LIGHT	-	-	-	-

* Amsterdam Airport Schiphol aims for increased radar separation minimum to 4 NM for additional safety

- ICAO radar separation minimum

The middle conflict in Figure 2.4 indicates that the trailing aircraft has a higher speed than the leading aircraft. It is the pilot's responsibility to maintain a safe distance with respect to the leading aircraft.

The third and most right conflict as shown in Figure 2.4 is the situation in which two aircraft are heading towards each other on the same taxiway. Also, for some apron areas there is only one way to both enter and exit. The situation in which two aircraft use the same route at the same moment should be avoided in all times, as it causes a gridlock in the taxi system due to the fact that aircraft cannot move backwards. Consequently a large inefficiency is introduced, which has to be solved by push back trucks. It is one of the main reasons why ground controllers want to stick as much as possible with the original taxiway directions of Alpha and Bravo. Also, the current direction of single lane taxiways like Quebec have to be clearly communicated.

2.1.3. Operational Constraints

Air traffic controllers are not completely free in solving the complexities that arise in an airport's ground operation. This subsection lists a few constraints that limit an air traffic controller's option space.

Separation

One of the regulatory frameworks is related to the minimum separation between two aircraft to maintain a safe operation. This separation can either be distance based or time based, which both have to be satisfied.

Distance based separation depends on two aspects, of which the first aspect is related to the turbulence created by an aircraft's wingtip vortices and the fast moving gasses that are released by the aircraft's engines. Since both features heavily depend on the size of an aircraft, the International Civil Aviation Organization (ICAO) grouped aircraft in 4 different categories. These so-called Wake Turbulence Category (WTC) are based on an aircraft's Maximum Take Off Weight (MTOW) [9, 10]. Another aspect is radar separation, which is applied to account for deviations of an aircraft's actual position and the position as seen by a radar. In addition, the radar has to be able to distinguish the radar pulses reflected by two aircraft, which is why separation between aircraft should be larger than the radar's range-resolution. ICAO standards state that that radar separation in the terminal area airspace should be at least 3 NM and may be reduced to 2.5 NM on final approach [9]. The required separation resulting from the wake turbulence separation and radar separation as defined by ICAO regulations are shown in Table 2.1 [9, 10].

Time based separation is a strengthened separation interval. It is applied between two departing or two arriving aircraft due to the effects of wake turbulence [9].

Selecting which runways to use

Another important constraint for ground operations is related to the runways that can be used by ATC. The choice of active runways is mainly constraint by weather conditions, airport demand and noise regulations, which all depend on one another making it a challenge to select the optimal runway configuration.

Wind conditions play an important role in selecting a valid set of runways, since each airport has wind limits to maintain safe runway operations. First of all, typically tailwind components are not allowed to exceed 10 knots. Secondly, crosswind is limited to 15-25 knots, depending on the airport [11]. AAS has set its

crosswind limit and tailwind limit on 20 knots and 7 knots respectively, which can be lowered depending on the visibility range and ceiling (cloud base) [47]. Sometimes, a specific runway cannot be used due to heavy (thunder)storms on the arrival or departure routes. Also fog should be taken into account, as dependent runways are not allowed to be used in case visibility is less than 5 kilometres or ceiling is below 1000 feet [71].

Additionally, AAS is characterised by its wave-like schedule of consecutive arrival and departure peaks. As a consequence of agreements with its community council, ATC does not have full freedom in selecting the type and number of active runways. Noise restrictions from Schiphol Community Council have led to a list of preferred runway combinations during either day and night, which is presented in 2013's agreement [2, 3]. The agreement requires that at least 90% of the time the most noise preferred set of runways is active. In addition, ATC is allowed to use 2+1 runways during inbound or outbound peaks. This means two runways can be used for landing and one for take off, or vice versa. During the so-called off-peaks, ATC is allowed to only use 1+1 (one take off runway, one landing runway). This same strategy is applied during night time (22.30 - 06.30 local time). For a limited period of time, two take off and two landing runways (2+2) can be used to make the transition from one peak to an other. Summarised, ATC has to adapt its set of active runways to match demand and comply with regulations, which is the main reason why AAS performs so many runway reconfigurations every day.

Other factors limiting the choice of active runways is related to runway dependence due to e.g. jet blast of departing aircraft on an other runway, or crossing runways (18L – 36R and 09 – 27, see Figure 2.1). Additionally, a missed approach path of a runway might overlap with a departing route, which is the case for runway 06 interfering with runway 09.

Additionally, sudden (operational) events could cause (temporarily) closing of a runway. Examples are a sudden change in wind direction, a bird strike, or aborted take off. Technical failure of runway systems (e.g. lightning) as well as required snow removal are also reasons to close a runway. It is also possible that an arriving aircraft declares an emergency, after which ATC can decide to temporarily open a runway for that aircraft to land. It has to be noted that most of the mentioned factors are rare events, but each of them requires a quick reaction in terms of a runway reconfiguration.

Slot regulation

Some other constraints worth mentioning are gate occupancy and slots issued by EUROCONTROL. Regarding the former, in case an arriving aircraft touches down on an airport, ATC would like to guide it as quickly as possible to its gate, in order to minimise the traffic it has to control. However, it can occur that the gate of the aircraft is still occupied, requiring ATC to hold the aircraft on the airport's infrastructure. This can be on a taxiway that is either in use or not used depending on the set of active runways, or on a specific remote parking location. Each of these locations limit ATC's usable workspace and increase traffic density. The latter scenario occurs when EUROCONTROL tries to limit the number of aircraft in a specific airspace or area, by giving tactical time slots to an aircraft in which it should take off (so called Calculated Take Off Time (CTOT)) [20]. Not only does this mean that a so-called regulated aircraft might hold longer at its gate than planned, it also requires ATC to prioritise this aircraft in the pre-departure sequence such that it can depart at its CTOT.

2.1.4. Operational Uncertainties

The process of an air traffic controller from landing an aircraft up to the moment it becomes airborne from the runway for its departing flight includes many aspects of uncertainty. This subsection lists the main sources of uncertainty that have an effect on the operational efficiency.

The main sources of uncertainty are:

1. time it takes to taxi from gate to runway or vice versa;
2. duration of push back;
3. duration ground handling;
4. moment in time an arriving aircraft touches down on the runway;
5. time it takes an aircraft to cross a runway;
6. time an aircraft has to spend in the departure queue;
7. time an aircraft is occupying the runway;
8. time when an aircraft is ready for its push back;
9. time when an aircraft receives its push back;
10. time an aircraft has to wait after push back to start taxiing;
11. time separation between two trailing aircraft on a taxiway;

12. selecting the fastest taxi route;
13. taxi speed;
14. runway exit used by an arriving aircraft (depending on its roll distance);
15. whether an aircraft is ready within its engine start-up window;
16. whether an aircraft is actually ready when it makes the call it is ready;
17. whether an aircraft follows the allocated taxi route;
18. delay caused by a runway reconfiguration;
19. weather / wind forecast.

Uncertainties 1 until 10 are related to the duration of specific actions, or the moment when particular activities are likely to happen. Consequently, air traffic controllers continuously have to adapt their plans, depending on whether their plans become reality. This leads to situations in which the selected routes are non-optimal (12). Uncertainty in runway exit used and aircraft speed requires ATC to take additional safety measures in terms of holding aircraft at position until others passed by (items 13 and 14). ATC does not have (full) information of what happens on the ground or in the cockpit, such that non-optimal decisions are made due to a lack of operational insights (items 15, 16 and 17). It is noted before that a runway reconfiguration causes an operational disturbance, of which the magnitude is unknown (item 18). Also, it is unknown whether the right set of runways is active, as weather conditions remain (to some extent) unpredictable (item 19).

So far limited literature analysed the effects of the operational uncertainty at airports. One study has come up with a list of 18 different sources of uncertainty [56]. Historic data of Detroit Airport was then used to perform the quantitative analysis for the uncertainty sources 10, 12, 13 and 18. Fast time simulations are used H. Lee and H. Balakrishnan to investigate the impact of the uncertainty sources 9, 11, 13 and 14 [44].

2.1.5. Agents involved in the Ground Operation

This subsection describes the different actors (agents) that perform relevant activities in the ground surface movement operations.

Aircraft

This report refers to an aircraft controlled by the pilot as "Aircraft". Focusing on departing aircraft, the individual goal of each aircraft is to become airborne as soon as possible and to comply with the airline schedule. For arriving aircraft it is the other way around, i.e. to arrive as quickly as possible at the gate. The objective to minimise taxi time, and thereby fuel burn, cost and emissions, are derived from the airline.

Delivery Controller

The delivery controller is responsible for communicating the en-route clearance to departing aircraft. The information includes squawk code, assigned take off runway and route clearance, and is shared via either radio or data link. Additionally, slots times (CTOT) issued by EUROCONTROL are communicated by the delivery controller. The individual goal is to provide each departing aircraft with its initial route clearance. Only after the pilot's confirmation, the aircraft will be given a spot in the pre-departure sequence.

Outbound planner

When departing aircraft report over the radio that they are ready, it is up to the outbound planner to decide whether the pilot should be forwarded to the start-up controller. A Collaborative Pre-Departure Sequence Planning (CPDSP) tool determines the engine start-up of each aircraft and lists which aircraft have (just) been cleared. This latter aspect is of importance to determine whether a queue is expected at the runway as well as the fact whether adjacent aircraft have just received their clearance, which could cause a conflict. The planner gives information about weather and airport conditions. Additionally, expected taxi time is entered in the CPDSP system to gather data about future traffic, which is used to determine whether new clearances can be given. The individual goal is to regulate the flow of aircraft (and therefore workload) that are forwarded to ground control.

Start-up controller

It is the start-up controller's task to give start-up clearances to the aircraft that are ready for departure. Decisions are based on the departure sequence planning of the outbound planner, the number of aircraft in the field, and length of the queues at the departure runway. The latter two aspects are also an indication of the

current workload of the ground controller. An individual start-up controller no longer exists at AAS. For aircraft on stands, and for taxi-out stands this task is performed by the ground controller and outbound planner respectively. The individual goal is to make sure that the aircraft's engines can be started safely.

Ground controller

The ground controller is responsible for giving push back and taxi instructions for both arriving and departing aircraft. In addition, the ground controller is responsible for maintaining separation between aircraft on the taxiways as well as on the apron area. Although the right-of-way standard applies, which state that aircraft coming from the right have priority, a ground controller aims at actively assigning priority to one of the two involved aircraft in order to maintain safety and to increase efficiency of the ground operation. The individual goal is to safely guide departing aircraft from their stand to the runway, while guiding arriving aircraft in the opposite direction. Additionally, the controller aims at obtaining an optimal departure sequence by prioritising aircraft on the taxiway.

Runway controller

A runway controller is responsible for providing a safe separation between successive departing aircraft, based on the values as provided in Subsection 2.1.3. Additionally, arriving aircraft should receive a landing clearance if the runway is free and available. If a taxiing aircraft wants to cross an active runway, the runway controller has to give a clearance when a safe crossing is possible. The individual goal of the runway controller is to maintain a safe separation between departing aircraft as well as arriving aircraft. The controller aims at adhering to the planned departure sequence as setup by the ground controller, although changes occur to minimise total time required for take off.

Ground handler

A ground handler prepares an aircraft for its next flight, by refuelling, cleaning and catering the aircraft. The handler is responsible for updating the aircraft's readiness times in accordance with the Airport Collaborative Decision Management (ACDM) programme, based on its own insights and information from the cockpit. The individual goal is to minimise the Turnaround Time (TAT) of an aircraft, such that an aircraft can depart as early as possible. In addition, the ground handler wants to minimise time difference between actual and target readiness time, in order to maximise the service level of the handler organisation.

2.1.6. Interaction between Agents

The previous subsection described the different agents that are involved in an airport's ground operation. The interaction that takes place between these actors are explained in this subsection.

Figure 2.5 shows the flows of interaction between the different agents:

- **Link 1:** A ground handler updates the flight readiness times of the aircraft it performs operations on and forwards these timestamps to the outbound planner.
- **Link 2 & 3:** Aircraft interact with ground handlers to discuss readiness status of the aircraft. From the aircraft's point of view this includes e.g. boarding and flight preparation, while the ground handler has to perform activities like refuelling and catering.
- **Link 4:** Aircraft interact with other aircraft to maintain separation. An aircraft should adapt speed when trailing a slower moving aircraft. In addition, an aircraft should reduce speed or even wait at a taxiway intersection in case another aircraft has priority as it comes from the right, due to restrictions from ATC, or when an arriving aircraft leaves the runway exit.
- **Link 5 & 6:** A departing aircraft requests information regarding clearance delivery, while the controller provides this initial route clearance. This clearance includes squawk code, assigned take off runway and route clearance.
- **Link 7 & 8:** When a departing aircraft is ready for departure, it informs the outbound planner. If the circumstances allow, the outbound planner will forward the aircraft to the ground planner. In addition, weather and airport information is communicated.
- **Link 9 & 10:** Departing aircraft request a ground controller for push back, which it receives if the situation allows. The controller also allocates a taxi route to both departing and arriving aircraft. In addition, the controller can give instructions to aircraft to solve conflicts and maintain separation. Arriving aircraft can also be guided to a remote holding position if its stand is still occupied. An aircraft should confirm (or read back) all ground controller's statements.

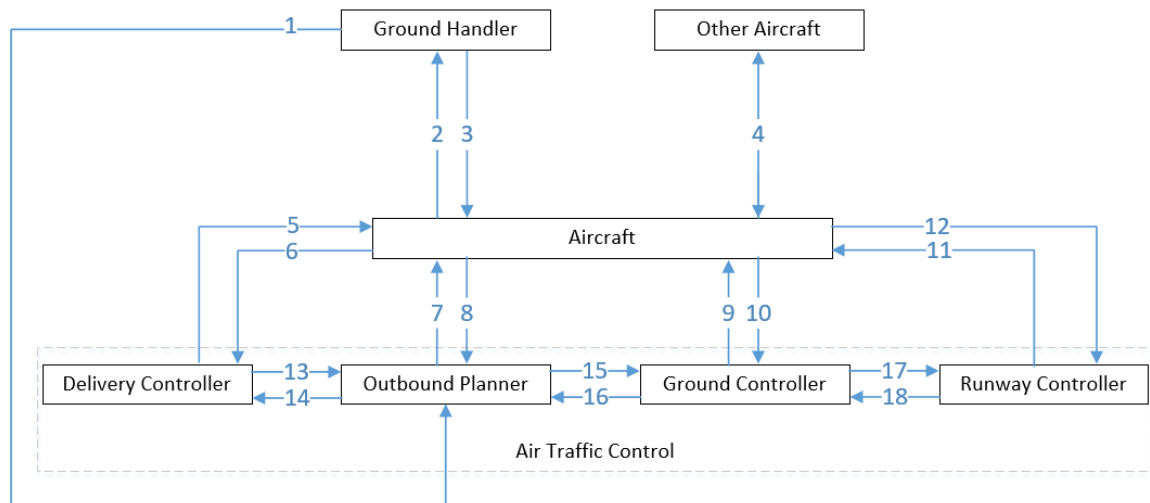


Figure 2.5: Diagram of interactions between the agents involved in a ground operation

- **Link 11 & 12:** A runway controller gives clearances to departing aircraft to line up on the runway and take off, arriving aircraft a clearance to land, and taxiing aircraft a clearance to cross an active runway. All aircraft should confirm the runway controller's statements.
- **Link 13 & 14:** The delivery controller hands over the responsibility of an aircraft to the outbound planner once the clearance delivery is confirmed by the aircraft. Circumstances like sudden runway reconfiguration might result in the fact that new clearances in terms of runway or departure route have to be given, requiring a hand back to the delivery controller.
- **Link 15 & 16:** Interaction is based on handing over the responsibility of an aircraft. It happens occasionally that an aircraft states that it is ready, while in fact it is not. Consequently the aircraft is handed back over to the outbound planner. By analysing the number of flight strips in front of the ground controller, the outbound planner can get insights in the ground controller's workload and thereby decide whether a new aircraft can be forwarded.
- **Link 17 & 18:** Responsibility of an aircraft is handed over from the ground controller to the runway controller at the moment a departing aircraft arrives at the runway, while the responsibility of arriving aircraft travels the other way. If a taxiing aircraft has to cross an active runway, the responsibility is temporarily in hands of the runway controller.

Depending on the traffic state and number of aircraft in the field, responsibilities of air traffic controllers can differ. During a quiet period of time, one air traffic controller can be delivery controller and outbound planner simultaneously. During busy moments, multiple runway controllers and ground controllers are present, each responsible for a specific region of AAS's layout.

2.2. Modelling of the Ground Operation

Since EUROCONTROL identified in 2001 that airport congestion is one of the main constraints to make further growth in aviation possible [15], multiple studies focused on different aspects to increase an airport's efficiency. The literature is reviewed to obtain the state of the art regarding the modelling of an airport's ground operation. Insights in the main directions and the used modelling techniques are discussed in Subsection 2.2.1, followed by the identified research gap in Subsection 2.2.2.

2.2.1. Modelling Techniques

The main research direction aims at determining relations that can be processed in a tool to support air traffic control in their activities. Four main topics are identified in relevant studies:

1. **Optimisation of runway sequence:** models are created to optimise the sequence of departures based on a flight schedule, taking the separation minimums between two departing aircraft into account.
2. **Optimisation of gate release:** to decrease taxi delays due to taxiway conflicts and a runway queue, the effect of allowing only a limited number of aircraft on the taxiway simultaneously is analysed.

3. **Shortest path taxiways:** different algorithms are designed to determine the optimal taxi route that aircraft have to follow, which could refer to optimal path in terms of distance as well as taxi time.
4. **Estimation taxi time:** historical data is used to determine the impact of operational parameters on the taxi-out time. The acquired relations are used to estimate the taxi-out times of current operations.

The studies are grouped based on the modelling technique used, as each of the techniques comes with its own advantages and disadvantages. The most relevant and interesting studies per modelling technique are discussed below. Each study is evaluated in terms of:

1. **Objective:** insights in the type of research, to gain an understanding of the approach and focus.
2. **Simplifications and Assumptions:** due to the complexity of the operation, simplifications are needed to transform it into a theoretical model. These simplifications must be carefully considered, which is why the level of abstraction, and effect of these assumptions, is assessed.
3. **Degree of uncertainty:** ATC creates tactical plans on how to guide aircraft efficiently towards their destination. In a stable and predictable operation, ATC is able to successfully execute these plans. However, challenges are faced in a dynamic operation, as ATC's plans are continuously disturbed due to expectations in aircraft's behaviour that are not met. The included level of uncertainty gives insights into the degree of dynamics.
4. **Level of validation:** validation is required to determine the extent to which the designed model represents the actual operation. Also, this is an important stage to evaluate and explain potential differences between the model and reality.

Types of Linear Programming

One of the most frequently used techniques to model an airport's ground operation is the application of a type of Linear Programming (LP). This technique can be used to model almost all processes. Although this modelling technique has the advantage that an optimum will be found, it also has many disadvantages. First of all, runtime of the algorithm increases exponentially with size as is confirmed by Malik et. al. [48]. Several studies show that the size of the problem is not only linked to number of aircraft, but also to the selected time discretisation [5, 51, 67]. In addition, linear programming is seen as a centralised approach as the autonomous behaviour of the aircraft agents is not taken into account. Consequently, neither uncertainty, nor interaction/conflicts between agents can be included.

Most studies had similar objectives: minimise taxi time and delay [5, 45, 46, 50, 51, 57, 60, 67], minimise time difference between actual and scheduled takeoff time [40, 45, 46], or maximise runway throughput [48].

By comparing the different studies, the following aspects are noted:

- Some studies completely exclude arriving aircraft [48, 60], while others analyse different strategies in which arriving aircraft are partly excluded [46]. Balakrishnan and Jung, and Balakrishnan and Lee prioritise arriving aircraft above departing aircraft [5, 45].
- Most studies do not include any validation aspects in order to determine the model's performance. An exception is made by one study which uses observations to validate the model [46]. Most studies use a randomised schedule instead [40, 48, 57, 60].
- As stated before, uncertainty cannot be taken into account using a form of LP. Different studies test the effects of uncertainty by running simulation runs with different parameters and flight schedules [5, 48], while Rathinam et al. [57] use randomly selected readiness time as well as a random gate location.

Types of Search Algorithm

A Search Algorithm (SA) aims at finding variables that create either a minimum or maximum, while satisfying multiple constraints. It does so by applying brute-force and heuristics to their case, depending on the type of algorithm. The largest advantage of the two main types of search algorithm (Genetic Algorithms and Tabu search) is that they can quickly provide the user with a relatively good solution. The reason *relatively* is used is because the algorithms do not guarantee a global optimum, as the algorithm sometimes ends in a local optimum [69]. A SA is solved from a centralised approach point of view, meaning that no interaction, conflict solving or behaviour of the autonomous different agents is taken into account; an optimum is found for the complete system, assuming all aircraft will adhere to their assigned routes and times. Uncertainty cannot directly be included in the algorithm.

Most studies based on a type of SA focus on optimal allocation of taxi routes [23, 35, 53], while others take the combination of runway sequence, allocation of taxi routes and optimal push back time into account [21, 36, 42]. Atkin et al. [4] analyse the runway sequence order, as well as taxi route allocation, while Koeners

and Rademaker [41] focus on runway sequencing. More than half the studies focus on minimising total time cost of taxiing aircraft [21, 23, 35, 36, 42, 53]. Gotteland and Durand [23] penalises routes of aircraft based on the length. Koeners [42] also analyses the distribution of taxi time of the individual aircraft, while Jiang et al. [36] focus on obtaining a fair distribution of individual taxi times.

By comparing the different studies, the following aspects are noted:

- An equal balance is made between the studies that include arriving aircraft [23, 35, 36, 53], and the ones that exclude them [4, 21, 41, 42]. Gotteland and Durand [23] are the only ones stating that arriving aircraft can be delayed up to 30 seconds in the air. Since Jiang et al. [36] forbid aircraft to hold at a node, this also means arriving aircraft are postponing their landing until their route is available.
- Most studies are based on an actual flight schedule [4, 23, 35, 41, 42] ranging from data of 9 flights [35] up to one full day [23, 53]. Koeners et al. [42] limit themselves to data of 1 departing runway only. A validation based on comparison of actual and simulated data is done by Jiang et al. [35]. Multiple studies compare their results with other algorithms to determine the efficiency [23, 36, 41, 42]. The model designed by Atkins et al. [4] is used in a real life situation to validate the model.
- Uncertainty is excluded in multiple studies [21, 35, 36, 53]. Gotteland and Durand [23] include uncertainty in taxi speed by iterating on a regular basis. Koeners et al. investigate the uncertainty in speed and push back time [42], or the effects of delay using different flight schedules [41].

Historical Data

Combining large data sets with techniques to analyse them results in valuable lessons regarding an airport's performance. Two directions to analyse historical data were found in literature; using data to predict an aircraft's taxi time [31, 37, 64–66] or using data to predict how many aircraft should be pushed back in a certain period of time (so-called push back rate) [65]. An advantage of this technique is that actual data is being used to make future predictions. However, since each situation on the ground surface of an airport is different, historical data of taxi times during a low density state is not representative for a high traffic density, or vice versa. In addition, this type of model only focuses on time predictions, leaving out the actual interactions and conflicts between agents. Furthermore, local congestion is left out of the model meaning that aircraft should be pushed back, even when the apron area is congested. Uncertainty in this model is included, a probability distribution of historical taxi time per taxiway is created, such that this uncertainty is taken into account when predicting the taxi out times. However, there are no insights in the exact source of delay.

Most studies aim at making an accurate estimation of the taxi-out times [31, 37, 64–66]. Simaiakis et al. [64, 66] use data in combination with Markov Chains to determine the optimal push back rate. Simaiakis and Balakrishnan [65] analyse the impact of number of aircraft in the field on taxi time performance, based on historical data.

By comparing the different studies, the following aspects are noted:

- Three studies include arriving aircraft [31, 37, 64], in which Idris et al. [31] conclude that arrival demand has a low impact on taxi times. Others exclude arriving aircraft in their analysis [65, 66].
- Uncertainty is included in all studies, since probability distribution of taxi time are used. Historical data allows the establishment of relations, however the exact of uncertainty is unknown due to the dependence of variables.
- All five studies use historical data to train the model and perform the analysis, after which a different data set is used to validate the model. The size of the data set is either 20 days [37], one month [31], or 1 year [64–66]. Some studies include multiple field tests to validate the created tool [64, 66].

Agent-Based Modelling

The popularity of the novel technique Agent-Based Modelling (ABM) is increasing thanks to its ability to give a natural description of a socio-technical systems [6, 74]. An agent-based model is created bottom-up where each involved unit or player is modelled as an individual agent with its own field of observation, capabilities to communicate or interact, as well as autonomous behaviour. The model developer is free to set the level of abstraction by specifying an agent's behaviour in terms of responsibilities and dependencies. Each agent is characterised by a set of observables, which might be related to each other by equations and which can be changed by an individual's activities. Additionally, the interaction, communication and information sharing between different agents can be specified. This makes it possible to create *artificial* boundaries, like a group of agents (e.g. a company), between which particular data is shared. This allows heterogeneous components to be formed at different levels [22].

Due to the fact that ABM is based on developing a model bottoms-up, it has some major advantages with respect to the earlier mentioned techniques. First of all, ABM allows complex behaviour on an individual level. As each agent has its individual behaviour, different variations and uncertainties can be applied on a local level. Additionally due to the model's flexibility, an agent's behaviour is able to change over time, e.g. it adapts to the circumstances or learns from earlier decisions. Additionally, uncertainty is easily added per individual. Many other techniques have difficulties to include uncertainty and a dynamic behaviour on local level, and are based on (high level) equations between observables. As the model complexity rapidly increases for larger systems, those techniques have difficulties to include uncertainty and a dynamic behaviour on local level. Often, parameter values originating from taking the space average of time average are applied in the model to maintain acceptable run times.

Another strong aspect of ABM is the fact that it is being designed from a low-level, meaning that e.g. individual processes are modelled instead of the parent processes, or individuals instead of groups. As a consequence, the interaction and autonomous behaviour of the different agents could show emergent behaviour of the complete system, which would not be obtained when modelling on a higher level. If the model is designed properly, different and more realistic insights are gathered with respect to models that are designed on a system level [6, 74].

Peng et al. [52] focus on route allocation to aircraft to determine airport capacity, where others combine route allocation with runway sequencing [43]. For both studies it has been the aim to minimise taxi time of the individual aircraft. The objective of Rafegas [55] is to understand the potential of ABM to increase airport capacity. Udluft has demonstrated the feasibility of decentralised control, by designing an agent-based mode to simulate an airport's ground operation [73]. For the latter two studies, taxi time is seen as (one of the) performance indicators to evaluate the performance of ABM.

By comparing the different studies, the following aspects are noted:

- Udluft based his conclusions on departing traffic only [73], while others included both arriving and departing traffic [52, 55]. No information about the flight scenario is provided by Lancelot et al. [43].
- Peng et al. [52] use a different algorithm to validate the model, although a fictitious airport with random schedule is used. Others tested their designed tool in a real-life environment [43]. Rafegas and Udluft [55, 73] have not validated their model, as both use a fictitious airport and random schedule.
- Stochastic perturbations is included in aircraft performance [52], or turnaround times of aircraft [55]. Others use real-time status updates of aircraft, and input of actual controllers, to discover best routes [43]. Udluft applies Monte Carlo Simulation to his operation [73].

2.2.2. Research Gap

Although multiple studies focus on the optimisation of an airport's ground operation, the theoretical results cannot be used in practice. The reason for this is that transforming a socio-technical process (like an airport's ground operation) into a theoretical model can only be done effectively if the actual dynamics of the operation are well understood. Large pitfalls that are identified in literature are related to leaving out essential aspects of the operation, including arriving aircraft, uncertainty as well as aircraft interactions and conflicts.

Transient state of operation

One of the current limitations in literature is that most studies base their results on simulating a steady-state operation. Discussions with (former) air traffic controllers revealed that it is not the steady-state that creates the largest challenges. Although maybe not optimal, ATC has standard procedures and strategies to deal with this kind of operation. Instead, one of the main challenges ATC has to deal with is the transient state of operation, which can be caused by e.g. a change in runway configuration, a traffic disruptions, or closure of airport infrastructure. The operational standards limit ATC's option space to deal with the dynamics and complexities that arise from these situations.

Hunter [29] analysed 27 airports to quantify the decrease in airport capacity as a result of runway reconfigurations. He correctly identifies that reduced capacity is caused by a loss of available departure traffic as a consequence of the disruption in steady traffic flows. The only studies that include a transient effect focus on a sudden taxiway closure [7, 53]. Although the initial steps are made to quantify the effects of a transient state, no research focused on improving the efficiency of ground operations in a transient state.

Oversimplified

Additionally, some important aspects are left out from the simulations models that are designed so far. Two

key aspects are discussed in particular. Firstly, not all studies have included arriving aircraft. By omitting arriving aircraft, traffic density is substantially lower on the airport layout and thereby important dynamics are removed from the actual operation. This results in an inaccurate indication of the number of conflicts, the dimension of congestion and thereby ATC's workload.

The second aspect is related to the dynamics of aircraft. Both the LP and SA modelling techniques are solved from a central point of view and aircraft interaction and conflicts are avoided in advance. Consequently, both techniques do not take the autonomous behaviour of the aircraft agents into account. Also, both techniques cannot include uncertainty directly. Only by running many simulations with different settings, this aspect can be taken into account. However, the dynamic aspects and uncertain factors make it a challenge for ATC to guide aircraft effectively around the airport, as the mental planning of air traffic controllers has to adapt continuously due to uncertain factors as are mentioned in Subsection 2.1.4. Consequently, by excluding (some of) these factors, a large part of the reality is removed from the ground operation.

Incorporating actual operation

Only a few studies performed observations to get a clear understanding of the operations and procedures. Idris and Hansman [32] dedicated their research to observing and analysing the departure process. A couple of studies observed historical data to get an insight in the operation [31, 37, 64–66]. Others first created a model before performing field-tests [64, 66] or human-in-the-loop experiments [7, 43, 49] to find out whether their tool was an accurate representation of the actual operation. So for the few studies that did include the actual operation, mainly in the initial and final stage of the model development process. As a consequence of the fact that almost no studies validate their results with the actual ground operation, practical applicability of the results is still missing.

Most studies focus on obtaining results that are "optimal" in theory, instead of developing the model in a representative manner. Assessing the modelled operation with respect to the actual operation is undervalued, although it is a requirement to obtain results that have practical relevance. Reviewing literature has revealed the need to perform an empirical study parallel to the development stage of a model-based study. An empirical study that compares the simulated and actual operation on flight level, could lead to essential insights that can be used to improve the model that is designed in the model-based stage. So far, no study has been conducted at this level of detail.

Decentralised control

Most studies focus on designing tools that allow a decrease in the workload of ATC, either by reducing tasks due to automation, or supporting the controllers with systems to enrich operational knowledge [43, 64]. However, the air traffic controllers are still seen as centralised actors, responsible for many aircraft in a large area of the airport.

All studies using the ABM technique focus on decentralised control, in which the decision making process is distributed over many fictitious agents placed on each of the taxiway intersections. Although the concept of decentralised control seems promising, it has only been applied to arbitrary airport layouts. Additionally, all studies focus on the model's outcome, instead of the concept of decentralisation itself. Consequently, comparisons between local and global decision making are omitted, although they would allow interesting evaluations in terms of performance and behaviour.

Identified gap

Limited to no knowledge of the actual ground operation is taken into account when developing a model. Due to misunderstandings and oversimplifications, essential aspects like uncertainty and aircraft conflicts are left out of the model. No studies were found that perform empirical studies in parallel to the development process, thereby neglecting the opportunity to create a model from an operational perspective. Additionally, most studies focus on obtaining a theoretical optimum for the steady-state operation, without having obtained a detailed understanding of the actual operation under investigation. Consequently, the theoretical results cannot be translated into practice. An example is the novel technique of ABM, which enables the evaluation of decentralised control. So far, this concept has not been studied under representative conditions, to enable the assessment between local and global decision making. The overall focus should be placed on obtaining practically useful results. It is therefore required to develop a representative simulation model of an airport's ground operation, using an understanding of the actual operation as foundation. The next chapter describes how this research aims at filling part of this research gap.

3

Research Framework

The research gap, as identified in the previous chapter, is used to draw the problem statement, which is found in Section 3.1. Section 3.2 explains how this research can contribute to acquire knowledge to solve part of the problem. Additionally, the research objective and research questions are given. The scope of research and methodological steps to meet the research goal are presented in Section 3.3 and Section 3.4 respectively.

3.1. Problem Statement

The demand in air transportation is growing at an average speed of 3.7% per year [30]. Currently airports in Europe are having troubles to accommodate this rapid growth. Capacity problems at airports have been the cause of 14% of the by EUROCONTROL issued delays in October 2017 [17]. As expanding infrastructure is a costly and timely manner, alternative ways are explored to increase capacity. The focus is placed on improving the operation using the existing infrastructure. One of the key players in the operation is Air Traffic Control (ATC) and has become a common topic in research. Proposed concepts to ease the ATC's workload are by expanding the team or by creating automated system to enrich an air traffic controller with operational knowledge. However, both cases still rely on a centralised controller who is responsible for many aircraft and tasks in a large area of the airport. Due to the constrained processing ability of humans, only limited local information can be absorbed and used for future decisions, while still maintaining a safe operation. To structure taxi flows, ATC makes use of procedures, protocols and standard taxi directions. However, these operational standards limit ATC's capabilities to use the scarce resources efficiently and thereby solve an operational disturbance or conflict quickly.

Amsterdam Airport Schiphol (AAS) is one of the airports facing capacity issues [19], partly due to the regular occurrence of runway reconfigurations as explained in Subsection 2.1.1. Each of these runway reconfigurations disturbs the operation, as multiple new traffic streams have to be integrated in the existing traffic structure. As these traffic flows are potentially conflicting with the already existing traffic, ATC has difficulties to use the available runway capacity effectively. The operational standards of ATC limit its capabilities to solve the disturbance quickly. Consequently, the effects of the reconfiguration remain noticeable for a longer period of time causing operational delays to increase.

It is suggested that decentralised control is able to use the taxiway system in a more efficient manner, leading to an improvement in airport capacity [68]. Shifting the decision making process to a local level, enables local controllers to focus on solving congestion and conflicts in its vicinity. As these controllers can adapt quicker and more efficiently to a change in conditions, it is expected that decentralised control is able to better manage the dynamics of an airport's ground operation.

So far, very few studies focused on the concept of decentralised control. The available studies implemented decentralised control in a simulation model from a theoretical perspective, and stopped at the point where the feasibility was demonstrated for simple layouts. Since the validation, and assessments with respect to the actual operations, are omitted, the potential of decentralised control is still unknown.

3.2. Research Objective and Questions

This section describes how this research aims taking steps to improve the operational capacity of airports, by developing a simulation model from an operational perspective. The research objective is given in Subsection 3.2.1, followed by the research question and breakdown in sub questions as listed in Subsection 3.2.2.

3.2.1. Research Objective

From the previous section it is concluded that ATC has difficulties to efficiently guide aircraft along the airport during a transient state of operation. Instead of solving the ground operation from a centralised ATC point of view, it is possible to decentralise the responsibilities of controlling an aircraft. This so-called decentralised control is based on the idea that many local controllers (agents) make local decisions, instead of one air traffic controller dealing with many aircraft and conflicts simultaneously. Decentralised control is able to cope with uncertainty and transient state of operations, due to the fact that local agents can focus on solving local congestion and conflicts. Not only can these agents adapt quicker and more efficiently to changing conditions, they also possess the ability to use the taxiway infrastructure in a more flexible way. Examples are flexibility in the direction of taxiway usage, and flexibility in routing structure to make a shortcut or bypass a stationary aircraft.

However, it is noted that two aspects are missing in literature. Firstly, a case study of an actual airport, allowing decentralised control to be investigated under representative conditions. Secondly, no empirical studies have been performed to either validate, or assess, decentralised control with respect to centralised control.

Agent-Based Modelling (ABM) is a modelling technique that allows decentralised control to be implemented. Additionally, it is known for its abilities to cope with uncertainty and for its suitability to model socio-technical systems. However, the strengths of ABM heavily depend on the amount of time that is spend on understanding the operation at stake. Consequently, transforming the dynamics of a ground operation into a theoretical model requires the procedures, interactions and constraints to be well understood. Only in this way, the obtained theoretical knowledge can be used in practice, meaning that designing a realistic ground operation model is key aspect of this research.

An agent-based simulation model is to be designed, based on the concept of decentralised control. The model is developed from an operational perspective, instead of the commonly used optimisation point of view. This is done by supporting the model-based study with an empirical study. Observing the actual operation, as well as comparing the simulated operation with the actual operation, leads to valuable insights and conclusions, that help further development of the simulation model. By combining these two studies, the implementation of decentralised control is validated in an iterative manner.

A case study is applied to the ground operations of AAS, for two main reasons. First of all, access has been granted to a wide range of operational knowledge regarding this airport. Operational data of actual flights tracks at AAS has been made available for this research, allowing a detailed comparison between simulated and actual traffic to identify a difference in behaviour. By understanding the underlying cause of this difference, it is possible to iteratively improve the model. Additionally, this research has benefited from the the expertise of aviation consulting company To70 and its network of (former) air traffic controllers. By having in-depth discussions, a deep understanding of the ground operation from ATC's point of view has been obtained.

Secondly, it is explained in Subsection 2.1.1 that a transient state of operation occurs frequently at AAS, due to the large number of runway reconfigurations. It is known that the transient state, caused by this operational disturbance, results in large operational challenges for ATC. Using AAS as a case study gives this research the opportunity to explore the boundaries of decentralised control.

The research objective is formulated as follows:

"the objective is to create an understanding of the principles and mechanisms of a decentralised air traffic control, by systematically comparing the emergent behaviour of an agent-based model to the actual ground operation."

3.2.2. Research Questions

This subsection presents the research question that has to be answered in order to meet the earlier mentioned research objective. Furthermore, a breakdown of this research question in terms of key questions is given.

The research is combining two types of studies. An empirical study is performed to evaluate the emergent behaviour of decentralised control with respect to the actual operation, as executed by ATC at AAS. Within this study, a model-based study is performed in which an agent-based model is developed. It is therefore that two research questions are defined.

The development process of the agent-based model is closely related to the first research question:

1. *How could decentralised air traffic control be implemented in an agent-based model to simulate actual ground operations?*

To answer this first research question, multiple sub questions have been identified:

- 1a. *Which agents, corresponding characteristics and interactions, and procedures are involved in an airport's ground operation?*
- 1b. *To what extent can the actual operational procedures and dynamics be integrated in the model?*
- 1c. *What are the requirements to implement decentralised air traffic control in the model?*

The evaluation of the emergent behaviour of decentralised control is closely related to the second research question:

2. *Up to what extent does the emerging behaviour of decentralised air traffic control match the behaviour of centralised air traffic control at AAS?*

To answer this second research question, multiple sub questions have been identified:

- 2a. *To what extent do the dynamics of the simulation model match the actual dynamics at AAS?*
- 2b. *To what extent do the routing strategies of decentralised air traffic control match the one of centralised air traffic control at AAS?*

3.3. Research Scope

This research focuses on developing a simulation model from an operational perspective. Firstly, actual operations are observed and understood before starting the development phase. Secondly, the concept of developing a complete model before validating it with an actual operation, is inverted; a case study is performed while simulating the actual operation. Based on the issues and observations with respect to the actual operation, knowledge is gained on new requirements to improve the simulation model. As this is a new direction within developing a model to simulate an airport's ground operations, not all aspects of the process can be taken into account. This section gives an outline of the research scope of this study.

In this investigation, the airport's ground operation is limited to the taxiway infrastructure. This means that only the segment between leaving the apron and arriving at the runway is included for departing aircraft and vice versa for arriving ones. Consequently, the apron operation is excluded, as this domain is a complex process in itself and no data is available. This also means that it is assumed that the gate is always available, and that no towing takes place.

This research focuses on the ground operation at AAS, using the actual airport layout, a derived flight schedule, and an understanding of the operation as starting point to design the agent-based model. The model development depends on assessment with respect to the actual ground operation at AAS, using the strategies as applied by AAS's ATC as reference.

3.4. Methodological Steps

The research question and related key questions are composed in the previous section. This section gives an overview of how the different questions are related to one another by presenting the methodology in a simple research framework.

The research framework is presented in Figure 3.1 and consists of four main phases. Subsection 3.2.1 stated the research objective and explained the essence of approaching this study from an operational point of view. It is why observations of, and comparison with, the actual operation is performed along the complete process of this research.

1. Obtain understanding actual ground operation;
2. Obtain routes as travelled in actual operation;
3. Develop agent-based simulation model;
4. Evaluate performance decentralised control.

Phase 1

The first phase of the research focuses on obtaining a clear understanding of the current ground operation AAS. This includes ATC procedures, execution of runway reconfiguration and the applied routing strategies. The state-of-the-art in research is found to get an idea of the work of others, the modelling techniques used and the knowledge gained from these studies. Summaries of the ground operation and literature review can be found in Section 2.1 and Section 2.2 respectively. Additionally, Key Performance Indicators (KPIs) are selected as assessment criteria to determine the performance of the actual operations, as well as the simulated operations, in later phases of the research.

Phase 2

The second phase comprises the development of an Automatic Dependent Surveillance-Broadcast (ADS-B) tool. Each ADS-B data point transmitted by an aircraft, consists of the location of that aircraft. However, filtering and processing of the data is required to work with realistic and useful data. Other steps involve determining the actually travelled ground path, and performing interpolation, such that it is known where each aircraft is located in time. A final step is to draw the flight schedules in terms of starting time, origin and destination from these sets of data, which is used as input for the simulation model. The processed data is used in later stages to compare simulated and actual ground tracks in order to determine the model's performance. An elaboration of these steps is found in Chapter 4.

Phase 3

The simulation model is developed in phase 3, using the gained operational knowledge from the first phase as input. The goal of this phase is to develop an agent-based model that enables decentralised control on AAS's layout, using an actual flight schedule as input. The simulated operation is continuously monitored with respect to the actual operation. The formalisation of new requirements, or observations of restricted behaviour, lead to the implementation of new features or types of agents. A schematic overview of the development phase is shown in Figure 3.2. Depending on the type of feature that has to be implemented or improved, it might be necessary to temporarily switch to a more simple airport layout to focus on a particular traffic scenario. A description of the developed model is given in Chapter 5.

By conducting this empirical study, both verification and validation are performed: prohibited and unsatisfactory behaviour is removed from the model. This empirical approach to develop the agent-based model provides the opportunity to compare the simulated and actual operation on flight level in real time. This unique perspective allows the researcher to acquire an understanding of decentralised control, as well as the modelling technique. Knowledge is gained regarding the steps that are required to make decentralised air traffic control work for the complexity of an actual ground operation and layout.

Phase 4

The fourth phase is related to running multiple days of actual traffic scenarios. Validation is done to determine how well the model is able to simulate ground operations. By comparing the similarities and differences between the simulated and actual operations, the emergent behaviour of decentralised control is assessed with respect to centralised control. Besides, a sensitivity analysis is performed to determine the robustness of the model, and to test whether the selected operational parameters correspond to reality. This phase is covered in Chapter 6.

This chapter explained how this research aims at developing a simulation model from an operational point of view. By iteratively comparing the simulated ground operation with the actual operation, knowledge is gathered regarding the requirements and effects of implementing distributed air traffic control. Additionally the research framework is presented. A summary of the first phase is given in Chapter 2. The second phase of

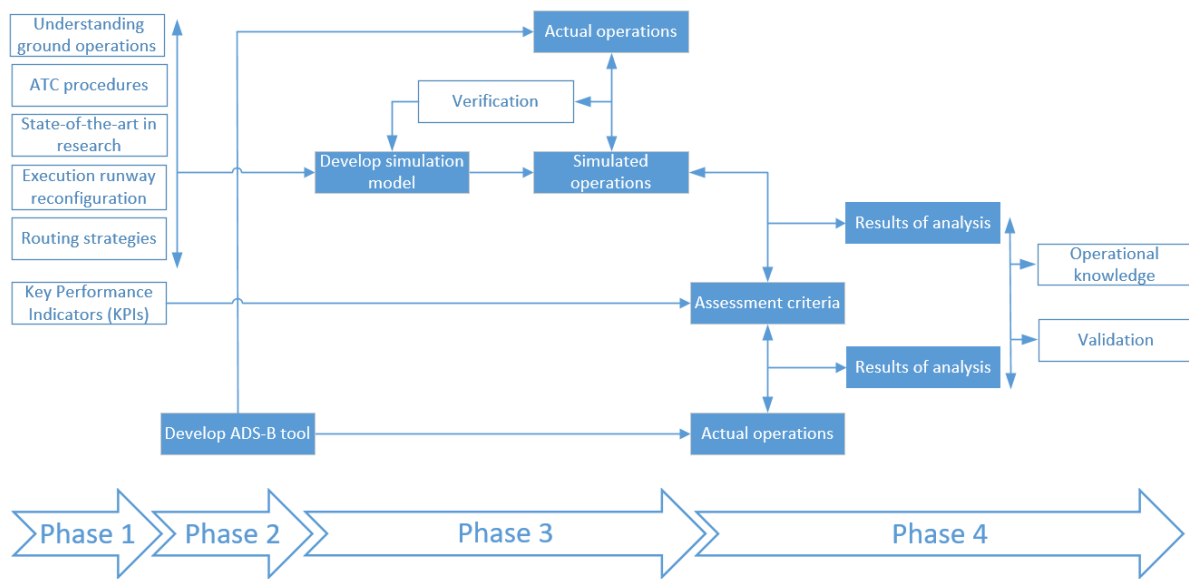


Figure 3.1: Research framework

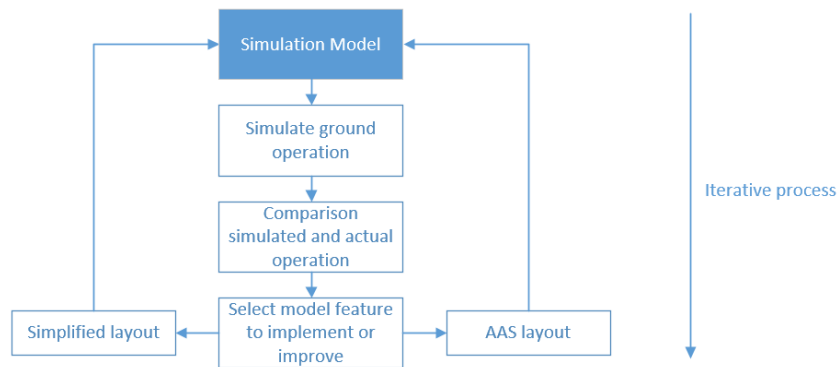


Figure 3.2: Iterative process to develop the simulation model

this framework is related to developing the tool to process actual ADS-B data, which is explained in detail in the next chapter.

4

Data Processing

The development of the simulation model is highly dependent on data from the actual operation. In order to possess high quality data which could be used to draw conclusions, it is of uttermost importance that this data is processed, filtered and analysed. Section 4.1 discusses the data sources that are used as input for this research. It is explained in Section 4.2 how Amsterdam Airport Schiphol (AAS)'s layout is processed to serve as airport layout for the model. A description of the steps that are taken to transform Automatic Dependent Surveillance-Broadcast (ADS-B) data into ground tracks and into a flight schedule is given in Section 4.3.

4.1. Data Sources

This section explains the different data sources used and why they are of importance for this research. Additionally, a summary is given with the content of each data file.

Airport layout

Two data sets are used to obtain the taxiway layout of AAS: the coordinates of taxiway points, and the taxiway segments that connect these points. The former file consists of 1113 data points, their ID, and their geographical coordinates in latitude and longitude. The latter file includes 1249 taxiway segments, labelled by their ID. Each segment is characterised by the two data points it connects and the maximum speed that is allowed on that particular segment. Both data sets are originating from official studies analysing the effects of expanding AAS's infrastructure, assuring the high quality of data.

ADS-B data

An aircraft equipped with a Global Navigation Satellite System (GNSS) receiver is able to determine its location and speed from GNSS signals. The derived position (including altitude), as well as ground speed, heading and identification code, are broadcast by an aircraft's ADS-B transmitter. This information is captured by ADS-B ground stations, and other aircraft, to determine the position of an aircraft [58].

By collaborating with aviation consulting company To70's ADS-B, access is provided to a database, which consists of data points of flights and ground movements at AAS. Combining data points corresponding to a certain flight results in a trajectory, which is useful to gain insights and characteristics of the ground tracks as travelled in the actual ground operation. However, it should be noted that not all ground tracks comprise the full path from the apron to the runway or vice versa. This is the consequence of aircraft that turn on their ADS-B transponder while they have already left the apron, or when the transponder is turned off too early. Also, there is a spot on AAS's taxiway infrastructure which has no coverage of an ADS-B receiver collecting the data. Erroneous data is included in the model as well, due to incorrect determination of the location, data transmitting, and data storing.

Operational expertise

A third set of data sources that proved to be of importance is related to acquiring an understanding of routing strategies and decisions as applied by Air Traffic Control (ATC) at AAS. *Flightradar24* is observed to determine routing strategies. Knowledge of how ATC solves conflicts is acquired based on conversations between pilot and ATC, as can be heard via *LiveATC*, as well as interviews conducted with (former) air traffic controllers and

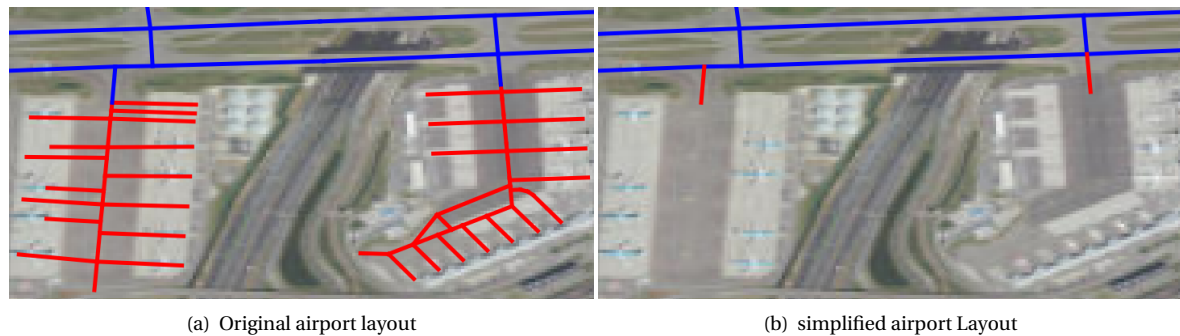


Figure 4.1: Effect of replacing an apron by an entry/exit node

experts. These insights are used to complement the ADS-B ground tracks that are incomplete. More information regarding this matter can be found in Subsection 4.3.2.

Runway usage

Besides, a database is present in which information regarding runway usage at AAS is stored. This data is originating from <https://www.lvn.nl/airtraffic>. Based on this data, it can be determined when a runway re-configuration takes place, which is of interest when analysing the simulated and actual operation. Although this website should have an update frequency of 5 minutes, it is found that some points in time were missing. As a result, the time window between two points can be up to 30 minutes. Also, there could be a delay between the moment ATC (de)activates a runway and the moment the first (or last) aircraft uses it. According to procedures, a departure runway first has to be activated before an aircraft can receive its push back clearance. Consequently, it takes both push back time and taxi time before the first aircraft takes off from the newly activated runway.

4.2. Create Airport Layout

Section 4.1 explains that the taxiway infrastructure of AAS includes 1113 taxiway points (so-called nodes) and 1249 taxiway segments linking these nodes together. A few actions have been applied to this layout to make it suitable for the simulation model, which are described in this section.

Apron removal

The apron areas are removed as it is explained in Section 3.3 that this research only focuses on the ground operation that takes place at the taxiways. This means that each of the aprons consisting of one or multiple gates and apron segments are removed. The nodes that connect the apron with the taxiway system acts as an entry and exit point of that specific apron. An example is given in Figure 4.1. Blue segments represent the taxiways, while red segments represent the apron segments.

Simplify layout

It is decided that the layout has to be simplified before using it in the simulator. Each taxiway point represents an agent in the agent-based model. This means that each level of detail in the taxiway layout negatively impacts the computational time of the simulator. It is therefore decided to simplify the layout as follows:

- Some "straight" segments consist of multiple taxiway points, to include the smallest detail of deviation in segment heading. As this level of detail does not add any noticeable accuracy to the model, the taxiway points in between are removed. An example of this simplification is given in Figure 4.2. The yellow dots represent the taxiway points.
- Some corners consist of multiple points to create a smooth turn. Again, as this level detail does not outweigh the increase in computational time, it is decided that corners are replaced by a single node. The effect of doing so is shown in Figure 4.2.
- At some parts on the airport, two intersections are located next to each other within a small distance. It is decided to merge these two intersections together, as can be seen in Figure 4.3.
- Parts of the taxiway infrastructure that are not (allowed to be) used are removed from the layout. This is done to make sure that the simulation model would not guide aircraft along these segments.

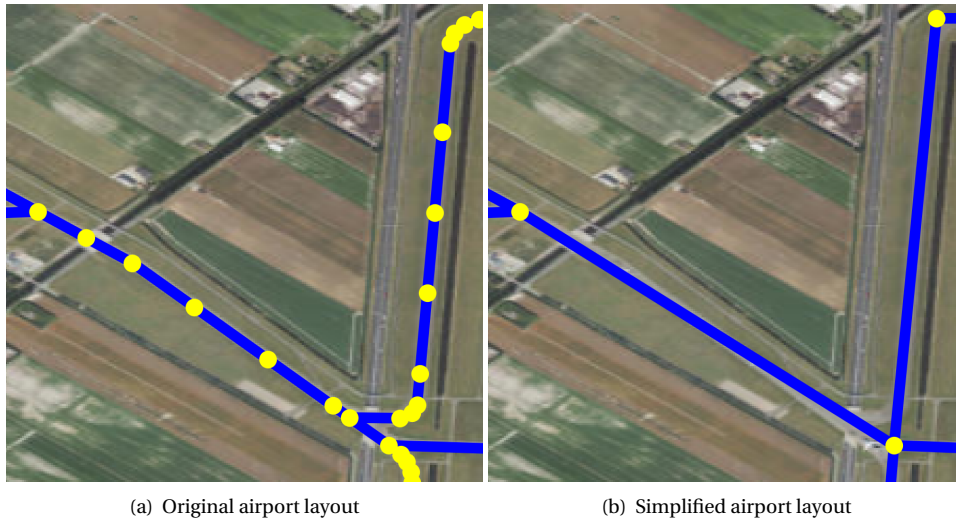


Figure 4.2: Effect of removing nodes in between a long segment and grouping complex corners to one node

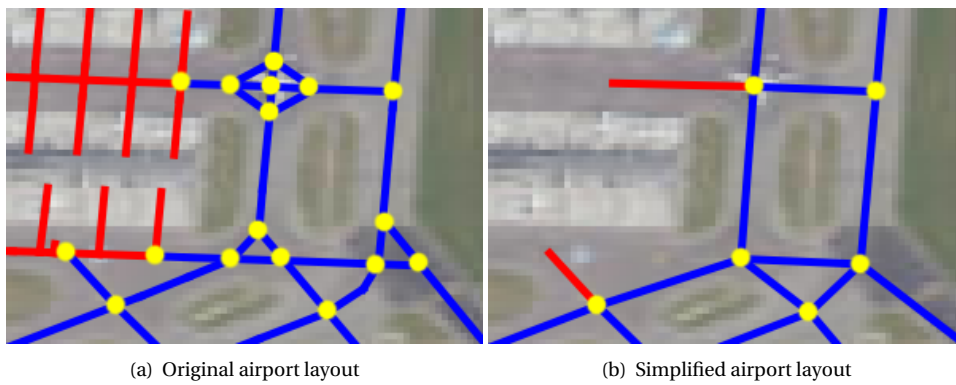


Figure 4.3: Effect of merging a cluster of nodes to one node

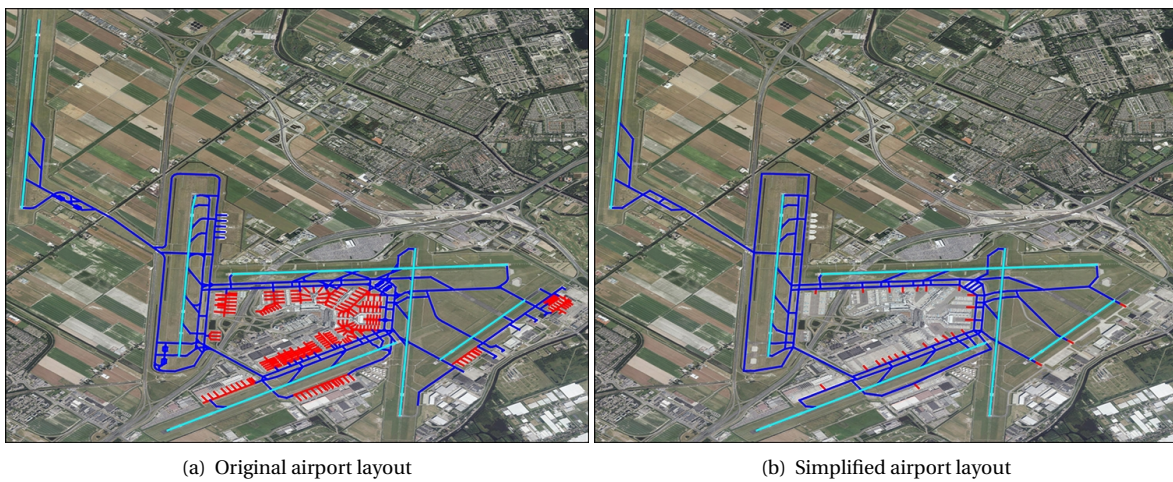


Figure 4.4: Effect of simplifying the airport layout

As a consequence of these simplification, the resulting layout consists of 218 taxiway points and 266 taxiway segments. Insights in the impact of simplification are shown in Figure 4.4. The light blue segments represent runways, while blue and red segments are taxiways and apron segments respectively. From this figure it can be concluded that a large number of nodes and segments are removed, without taking away the layout of AAS.

The resulting layout data is used to define AAS's layout. Apart from an identification number and Cartesian

coordinates, additional information is manually added to each of the data points:

- The type of node: either an apron entry/exit, an intersection point, a stop bar or a runway entry/exit.
- Both stop bars and runways receive an identification number of the runway it relates to, where each runway at AAS is given a number from 1 to 6.
- A Runway Mode of Operation (RMO) is characterised by the direction of usage, and whether it is used for arrivals or departures. Depending on the RMO, some stop bars have to make sure that aircraft can no longer cross the adjacent segment. This can be due to a conflict between the taxiing aircraft and the aircraft that is either landing or taking off, or due to interference with the runway's Instrument Landing System (ILS). Each stop bar possesses the information if, and for which RMO it has to prohibit an aircraft from using the next segment.

4.3. Process ADS-B Data

To compare output of the simulation model with the actual operation, it is required to obtain an actual flight schedule, as well as insights of the actual ground tracks. This section describes the development of a tool to transform the individual ADS-B data points, to a set of travelled ground tracks and a flight schedule. The data processing is divided into four steps:

- Filter and clean ADS-B data
- Determine travelled ground path
- Interpolate to obtain location at each point in time
- Derive flight schedule

These four steps are covered in Subsection 4.3.1 until Subsection 4.3.4. The limitation of the ADS-B tool are mentioned in Subsection 4.3.5.

A case study is applied to the ground operations at AAS, which is explained in detail in chapter 6. The first 14 days of May 2016 (1st of May - 14th of May 2016) are used as input for this case study. The overall data quality as mentioned in this section refers to the ADS-B data of these 14 days.

4.3.1. Filtering and Cleaning

Access has been granted to a large database, consisting of ADS-B data points. This database covers more data than AAS only. Additionally, when working with actual data, it is expected that erroneous data is present in addition to the useful data. This means filtering and data cleaning is required to obtain the correct data. Both aspects are covered in this subsection.

The database used for this research consists of multiple tables, all linked together via specific properties as can be seen in Figure 4.5. Filters are applied on different tables, to create a set of flights of interest. A few remarks have to be made before further explanations regarding filtering are provided:

- Apart from aircraft, also airport ground vehicles transmit ADS-B signals. To make identification possible, each of the mode-s transmitters has a unique 24-bit address. This bit address is often represented as six hexadecimal characters (hexcode). A list is made of all hexadecimal addresses that have been in the air at least once, such that only aircraft tracks can be selected.
- A commonly used ADS-B transponder is the mode-s transmitter. Each ADS-B data point emitted by this transmitter is characterised by an International Civil Aviation Organization (ICAO) type code. This type code is a number between 0 and 31 and provides information regarding the accuracy, and position type like ground or airborne [34]. This breakdown is of interest to only select ground track data points.
- An algorithm that fills the database with the most recent data tries to classify a flight into multiple segments, namely *departing*, *cruise*, *arriving*. This classification depends on multiple conditions, including (a change in) altitude.
- The ADS-B receiver does not have full coverage of AAS's layout, resulting in blind spots from which only very limited signals are captured. When plotting a heat map of the ADS-B data points of the first two weeks of May 2016 as is done in Figure 4.6, it can clearly be seen that only a few data points are traced in the area southeast of the terminals.
- If a set of ADS-B points is tracked and the time interval between two points is above a specified limit, this sub track is cutoff. In case a track of data points is found which does not become airborne, this track is labelled *ground*. Examples are tracks from ground vehicles. Normally, the flight path of a departing

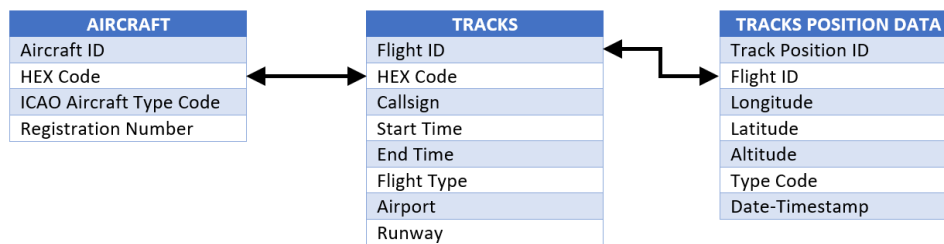


Figure 4.5: Links between database tables



Figure 4.6: Heat map based on ADS-B data points

aircraft ranges from its apron up to cruise level, and is labelled *departing*. However, as is found along the development of the ADS-B tool, it is likely that departing aircraft which pass the blind spots of the ADS-B receiver, have two ground paths: a *ground* segment between the aircraft's gate and the blind spot, and one *departing* track between the blind spot and cruise level. The same might happen for arriving flights. Initially, only *arriving* and *departing* segments were selected for further processing. However, vital information like the used apron could be missing if the *ground* segment is not included. It is therefore decided to merge a flight's *ground* segment with the *departing* or *arriving* segment of the same aircraft, if the time difference between the two segments is less than 10 minutes.

- Another algorithm places gates at each of the runways. By connecting the ADS-B data points of each flight, it is tested through which of the runway gates a flight has passed. The corresponding runway is selected as runway used, and is stored for each flight.

An overview of the steps that have been taken to filter the relevant tracks is shown in Figure 4.7. This data is cleaned and processed according to the steps as outlined in Figure 4.8. Information below explains in more detail why and how each of the steps are performed.

Data filtering

Before the actual flight data processing can start, multiple filters have to be set to retrieve the flight data of interest. It has to be determined which of the data points correspond to aircraft, in order to remove ground vehicles. The earlier mentioned overview of all hexadecimal addresses that have been in the air is used to filter relevant codes.

The ADS-B database includes information of flights that were in range of the linked ADS-B receivers located around the globe, and are stored for multiple years. To only select data of flights that are both useful and meaningful to determine the path travelled, a few filters have to be applied:

- Flights with airport AAS as origin or destination.
- Flights within a specific period of time. It is possible to set conditions whether the flight's starting time is within a specified time window, to focus on a particular traffic scenario or runway reconfiguration.
- Only ground positions are selected, based on the ICAO type code for which the height is zero.

For the first two weeks of May 2016, this resulted in 13,181 unique arriving and departing flights.

The following cleaning technique is also related to checking the flight's credibility. As mentioned earlier, it is found that looking only at the flight segments *arriving* and *departing* does not cover the full ground path of an aircraft. It is checked whether a *ground* track can be merged with the flight's *arriving* or *departing* track. Merging is done in case the hexcode of the aircraft matches, and the time difference between the last data point of the *arriving* segment and first point of the *ground* segment is less than 10 minutes. For departures, it is the difference between the *ground* segment and *departing segment*. For 2,832 flights, a *ground* segment could successfully be merged with the *arriving* or *departing* segment.

Two more filters have been set:

- Flights of which the ground path lasts between 60 seconds and 3600 seconds.
- Flights that have at least 20 data points.

A total of 796 flights (6%) did not satisfy both conditions. The data processing part continues with the remaining 12,385 flights and over 3 million ADS-B data points.

Remove erroneous data

A boundary has been drawn around AAS's taxiway infrastructure, as is shown in Figure 4.9. For each flight and its corresponding data points, it is checked whether it is inside this boundary. Potential causes of points outside this boundary is due to system or measuring errors, or incorrect labelling in terms of type code or flight segment. In case less than 20% of a flight's data points are labelled incorrect, it is decided that only these erroneous points are removed. Once again it is checked whether the flight satisfies the condition of having at least 20 useful data points. If more than 20% of the data points are labelled incorrect, the complete flight is removed from the data set, as the credibility is doubted. An example of a flight for which a point has to be removed is shown in Figure 4.11. A total of 38 individual data points have been removed for 36 unique flights. Also, one complete flight is removed, since more than 20% of the data points were outside the boundary.

Determine a flight's apron and runway

The need to match the *ground* segment to the *arriving* and *departing* segments was raised, as the flight segments did not cover the full ground path. Figure 4.10 shows an example of an *arriving* segment (in yellow) that is merged with its corresponding *ground* segment (in orange). As can be seen, thanks to this *ground* segment, the apron of this flight can be determined.

Still, it is found that 5,194 flights (42%) does not start or end on an apron. As the database does not cover information regarding the gate or apron, this means no full ground track can be determined as long as no apron is assigned. Consequently, it is decided to do an analysis on the most-used apron per aircraft hexcode. It has been assumed that the missing aprons can be replaced by the aircraft most used gate, for which historic data of 2016 is used. Only for the hexcodes for which no gate could be found in historical data, a manual apron allocation has been done for 13 flights, based on the airline and the aircraft type. Historical data was missing for 15 out of the 5,194 flights. It has been decided to remove these flights, as a randomly selection would be too arbitrary.

It is mentioned earlier that a flight's runway is selected based on the runway gates it crosses. It is tested whether the first data point of an arriving flight or last data point of a departing flight is within 100 metres of the centre line of the runway. The difference is larger than 100 metres for 337 flights. For 3 flights, a different runway is at least 200 metres closer than the runway specified. Based on visual inspection, these tracks are removed as no clear conclusion about the correct runway could be drawn. It also occurred once that the flight tracks does not cross any of the runway gates. It is then decided to allocate the runway that is closest to the track's endpoint (last data point departure aircraft, first data point arriving aircraft).

Get data points between apron and runway

The final step in the data cleaning process is to select the relevant taxiway links of the ground path. As an aircraft movement is only going to be simulated between the apron exit and runway entry, or vice versa, this part has to be cut out off the full ground path. For departing aircraft, if data points are found on the apron area, the last data point in this area is selected. Otherwise the first data point is chosen. For the runway element, either the first point within a box of 100 metres around the runway's centre line is selected, or the final point of the ground path in case no point is found in the runway box. For arriving aircraft, a similar strategy is applied.

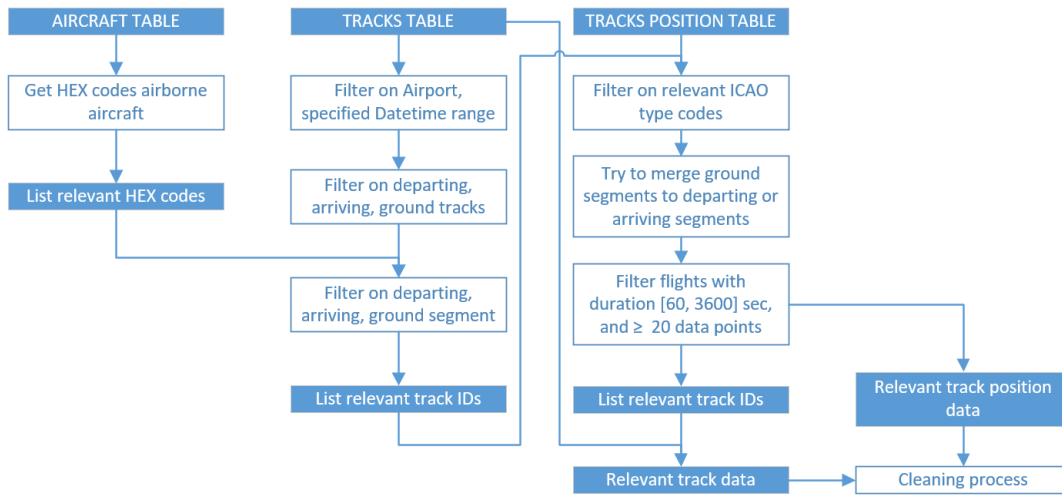


Figure 4.7: Steps required to filter track data

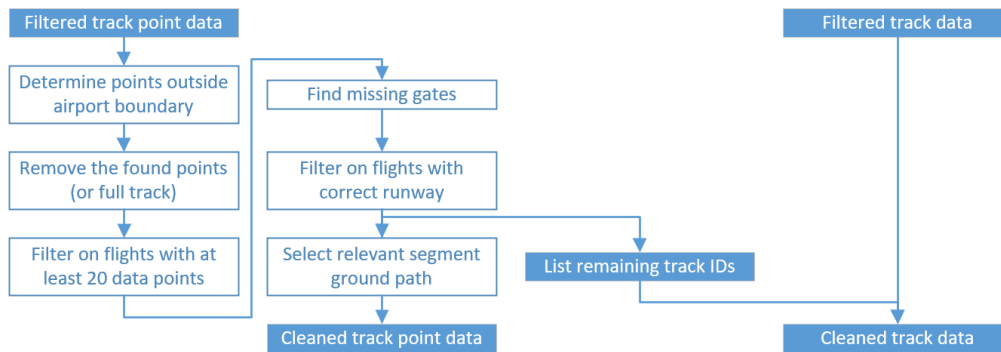


Figure 4.8: Steps required to clean and process track data



Figure 4.9: Airport boundary

The filtering and cleaning process resulted in 12,366 flights with 204 data points on average. In Now that the interesting route segment is both selected and filtered, the travelled route can be determined.

4.3.2. Determine ground path from location points

The cleaned set of ADS-B points is used in combination with the airport layout, as created in Section 4.2, to determine the ground paths travelled. The route travelled cannot simply be read from the data points as a result of:

- The simplified airport layout, meaning that the location of taxiways deviates from the measured position of the aircraft.
- Deviations between the actual position, and measured position of the aircraft.
- Incomplete ADS-B data, due to ADS-B transmitters that were turned off or the loss of the ADS-B receiver which does not have full coverage.

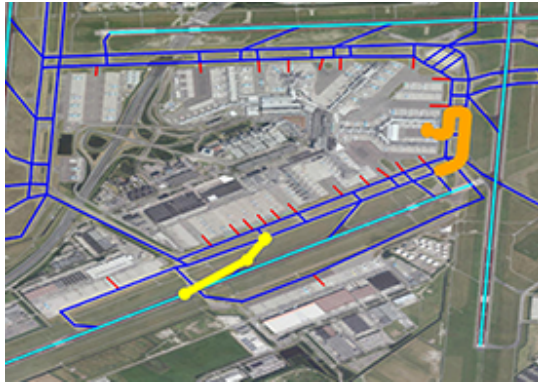


Figure 4.10: Merged arriving segment and ground segment

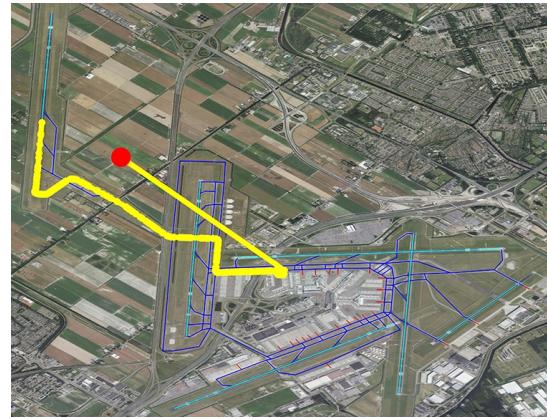


Figure 4.11: Points outside the airport boundary

Due to the uncertainty in the determination of the path travelled, it is decided to use Dijkstra's algorithm to obtain the most likely ground path. The airport layout is transformed into a network graph, for which the initial weight w_i is set to the distance of each taxiway segment i , s_i . The network graph is tweaked for each flight individually, in order to give preference to the most likely travelled segments. The origin and destination are based on the apron and runway points used.

Preference most likely travelled segments

When determining a flight's ground path, preference is given to the taxiway segments that are closest to the ADS-B points. This preference is given by lowering the weights of segments that are likely to be travelled. The new weight of each segment i is calculated using Equation (4.1). Variable $x_{\text{closest},i}$ indicates the number of times segment i was closest to a data point. Consequently, the weight of a segment is inversely proportional to the number of times it is closest to an ADS-B point.

$$w_i = \frac{s_i}{x_{\text{closest},i} + 1} \quad (4.1)$$

An elaboration on how the closest segment is determined for each ADS-B point is provided in Appendix A. This step results in a flight specific network graph, with weight reductions that are applied to segments that are most likely to be travelled.

Estimation of origin and destination

Before Dijkstra's algorithm can be applied on the created network graphs, it is required to determine the exact origin and destination of each flight. For a departing flight this is the apron exit node and runway entry node, while it is the runway exit node and apron entry node for an arriving flight. The apron node is found by selecting the closest node of that specific apron to either the first or last point of a departing and arriving flight respectively. The same holds for determining the runway node. This runway node that is closest to the last point or first point of a departing and arriving flight respectively, is selected.

Apply Dijkstra's algorithm

All aspects of having an adjusted network graph, an origin and a destination are in place. Therefore Dijkstra's algorithm is used to determine the expected ground path travelled by each aircraft. All calculated ground paths are saved automatically, as well as a figure that shows the ground path trajectory on the airport layout.

Manual corrections

Each figure of the determined route is inspected manually, to determine whether the assigned trajectory is in line with the locations as transmitted by the aircraft. Due to a lack of data, Dijkstra's algorithm believes it is "best" to make a shortcut. Examples are given in Subfigure 4.12(a) and Subfigure 4.13(a). The ADS-B receiver installed at AAS does not cover 100% of the airport, meaning that there are blind spots in the southeastern region of the airport as shown in Figure 4.6. Due to the limited coverage in this area, there is almost no weight reduction in the network graph to determine the most likely route. Consequently, it occurred on a regular basis that Dijkstra's algorithm resulted in a path that does not match the routing strategy as applied by AAS's ATC.

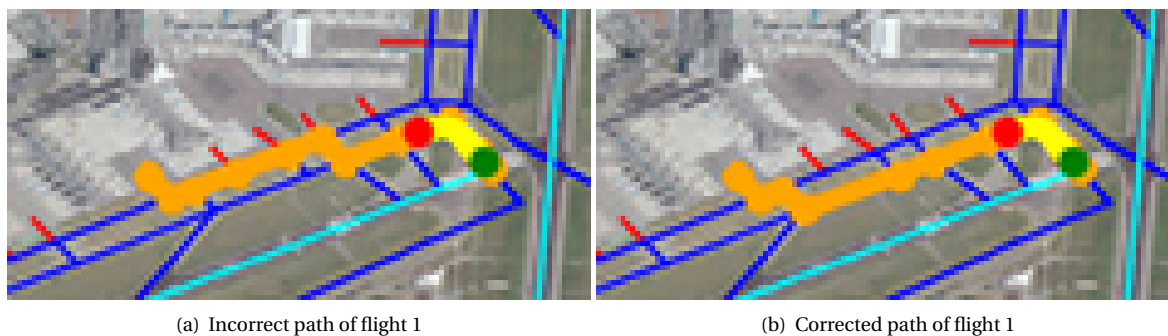


Figure 4.12: Corrections made to path of flight 1

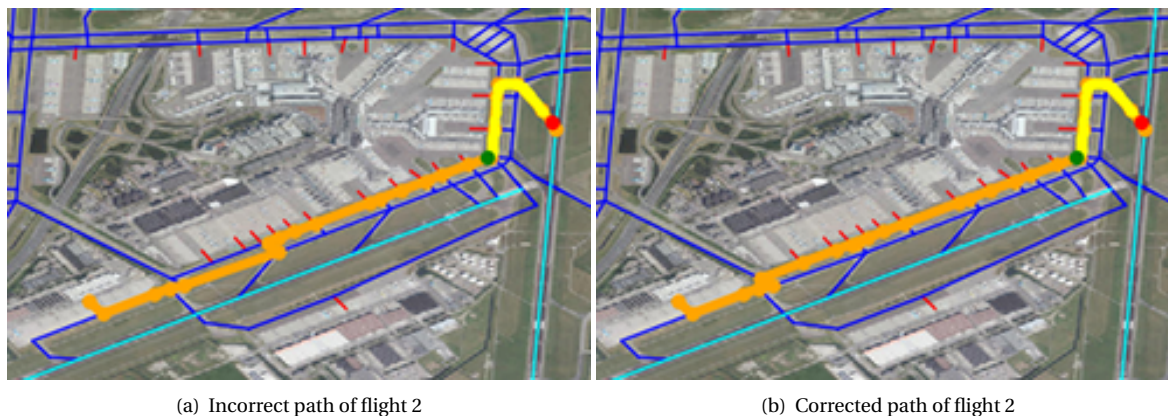


Figure 4.13: Corrections made to path of flight 2

When analysing the time stamps of a flight's data points, it is found that they are not collected at a fixed time interval. These time stamps are stored in the dataframe with a level of accuracy of 1 second. Consequently, the time difference between two data points could range from 0 seconds up to minutes. It is the latter of these two values that could cause issues. Due to a lack of data points along part of the path, there is almost no weight reduction in the network graph to determine the most likely route.

The acquired understanding of routing strategies as applied by ATC is applied to improve or correct the paths found. For the period 1st – 14th of May 2016, corrections had to be made for around 30% of the ground paths. Examples of corrections are found in Figure 4.12 and Figure 4.13. For the former of the two flights, it is concluded that ATC would guide this aircraft directly to the outer taxiway (taxiway Bravo, see Figure 2.2), before lining up on the runway. For the second flight, the path is changed in such a way that the aircraft follows taxiway Alpha till the end, instead of switching to taxiway Bravo halfway. This decision is based on operational knowledge regarding the routing strategies. Additional changes that had to be made regularly are related to flights where the apron entry that is closest to the last ADS-B data point does not correspond to the actual apron entry used, as seen in Figure 4.14.

The filtered and cleaned ADS-B points are used as input, to determine the travelled ground path. These ADS-B points are mapped to the closest taxiway segment, which then get a reduced weight in order to stimulate Dijkstra's algorithm to include these weights in the most likely travelled route. This route is used to estimate the aircraft's location at each point in time.

4.3.3. Interpolation to obtain missing data

As the simulation model is going to simulate ground operations at a time step of 1s, it is useful to make sure also the actual data is at the accuracy level of 1 second. It is noted before that the current time difference between ADS-B points can range from 0 seconds up to minutes. Therefore it is required to interpolate between the already known locations along the found path.

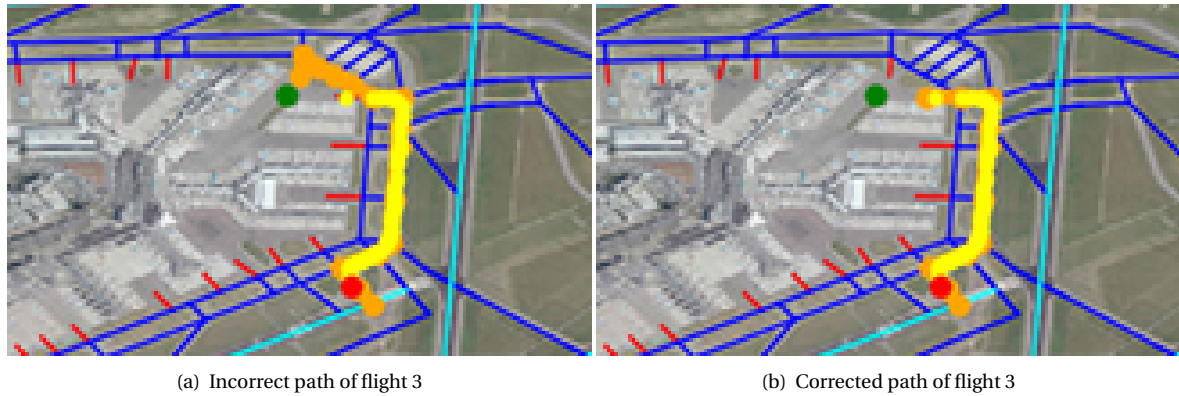


Figure 4.14: Corrections made to path of flight 3

A few steps are required before interpolation can be applied:

- Each of the ADS-B data points is projected onto the route, based on the shortest orthogonal distance. In case no taxiway segment along the route is found within a distance of 250 metres, this point is assumed to be erroneous and thereby removed.
- The ADS-B points have a time stamp with a 1 second accuracy. It occurs that multiple ADS-B data points are found with a similar time stamp. Attempts have been made to include as much real data as possible, by allowing small time shifts. A description of the applied protocol is given in Appendix A. Nine percent of the data points could be shifted. Twelve percent of the data points had to be removed as there was no possibility to shift the similar time stamps.
- In a number of cases it occurred that a point projection on the route resulted in a location that is forward in time, but backwards along the route, with respect to the previous data point. This occurs in cases of e.g. an incorrect projection, or a push back at the departure gate. The assumption is made that an aircraft can only move forward along the path. Details of the procedure to remove this inappropriate behaviour can be found in Appendix A.

Based on the remaining data points, interpolation is performed. In case the first node is more than 1 metre away from the origin, interpolation is done between the origin and first node using the speed $v = 30$ knots. This speed value is coming from the maximum speed v_{max} , as set in the simulations. Also, in case the distance between the last node and destination is more than 1 metre, interpolation is performed with the same parameters. For all other nodes in between, interpolation is performed to determine the locations at the missing time stamps.

On average, 279 points have been added per flight. This brings the total number of data points to 5.6 million, which is 458 points on average per flight.

4.3.4. Derive Flight Schedule

The final step in processing the ADS-B data is related to obtaining the flight schedule. This flight schedule serves as input for the simulation model, and includes an aircraft ID, an origin, a destination, a start time and information whether a flight is arriving or departing. The origin and destination of each flight have already been determined in Subsection 4.3.1. Therefore, the start time of each flight is determined by selecting the time stamp corresponding to the first data point in the set of flight data, composed of both actual and interpolated data. All aircraft are ordered on start time, and receive an ID accordingly. All tools are written in the programming language *Python*, which has the characteristic to start counting at 0. Therefore, the first flight has aircraft ID 0, while the last flight in a set of N flights has aircraft ID $N - 1$. Additionally, the column whether each flight is arriving or departing is added.

The result is stored in a *.txt* file, which is imported in the simulation model to create and spawn aircraft agents according to this schedule. More information regarding the model can be found in Chapter 5.

4.3.5. Limitations of the ADS-B Tool

It is always a challenge to work with actual data, and the development of the described ADS-B tool is no exception. While analysing the actual flights and the created ground paths, a few limitations are found.

Manual corrections

The data source of the ADS-B data, the ADS-B receiver, does not have full coverage of AAS airport. As a result, it is found that the limited number of ADS-B data points make it sometimes difficult to obtain the actual ground path. The blind spots in the routes have been adapted based on an understanding of ATC's routing strategies, but it is unknown whether the right corrections have been made. The same aspect applies for flights which have a large time gap in between two data points. It is difficult to determine the most likely route in this part, as almost no weight reduction is performed in the network graph. As a result, corrections are made based on gained operational insights. Additionally, due to the fact that the ADS-B path was not always complete from apron to runway or vice versa, assumptions have to be made regarding the apron, as well as the runway entry or exit. Therefore, ground paths are created according to the best available knowledge, but no guarantees can be made whether this has been the actual path travelled.

Odd flight behaviour

Some flights included strange behaviour in their path, which are labelled as problematic flights by the tool. Examples are shown in Figure 4.15, where the yellow dots are ADS-B data (red is first data point, green is the last data point) and orange is the found ground path. In Subfigure 4.15(a), an aircraft travels in a circle before travelling towards the runway. The operational explanation is that due to the fact that this specific apron has only one point to both enter and exit the apron, the aircraft has to make room for another aircraft to enter the apron. However, for the departure sequence it is required to get behind an other aircraft, and therefore it has to make a detour on the taxiway layout. Subfigure 4.15(b) shows a departing flight which lines up on the runway, leaves on the next runway exit before lining up once again. Some flight paths are adapted manually, while 10 flights are removed from the data set as no clear path could be determined.

Simplified airport layout

Another issue found is due to simplifications in the layout of AAS, as explained in Section 4.2. As a consequence, the actual route could not be made in the simplified layout. An example is shown in Figure 4.16. Instead, the best possible path is selected in orange, even if it means that interpolation resulted in very high speeds that are impossible on a taxiway layout.

Stationary aircraft

Some gate stands at AAS require a departing aircraft to be pushed back upon one of the main taxiways. In practice this means that an aircraft is pushed back and holds stationary for some time at this location. However, the tool cannot differentiate between an aircraft that has to wait on the taxiway after receiving its push back, or that it has to wait in a runway queue soon after the aircraft left from its apron. This means that the ADS-B tool assumes the aircraft is already taxiing, as it has already left the apron area. Consequently, the start time of this aircraft is set at the moment the aircraft leaves the apron area, even if the aircraft is actually not taxiing yet.

Missing flight tracks

There is another big limitation of using ADS-B data, even though it is not linked to the designed ADS-B tool itself. It is found that around ADS-B data of 900 aircraft tracks is found on average for each day in May. However, the average number of flights per day in May 2016 has been 1,400 [25]. This difference is caused by aircraft that do not have an ADS-B transmitter. This transmitter is only mandatory for aircraft above a specified mass limit, and from June 2020 onwards [18]. The Fokker 70 is one of the aircraft which does not have an ADS-B transmitter. In 2015, KLM Cityhopper still had 19 Fokker 70s, which was brought back to 9 in 2016 [39]. As these aircraft fly frequently each day, this has a large impact on the number of missing ADS-B flight tracks.

Unfortunately, more relevant aircraft tracks are missing, namely the ones of aircraft that are being towed. Due to the fact that the take off and landing rights (so-called airport slots) are scarce at AAS, unfavourable combinations in terms of turn around time are issued. Whenever the Turnaround Time (TAT) between landing and take off is above a defined limit, aircraft could be towed to a parking location to make room for other aircraft to use the gate. During this towing, an aircraft does not transmit ADS-B signals, meaning that also these movements on the taxiways are unknown. Although the towing trucks do transmit their location, the quality, in terms of frequency and accuracy, is too low to use it. Towing an aircraft is a slower process, meaning a queue of aircraft is formed behind the towing truck, which creates some congestion and thereby increases the taxi time of these flights.

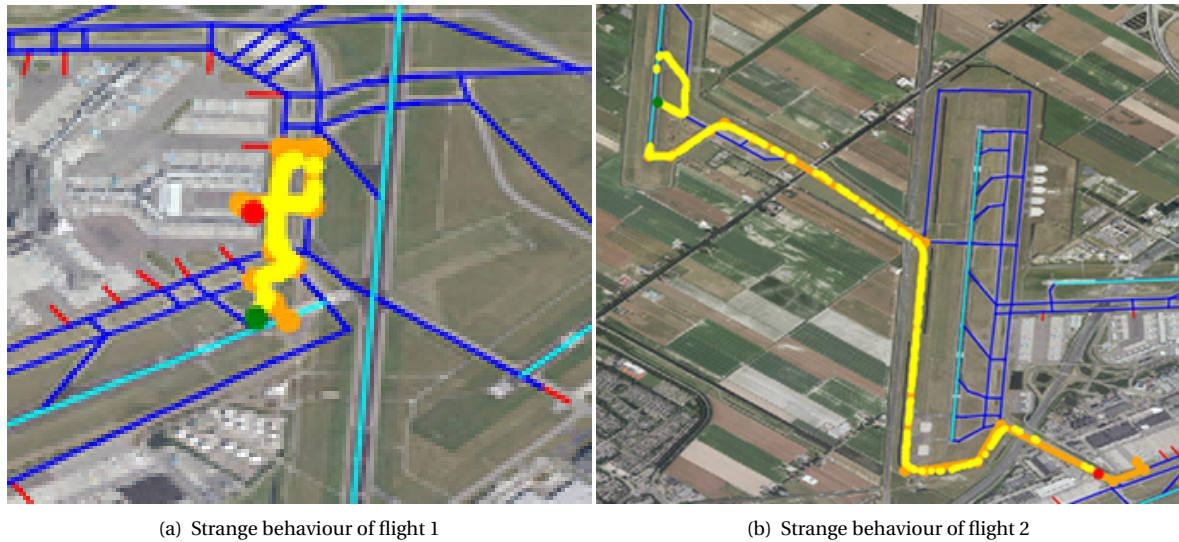


Figure 4.15: Strange behaviour of flights



Figure 4.16: Restrictions of a simplified layout

Missing data

Due to the lack of some ADS-B data and removal of flights that could not be processed, the flight schedule to simulate the operation is not complete. Additionally, when replaying ADS-B, one could see that an aircraft is slowing down without any noticeable traffic near that aircraft. It should therefore be noted that the missing flights have had their impact on the performance of the operation, meaning that certain observations in the actual operation can not be explained. Still, due to all filtering and processing that took place, a large set of high quality flight tracks is found. This data can be used to develop the model and perform a detailed case study.

In summary, ADS-B data from a large database is filtered, cleaned and processed, to obtain a set of useful and high quality flights. The individual data points are used to determine the most likely travelled routes. Manual inspections and corrections are applied to make sure these paths correspond to the routing strategies as applied by ATC of AAS. The ADS-B data points are projected onto this ground path. Interpolation is then performed to supplement the actual location data with an estimation of the aircraft's position for the missing time stamps. This is done to make sure the aircraft's location is known for every second in time, such that the update frequency is similar to the time step that is used in the simulation model. Subsequently, an actual flight schedule is derived based on the apron entry/exit and runway entry/exit of each flight, as well as the starting time.

Now that the actual flight schedules are drawn and ground paths of the actual operation are prepared for comparison with the simulated data, both items can be used to develop the simulation model that is able to guide aircraft according to this flight schedule. The development process is explained in the next chapter.

5

Agent-Based Model for Ground Operations

It is explained in the research framework in Section 3.4 that the simulation model is developed from an operational perspective. The simulated operation is continuously monitored and compared with the actual operation, to determine the model aspects that require improvements. This is done to get closer to the actual operation or to solve model issues. The assumptions used to develop the model are listed in Section 5.1. This research continues on an agent-based simulation model that has been designed by Udluft [72], which is briefly discussed in Section 5.2. This baseline model is further developed, based on requirements that have arisen from assessing the simulated operation with respect to the actual operation. Improvements and model expansions have been made to enable decentralised control for the ground operations of Amsterdam Airport Schiphol (AAS). A specification of the resulting agent-based model is given in Section 5.3, after which the model implementation is covered in Section 5.4. An overview of the model analysis and improvements that followed are described in Section 5.5. The simulation parameters are listed in Section 5.6.

5.1. Assumptions

As mentioned before, this research continues on earlier work that developed an agent-based model to simulate an airport's ground operation. The assumptions that have been made to further develop the simulation model and enable a case study for AAS airport are covered in this section.

Regarding the operational aspects, only taxiing traffic is modelled from the apron exit until the runway entry, or from the runway exit to the apron entry. Aircraft agents are obliged to travel to the exact same origin and destination (apron-runway point pair) as found in Automatic Dependent Surveillance-Broadcast (ADS-B) data. As no apron operations are included, it is also assumed that no towing operations take place to tow an aircraft to a long term parking position, or towards the maintenance facilities that are located in the eastern part of AAS. Minimum time separation values are specified for runway usage, where the only distinction in terms of occupancy time is made between arrivals and departures. Additionally, an apron occupancy time is specified for arriving at, or leaving, the apron. This time is included as it often occurs that aircraft moving in the apron area block other traffic, of which the push back process is an example. It should be noted that the occupancy time is applied before an aircraft leaves the apron or runway exit. It is also assumed that the gate of arriving aircraft is always available, such that no remote holding at a temporarily parking spot is required.

Aircraft agents have to be separated by at least a minimum separation distance, which is set to be constant for all aircraft. This separation distance is based on taxiway distance, meaning the closest point of approach is not included. Aircraft agents are only released from their origin if a safe operation is guaranteed. Also, it is decided that the earlier mentioned occupancy time only starts counting down if this safety condition is met.

Each aircraft agent has identical characteristics, and therefore every aircraft is treated similarly. This includes speed behaviour specifying that each aircraft aims at travelling at its maximum speed v_{\max} , unless it is required to slow down to the maximum allowed turn speed v_{turn} when it takes a turn that is sharper than the specified limit (see Section 5.6). Aircraft can change their heading instantaneously, as well as their acceleration. Braking is only performed in case the required deceleration is above the comfort deceleration dec_{com} .

No runway schedule is specified, meaning that the time, mode and direction of usage are not predefined. It is assumed that the runway usage does not depend on weather conditions, but instead on a flight's origin and destination as listed in the flight schedule that is fed into the model. Additionally, it is assumed that each runway is operating independently.

5.2. Baseline Model

This research continues on work done by Udluft [72], who modified the Open Source Simulator for ATM Research (OSSAR) resulting in an Agent-Based Modelling (ABM) tool to simulate ground operations. This tool has been developed in *Python 2.7*, using the *Pygame* module as interface. The simulator is designed in such a way that new agents, functions or layouts can easily be integrated or changed from one to another. Due to this characteristic and the fact that the basic operational dynamics are already implemented, the simulation model provides a solid baseline to focus on understanding and further developing the concept of decentralised control. The aim of this section is to provide an overview of the work that has been carried out so far, while more details of this baseline model can be found in Appendix B. Subsection 5.2.1 presents a summary of this baseline model. The limitations and weaknesses are described in Subsection 5.2.2.

5.2.1. Summary Baseline Model

Udluft's aim was to demonstrate that decentralised air traffic control is a feasible technique to perform ground operations. A simple airport layout is used, consisting of 3 gates, 18 intersections, 2 runways with two runway entries each. Udluft managed to implement decentralised control by placing local agents on each of the taxiway intersections, as well as on gates and runway nodes.

Aircraft agents are spawned into the network, based on a random flight schedule in terms of origin (gate), destination (runway entry) and start time. A gate agent adds the aircraft agent to the taxiway network, after which the intersection agents are responsible for guiding the aircraft agents safely and efficiently towards the runway. The moment an aircraft agent reaches its destination, it is removed from the network.

Intersection agents are able to coordinate with each other, in order to increase the efficient use of scarce set of taxiway segments. Coordination has been applied in the form of an auction protocol. Air Traffic Control (ATC) agents can place a bid on a taxiway segment, depending on the interest of the aircraft they are responsible for. The taxiway segment is allocated to the intersection agent with the highest bid, as he needs that segment the most. A detailed explanation of the coordination mechanism can be found in Appendix C.

5.2.2. Limitations Baseline Model

A large portion of Udluft's research focused on developing the ground operations module that is based on the OSSAR simulator. His research made a great contribution to the state of art in Air Traffic Management (ATM) simulations by showing that decentralised control is a viable technique to perform an airport's ground operation. However, its conclusions have restricted validity as a consequence of three main limitations:

- The conclusions are based on the use of a simple layout. Although not one uniform layout can be selected that covers all possible airport aspects, a few important infrastructural elements are missing. Currently the full layout is symmetric and each taxiway has a parallel segment, while operational challenge arise if only a single lane taxiway connects two parts of the taxiway network. Also, new agents might have to be introduced in case of runway crossings, an aspect missing in the used layout.
- Only departing traffic is modelled, of which the gate and runway entry are selected at random. The amount of conflicts that occur are limited, as there are almost no crossing flows of traffic. Instead, realistic flight schedules including both arriving and departing traffic should be used. This would put the required level of stress on the taxiway system, such that decentralised control is subject to a more realistic and demanding operation.
- Verification of the model is based on observations and small test scenarios. It would be very useful to develop this model in greater detail, such that it can be used in a case study of an actual airport. By assessing the model's performance with respect to the actual operation, insights are gathered on the requirements of decentralised control. Also, historical data can be used to evaluate and validate the concept of decentralised control. This way, it can be determined whether ABM and decentralised control are suitable approaches to manage the complexity and dynamics of an airport's ground operation.

All of these components are taken into account in this research, as this research aims at expanding the baseline model while performing a case study for AAS airport. The model's airport layout will closely match the

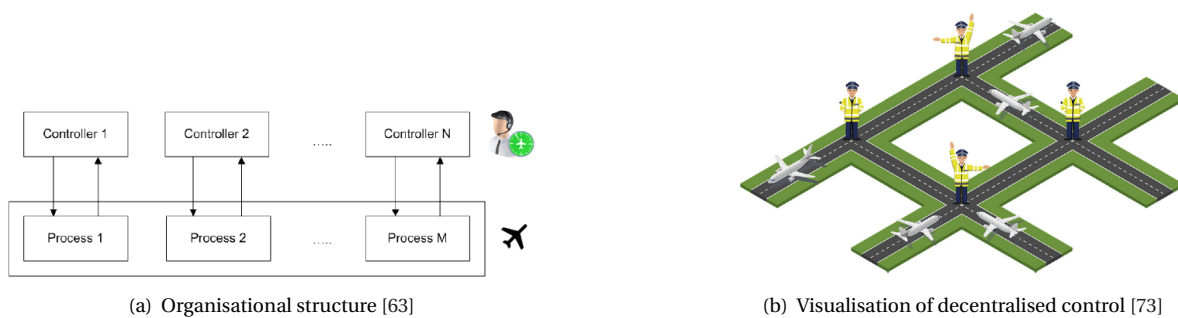


Figure 5.1: Decentralised Air Traffic Control

current layout of AAS. Based on historical data, an actual flight schedule is created. As this flight schedule serves as baseline for both the realised and simulation ground operation, the model can be validated. Also, the source code of the baseline model has been analysed in-depth, revealing a few weaknesses in the implementation and approach taken. These aspects have been used as input for further developments and improvements, which are described in the upcoming sections.

5.3. Model Specification

This section provides a specification of the developed agent-based model. The concept of implementing decentralised control in an agent-based model is briefly described in Subsection 5.3.1. A specification of the agents and their behaviour is given in Subsection 5.3.2, followed by a specification of the environment in Subsection 5.3.3. The interaction among agents, and the interaction between the agents and environment are presented in Subsection 5.3.4 and 5.3.5 respectively. It should be noted that this specification focuses on the developments that have been made with respect to the baseline model, as covered in Appendix B.

5.3.1. Implementing Decentralised Control in an Agent-Based Model

The concept of decentralised ATC is shown in Figure 5.1. The decision making is shifted to a local level by placing fictitious controllers on all taxiway intersections. Each of these intersection agents is responsible for its own process, consisting of all incoming and outgoing taxiway segments as is visualised in Subfigure 5.1(b). Often, a local agent is responsible for only a few aircraft simultaneously, which is why this agent can focus on solving this local conflict. Due to this limited responsibility, this agent can base its decision on a detailed understanding of the local circumstances, in order to solve this conflict both safely and efficiently.

The quality of the agent-based model is highly dependent on the effort that has been put in establishing the specification of the individual agents. The specification includes the definition of an agent's behavioural and cognitive characteristics. Additionally, the interaction in terms of observation, communication and coordination has to be specified. This interaction can either be among agents, or between an agent and the environment.

Summarised, the agent-based model is developed on the level of agent behaviour. As an agent acts based on observations, interactions and its understanding of the operation, its actions are not known in advance. Consequently, the overall behaviour of decentralised control is undefined, since it emerges from the local behaviour of agents.

5.3.2. Agent Specification

The baseline model only simulated traffic from their gate towards their runway, by means of five types of agents (see Appendix B). The apron agent acts as a source, while the runway agent is responsible for the task as sink of the taxiway network. As an actual flight schedule requires the runways and aprons to act as both a sink and source, a full makeover of the ATC structure has been required. This led to a redistribution of responsibilities, functionalities and properties across different agents. Additionally, a number of new agents had to be introduced, bringing the total amount of ATC agents to 7. Figure 5.2 shows a schematic overview of these agents. A specification of each of these agents is given below.

Source agent

Source agents are responsible for safely releasing an aircraft agent into the taxiway system, either at the

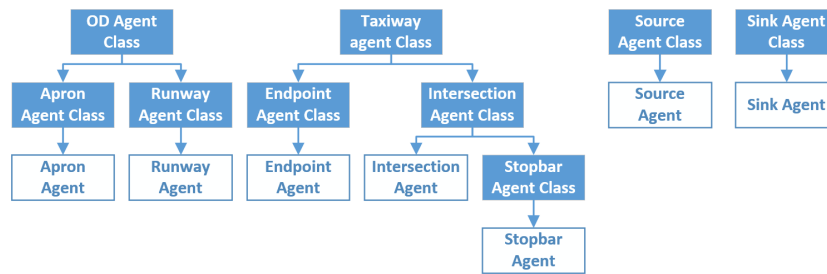


Figure 5.2: Structure of Air Traffic Control agents

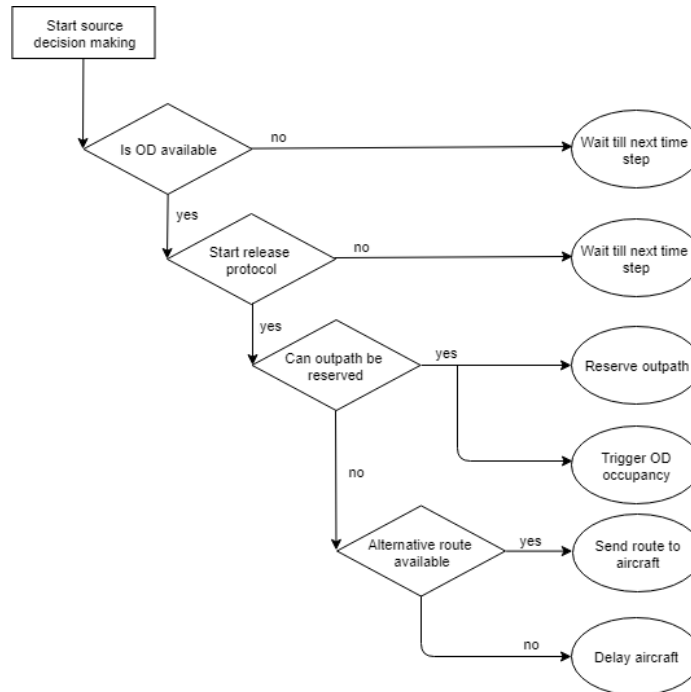


Figure 5.3: Decision making process of a source agent

apron or runway (Origin/Destination (OD)). At each time step, the source agent checks in the flight schedule whether it has to prepare for an upcoming release. From this schedule, it knows the intended apron or runway exit, as well as the aircraft's destination. The source agent also has access to the aircraft's current route, as well as the availability of the OD it belongs to. Additionally, it monitors the surrounding taxiway segments in terms of their occupancy and reservation status.

Timely in advance, the source agents starts to check whether the OD is available. If yes, the agent analyses which ATC agents have to be reserved to guarantee a safe outpath for the upcoming aircraft. In case all required resources can be reserved, this reservation is made and the OD's occupancy status is triggered. Otherwise, the source agent tries to find an alternative route for the aircraft. The decision making process of the source agent is shown in Figure 5.3. Information regarding the reservation is found in Subsection 5.4.2

Sink agent

Aircraft agents are removed from the taxiway network by the sink agents. When an aircraft reaches its destination, apron or runway, they are handed over towards the sink agent. The sink agent only has behavioural properties, for which it reacts to the observation that it has aircraft under its control: it triggers the occupancy time of the related OD and carries out the final administrative work of an aircraft agent.

Apron agent

The apron agent is responsible for accommodating the flow of aircraft agents entering and leaving the apron. This agent has an understanding of the aircraft agents willing to cross the apron boundary, as well as their direction of travel.

Every time step, the apron agent reduces its remaining occupancy time by 1 time step, until it reaches

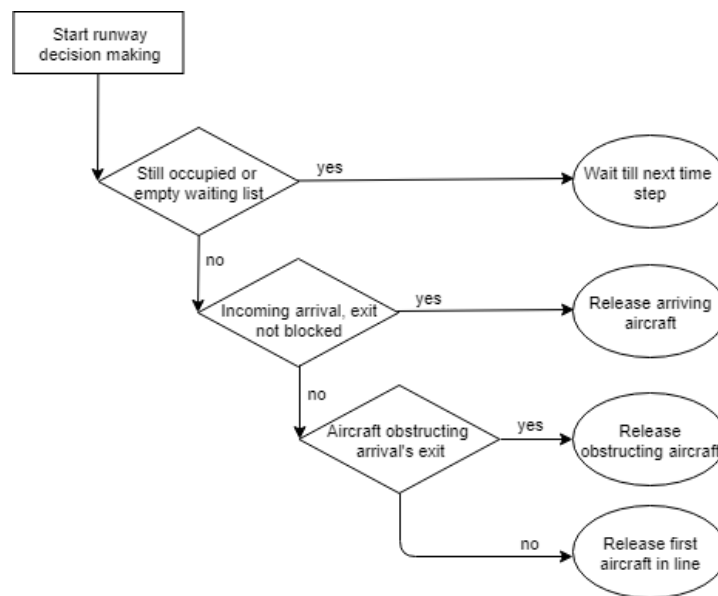


Figure 5.4: Decision making process of a runway agent

zero. The moment it reaches zero, it hands over the first aircraft in its queue: an arriving aircraft to the endpoint agent, or departing aircraft to the sink agent. The implementation is described in Subsection 5.4.4.

Runway agent

The runway agent is responsible for managing the flow of aircraft, that is willing to either take off, land or cross the runway. Just like the apron agent, the runway agent has insights in all aircraft that are waiting in the queue, as well as their intention to use the runway. Also, the runway agent can consult its source agent regarding the estimated touch down time of arriving aircraft, which have priority over any other traffic.

Every time step, a runway agent's occupancy time is reduced by 1 time step, until it reaches zero. Unless the runway agent is still occupied or the waiting list (runway queue) is empty, the runway agent evaluates whether it has to wait for an upcoming arrival aircraft. If there is an arriving aircraft and the intended runway exit is free, this agent is released to the endpoint agent and removed from the waiting list. Otherwise, if the intended runway exit is blocked, the most obstructing aircraft is released first. In all other cases, the first aircraft in the waiting list can use the runway. Figure 5.4 visualised this decision making process, while more information regarding the implementation is found in Subsection 5.4.4.

Apart from managing the runway usage, a runway agent is responsible for drawing up the schedule of future runway usage. Arriving times and runway exits are acquired from the flight schedule, while the runway entries and estimated departure times of departing traffic are known from surrounding ATC agents.

Every 10 seconds, the runway agent combines in the estimated times for both arrivals and departures. The expected direction of use is derived from the locations of the runway entries and exits to be used. Subsequently, the runway agent tries to merge similar modes of operation in larger blocks. The different blocks of operation are shifted to create a consecutive streak of future operations, for which priority is given to arrivals. This leads to a runway schedule of future modes of operation, which is shared with others. This topic is explained in more detail in Subsection 5.4.3.

Endpoint agent

Endpoint agents are responsible for slowing down aircraft which are reaching their destination, either to line-up on the runway or enter the apron. In both cases, the OD is informed that the aircraft is ready to leave the taxiway network. Aircraft that are spawned into the taxiway network, start at the endpoint agent. As the source agent has already reserved an outgoing path for this aircraft, the endpoint agent only has to hand over the aircraft to its neighbouring ATC agent.

Intersection agent

Intersection agents are responsible for controlling the aircraft, by giving route, speed, and stop commands.

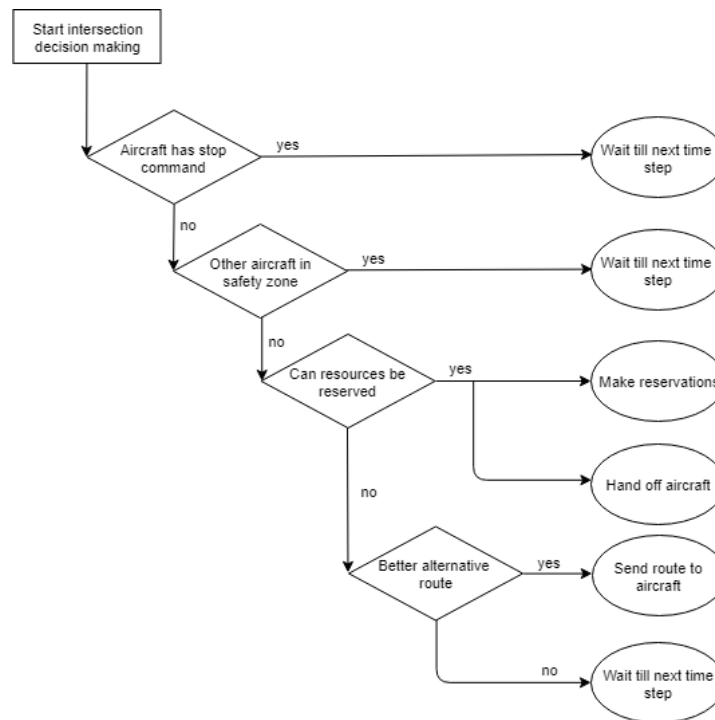


Figure 5.5: Decision making process of an intersection agent

An intersection agent has control over all the aircraft on its inbound segments as well as the aircraft it holds a reservation for. It has knowledge of the position, speed and current route of all aircraft that are on its adjacent segments. Also, it has an understanding of the traffic state of the network, as well as the reservation status of the surrounding segments.

The intersection agent uses this information to continuously determine the shortest path for all aircraft under control, by means of Dijkstra's algorithm. To solve conflicts, the agent can make a manual correction to the shortest path by forcing an aircraft to make a detour. Also, the intersection agent estimates the aircraft's remaining taxi time to reach its destination.

Before handing over an aircraft agent towards the next ATC agent, the intersection agent has to check whether any future ATC agents or taxiway segments have to be reserved in advance to guarantee a safe hand-off. For each hand off, it is checked whether no stop command applies to the aircraft, and that no other aircraft is within the safety zone of the intersection agent. Based on the aircraft's current route, the required reservations are derived. The intersection agent checks whether all resources can be reserved. If yes, the reservations are made. Otherwise, the intersection agent makes the trade of to wait or use an alternative route. The decision making is shown in Figure 5.5. The reservation protocol is explained in Subsection 5.4.2.

Stopbar agent

Stopbar agents are responsible for accommodating a safe runway operation, by controlling the flow of aircraft willing to enter, leave or cross a runway. The agent also registers aircraft, that are requesting a runway operation, to the runway's waiting list. The stopbar agent has access to the future modes of runway operation, and the runway's flight schedule in terms of arrival times and associated runway exits. Additionally, it has an understanding of the intention of aircraft requesting a runway operation, as well as the intended time of use.

Stopbar agents act as a conditional diode, in the way that they can reject an aircraft's request to enter a segment. For each request, the stopbar agent evaluates whether it approves the agent to enter the segment. In case of an aircraft leaving the runway, either after arrival or runway crossing, this request is always approved. For aircraft willing to cross or depart from the runway, the stopbar agent tries to find a gap in the runway schedule that allows this operation, without delaying any of the upcoming arrivals. If there is no arriving aircraft willing to use that particular segment in the meantime, the clearance is given. In all other situations, the request is rejected. The decision making is shown in Figure 5.6, while the implementation is covered in Subsection 5.4.5. After receiving a clearance to enter, the aircraft agent will be brought to a hold at the stopbar

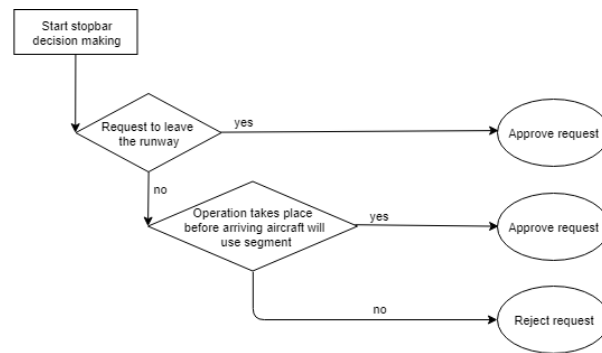


Figure 5.6: Decision making process of a stopbar agent part I

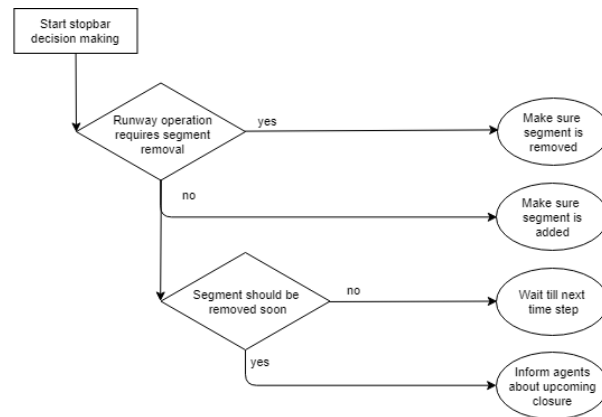


Figure 5.7: Decision making process of an intersection agent part II

agent. The aircraft is added to the waiting list, and is being when it receives a go from the runway agent.

Apart from controlling aircraft willing to enter the runway, the stopbar agent is also responsible for removing segments in the taxiway system. Removing these segments is necessary to prevent interference with the runway operation. The stopbar agent knows the future Runway Mode of Operation (RMO), as well as the removal conditions, and availability status of the segments that have to be removed. In case the removal procedure starts, the stopbar agent alters the state of the taxiway network, which is captured by all other ATC agents.

At every time step, the stopbar agent consults the runway agent regarding the latest update of the runway schedule's. The stopbar agent evaluates whether the removal conditions are satisfied with the upcoming RMO. While counting down towards the new RMO becoming active, it urges the other ATC agents to update the routes of aircraft in which this particular segment is excluded. The moment the new RMO becomes active, it removes this segment from the network until the removal conditions are no longer satisfied. A visualisation is shown in Figure 5.7, while the implementation of this feature is covered in Subsection 5.4.6.

These seven types of ATC agents are implemented in the agent-based model to enable decentralised control. The eight type of agent is the aircraft agent, which is described next.

Aircraft agent

An aircraft agent follows the commands of the ATC agents, maintaining a safe distance with respect to the other aircraft agents. The goal of to agent is to reach its destination as quickly as possible, and tries to accelerate up to its maximum speed where possible. No major changes have been made to the aircraft agents, so the specification in Appendix B still applies.

A simple example of the different agents that an aircraft encounters along its flight is presented in Figure 5.8. A departing aircraft is "created" by a source agent at the time of departure, after which it is added to the readiness queue of the apron. The aircraft agent enters the taxiway system via the endpoint agent, after which it passes multiple intersection agents along its route towards the destination. The stopbar agent has to give permission to enter the segment towards the runway, after which it is processed and removed from the

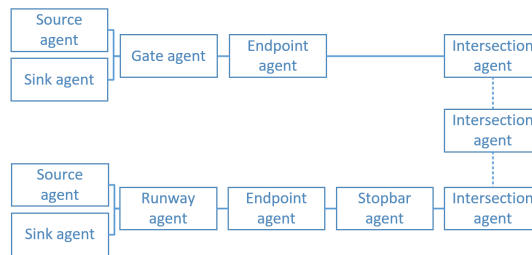


Figure 5.8: Example of relation between Air Traffic Control agents

airport by the sink agent.

In case the remainder of this report mentions a *node*, it refers to a taxiway agent; so either an endpoint, stopbar, or intersection agent. Additionally, for convention it is decided that in case an aircraft has passed ATC agent *A* and travels towards ATC agent *B*, then *A* and *B* are seen as previous node and next node respectively. The intersection, stopbar and endpoint agent together comprise the taxiway agents.

5.3.3. Environment Specification

The environment of an agent-based model consists of the objects that are not related to an agent. The agents are able to observe the environment and act upon it. This subsection describes the two types of objects that are included in the model.

A flight schedule is defined by all flights that take place during the simulated hours. Each flight is characterised by an identification number, as well as its starting time. Additionally, the point of entry and exit in the airport layout is specified. This object is static and can only be accessed/observed by the source agents.

The taxiway network is a directed graph representing the airport's taxiway infrastructure. This object is similar to the one in the baseline model as described in Appendix B, apart from one aspect. The directed taxiway edges have a new variable that could store a reservation, consisting of the issuer and aircraft it is reserved for. This information can be accessed by all ATC agents. Placing a reservation on a directed taxiway edge leads to the removal of the edge that opposes this reservation. More information is found in Subsection 5.4.2.

5.3.4. Interaction among Agents

The types of interactions that were already in place in the baseline model, are explained in Appendix B. Similar interactions among the agents are still in place, although reorganised over the new types of agent. This subsection focuses on the newly implemented types of interaction.

Discuss and make reservation

ATC agents are able to overrule the decision power of other ATC agents by forcing a reservation on the other agent regarding the aircraft that has to be handed off next. This could be one of the requirements for a safe release of aircraft, as explained in Subsection 5.4.2. To do so, interaction is required in terms of:

- Asking the ATC agent whether it is already reserved by another agent, followed by a boolean reply.
- Forcing a reservation on the other agent, consisting of the aircraft's identification number, the segment it is coming from, as well as the taxiway segment it has to travel next.

Determine future modes of operation

For each time step, an intersection agents determine the shortest path for all aircraft under its control. Dijkstra's algorithm is applied to a network graph of the current state of the taxiway network, where the weights of the segments are an estimation of the taxi time. Consequently, intersection agents have an estimated taxi time for the aircraft under their control. When runway agents update their runway schedule, they need to know when departing aircraft can take off. Additionally, the arrival times are need to be known. The following interactions takes place:

- A runway agent sends out a global message with the request to receive estimated departure times for that particular runway. All intersection agent that are responsible for an aircraft whose destination is the runway, reply with the estimated time the aircraft reaches the runway.
- A runway agent consults its source agent regarding the expected arrival times, which it receives from the source agent.

Share runway schedule

A runway agent draws up its runway schedule of its future modes of operation. This information is being shared with the stopbar agents that belong to the runway. This information is used by the stopbar agents to determine if, and when taxiway segments have to be closed. The interaction is simple:

- A runway agent sends the latest version of the future runway schedule to all linked stopbar agents, the moment it has been updated.

Request to pass stopbar

To enter a segments heading towards the runway, an aircraft agent requires approval from a stopbar agent. The neighbouring ATC agents is responsible for submitting a request to the stopbar agent, including the aircraft's remaining taxi time and intended runway operation, e.g. taking off or crossing. The stopbar agent requires insights in the runway's source schedule and waiting list, to evaluate whether the request can be approved without delaying arriving aircraft. The conclusion is shared with the neighbouring ATC agent. Summarised, the following interactions are performed:

- An ATC agent submits a request to let an aircraft agent enter the segment towards the stopbar agent. This request includes the estimated taxi time to reach the runway, as well as the intended runway operation, e.g. taking off or runway crossing.
- The stopbar agent consults the runway agent for its current waiting list, and overview of arrival times. This latter request is forwarded towards the runway's source agent.
- The stopbar agent sends its approval or rejection to the ATC agent

5.3.5. Interaction between Agents and Environment

The already included interactions between agents and the environment are covered in Appendix B. This subsection describes the interactions that have been added to the model.

Reservation taxiway segments

To safely hand off an aircraft, it might be required to reserve a few taxiway segments. This is done to prevent other aircraft from entering these segments, which could lead to a loss of separation in case of short taxiway segments. An ATC agent can alter the reservation status of a directed edge by making a reservation, or cancelling it when the aircraft is travelling across it. By forcing a reservation on a directed edge, the opposing edge is automatically removed. ATC agents have access to the state of the taxiway network, which does include this reservation status. The reservation protocol has added two new types of interaction:

- An ATC agent can force or remove a reservation on a taxiway segment by altering the reservation status of the directed edge.
- The reservation status is observable by all ATC agents,

5.4. Model Implementation

This section discusses the implementation of the new model features and extensions. Subsection 5.4.1 allowing close monitoring of the actual and simulated operation. Subsection 5.4.2 explains the implementation of the reservation protocol. Runways have been made intelligent as can be read in Subsection 5.4.3. New procedures to enter or leave the taxiway network are covered in 5.4.4. The new stopbar agent is introduced in Subsection 5.4.5. The process to close taxiways is described in Subsection 5.4.6.

5.4.1. Model Interface

One of the key aspects of this research is the continuous comparison between the simulated and actual operation. A new interface has been designed, which allows close monitoring of both operations, to identify potential inaccuracies or operational differences. This simple, but very useful interface, is shown in Figure 5.9. Arriving aircraft are shown in red, while departing aircraft are green. The left image represents the simulated operation which is running "real-time", while the right image displays the actual operation from the processed ADS-B tracks as explained in Section 4.3. In case odd behaviour is identified, it is very easy to switch to the debugging mode. One can run the simulation time step by time step, while observing the related aircraft behaviour using the interface. Apart from this dual screen mode, the interface also allows the researcher to show only the simulation, which allows the researcher to temporarily focus on a different airport layout to test very specific scenarios. Additionally, this interface can be used to replay the operation.

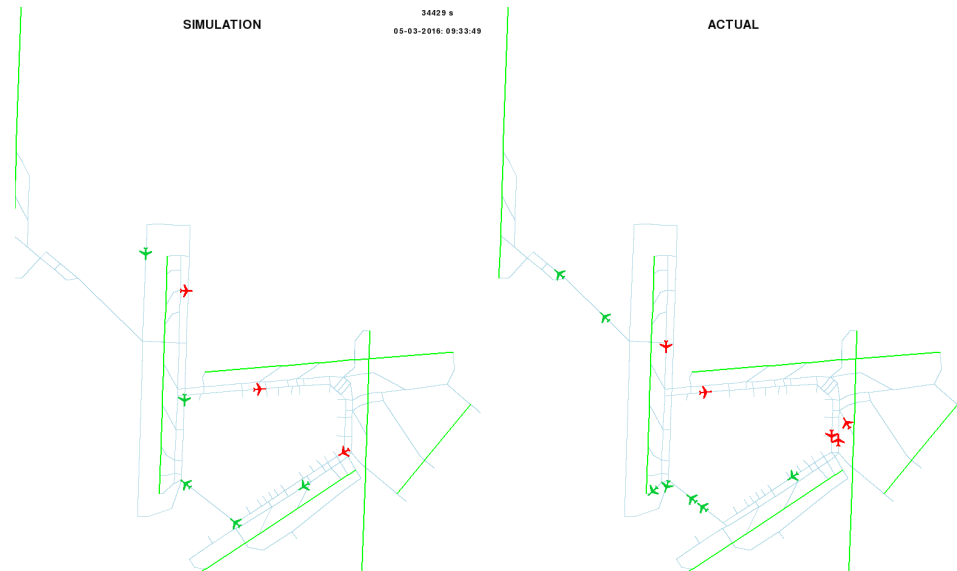


Figure 5.9: Model interface to closely monitor the simulated and actual operation

5.4.2. Reservation Mechanism

Issue

With only one aircraft agent on the taxiway network, the agent-based model is able to safely guide the aircraft towards its destination. Adding more aircraft to the system leads to congested areas, which eventually results in separation infringements and related gridlocks. These infringements are caused by taxiway segments with a length that is about equal or smaller than the separation distance specified. In the baseline model, whenever an intersection agent wants to hand off an aircraft, it checks whether there is no other aircraft agent on that particular segment within twice the separation distance. In case there is none, the aircraft receives its hand off, after which it is being guided towards its next link. However, for small segments this aircraft directly enters the safety zone of the next node, which is defined by the minimum separation distance d_{sep} . If this other agent is simultaneously handing off another aircraft, both aircraft within the safety zone receive a command to come to a full stop to limit the impact of the minimum separation distance that is violated.

This issue concerns all segments with a length shorter than $2v_{max} \cdot dt + \frac{v_{max}^2}{2 \cdot dec_{max}} + d_{sep}$, equalling the distance travelled at maximum speed v_{max} during two time steps dt , the distance required to decelerate from maximum speed to get to a full stop, and the separation distance d_{sep} . This taxiway length is referred to as the reservation distance d_{res} . An ATC agent hands over an aircraft agent to the next agent, the moment it has passed. Practically, this could mean an aircraft has crossed the node by slightly less than $v_{max} \cdot dt$ metres. The process of handover itself takes one time step dt , in which the aircraft could travel another $v_{max} \cdot dt$ distance, after which the next agent has its first opportunity to command the aircraft to come to a full stop. For the parameters used along the model development phase (see Section 5.6), this results in a required segment length of 204 metres. The order size of this issue is large, as 190 out of the 266 taxiway segments of the simplified layout of AAS is shorter than 204 metres.

Implementation

Before handing off an aircraft, the ATC agent has to be sure that the upcoming ATC agents are able to accommodate this aircraft and that they do not start the handing off procedure of another aircraft. These demands require the implementation of a reservation mechanism, which allows an ATC agent to "reserve" the operation of any other ATC agent as well as the taxiway segments to be used.

The main implementation of the reservation system requires changes for all ATC agents, the aircraft agents and the network graph (environment). First of all, each taxiway agent has to be able to store and manage a reservation. A reservation consists of three parts, namely the involved aircraft agent, the segment it comes from and the segment it would like to use next. The former is being referred to as *reserved aircraft*. A function can be activated by other ATC agents to store a reservation. This triggers another function, which applies a reservation to the segment the aircraft intends to use next. Also, the taxiway link opposing this reserved edge

is removed. As long as a reservation is in place, only this aircraft agent is allowed to be handed off. In all other cases, the taxiway agent sorts the aircraft based on the time they are under its control of the ATC agent. The reservation drops the moment the aircraft has been handed off by that particular agent.

It should be noted that only one reservation can be stored at each individual agent or segment, and this reservation is based on a First Come, First Served (FCFS) basis. Agents are not able to reject a reservation.

Each ATC agent requires the ability to reserve other agents in name of the aircraft it has under control. From the aircraft's current route, an ATC agent is able to deduce which set of ATC agents and taxiway segments are within the reservation distance d_{res} . When the ATC agent wants to hand off the aircraft, it checks whether all agents and segments within the set can be reserved. Only if all these resources available and neither of the agents is currently handing off an aircraft, the hand off clearance is given. A reservation is placed on all relevant agents and segments, and the aircraft is handed a list with all reserved ATC agents. Otherwise, the aircraft does not receive its handing off clearance and has to wait for the next time step before the same condition is checked. An aircraft agent possesses a list of all taxiway nodes that are reserved for that specific aircraft. ATC agents can request access to this list, and inform the aircraft to add or remove an agent from this list.

Effect

The implemented reservation mechanism allows ATC agents to claim the use of other resources, to be sure that they are aware and prepared for the upcoming release of the aircraft agent. Consequently, aircraft agents can be handed off safely thanks to this reservation protocol.

5.4.3. Runway Schedule

Issue

Runway operations have a large impact on the ground operations: arriving aircraft have to be absorbed quickly by the surrounding ATC agents, while departing traffic could cause to congested areas as runway queues are formed. Also, particular taxiway segments might have to be closed depending on the RMO, as traffic could interfere with the runway operation. Therefore, a runway reconfiguration could cause a major operational disturbance. Currently, the ATC agents have no insights when reconfigurations will take place, which raises the need to have insights in (future) runway usage.

Implementation

A runway schedule is created by providing runway agents with a feature to determine their future RMO. A runway agent consults its source to obtain the arrival times t_{arr} . Also, the runway collects a departure schedule by asking all ATC agents whether they are responsible for an aircraft which is heading towards that runway. If yes, the intended runway entry and estimated departure time t_{dep} is returned. This is known since the ATC agent is responsible for finding the best route, meaning it has an estimation of the taxi time. By combining both schedules with the standard occupancy times for arrivals and departures, as well as a $t_{margin}^{runway} = 30$ seconds safety margin prior to and following the flight, the runway agent is able to construct arrival and departure blocks. Within these blocks, the respective flight blocks are merged if they are less than 5 minutes apart and in similar direction. This direction of runway usage is determined based on the assumption that an arrival is heading towards the runway's end that is closest to the runway exit, while departures take off in the direction away from the runway's end surrounding the runway entry. The subsets of arrivals and departures are placed sequentially sorted on start time, and shifted in case they overlap. Remaining gaps are labelled "unused". The future modes of operation are characterised by the operation type, start time, end time and direction. This process is repeated every $t_{future}^{runway} = 10$ seconds. The future runway schedule is shared with the stopbar agents. Figure 5.10 shows the decision logic that a runway agent applies to determine its future usage.

Effect

This implementation enables the draw up of a runway's future modes of operation helps ATC agents to timely react to (upcoming) changes in the runway configurations.

5.4.4. Entering or leaving the Taxiway System

Issue

An actual flight schedule requires the apron and runway to be used as both source and sink. Consequently, procedures are required to make sure the single lane taxiway segments towards an endpoint of the apron and runway can handle both incoming and outgoing traffic. These procedures are based on the concept that ar-

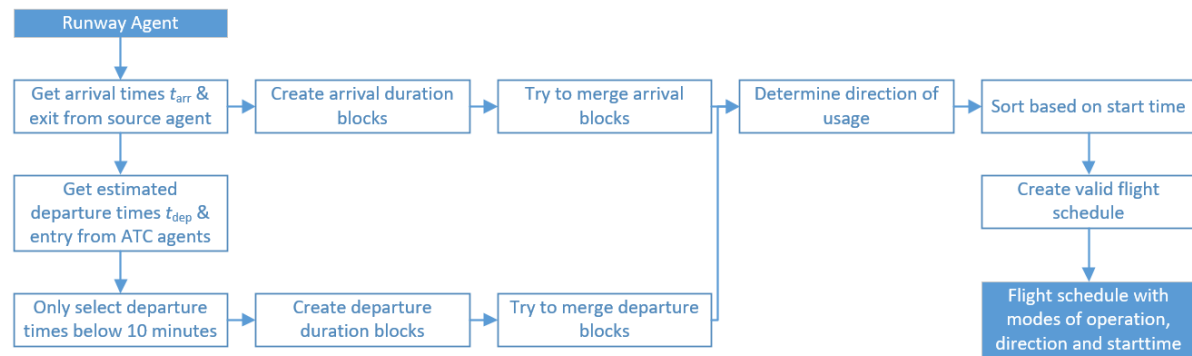


Figure 5.10: Procedure to estimate the future usage of a runway

iving traffic has priority to and from an endpoint. Priority on a runway is required to quickly make room for the next arrival, which has limited to no room to absorb delay. Additionally, ATC prefers holding a departing aircraft at its gate above holding an arriving aircraft at a taxiway, as this latter, which leads to a more congested area. Additionally, removing an aircraft from the taxiway network results in a decrease in ATC workload. It is why departing aircraft do not receive a push back in the actual operation, in case an arriving aircraft is about to enter the same apron.

In the baseline model, the separation time between two departing aircraft is specified. However, no occupancy time of the apron or runway is specified for arriving aircraft.

Implementation

It is decided that removing an aircraft from the network always has to trigger an occupancy time for the OD it has used. An apron becomes temporarily occupied the moment an aircraft enters the apron, as an aircraft that is about to park is taxiing at slow speed and could thereby block the apron's main segment. On the other hand, an apron is blocked for a fixed time interval preceding the scheduled apron release, to emulate an aircraft that is being pushed back, and finishing its checklist on the apron's main segment. As a result, the occupancy time takes place before releasing an aircraft into the network. ATC aims at minimising the amount of aircraft moving simultaneously on the taxiway, and therefore prefer to delay a push back in case an arriving is about to enter that same apron. This behaviour has been emulated by source agents, as they analyse the surroundings before deciding whether an arriving aircraft can reserve the apron and outgoing segment.

A similar approach is used for the runway: an aircraft on final approach receives a landing clearance in a timely manner, in case the runway and expected runway exit are free to use. Consequently, the runway is set to be occupied to make sure that the arriving aircraft can touch down and use an exit at high speed in a safe manner. From the flight schedule, it is known when an aircraft should touch down. The occupancy time t_{arr}^{runway} in advance, it is checked whether all conditions are satisfied. These conditions include availability of the runway and a successful reservation of an outpath the upcoming aircraft. In case both conditions are satisfied, the occupancy time starts counting down till 0 seconds which is when the aircraft is released. Otherwise, the arriving aircraft will be delayed as no safe release is guaranteed.

Effect

By applying occupancy times in advance of an aircraft release, more representative conditions are created for the use of runways and aprons. Additionally, now that the reservation of an outpath is a condition to release an aircraft, it is assured that there is no loss of separation the moment an aircraft is released into the network.

5.4.5. Stopbar Agent

Issue

Occasionally, simulations are performed using random schedules. Randomness in start times and destinations leads to chaotic flows of traffic, enabling the opportunity to evaluate the model under challenging conditions. It is found that many arriving flights are delayed due to departing flights that enter the runway entry segment just before the runway wants to reserve the opposing segment for the arrival flight. The time it takes before the departing aircraft takes off and the departure occupancy time has passed, causes large delays for the arriving aircraft. This scenario is highly undesired and far from reality, as arriving aircraft have top priority due to the limited options to process delay and related big impact on the airspace operation.

Looking at the location of AAS's runways in Figure 2.1, it is seen that multiple runway crossings can take place depending on the runways in use. In case runway 18C – 36C is used, aircraft taxiing towards or from runway 18R – 36L cross runway 18C – 36C in the middle. Taking off from runway 18L requires (at least all) heavy aircraft to cross runway 09 – 27, before lining up on the start of runway 18L for take off. Also, one of AAS's cargo platforms is located south of runway 06 – 24 and can only be reached by crossing this runway. Consequently, enabling runway crossings is an important requirement to simulate AAS's ground operation. However, runway crossings are still not possible as the current types of ATC agents do not allow this kind of operation.

Implementation

To incorporate new features to support the runway operation, a stopbar agent has been introduced as new type of agent. A stopbar agent can be seen as an intersection agent with additional responsibilities, which is why this stopbar agent class is a child of the intersection agent class (see Figure 5.2).

Stopbar agents are added halfway the segments that connect the runway's endpoint with the main taxiway layout. When a departing aircraft is a reservation distance d_{res} away from entering the runway entry segment, the stopbar agent is being contacted to determine whether the aircraft can use this segment. Firstly, the stopbar estimates the required time duration during which the departing aircraft blocks the runway based on remaining taxi time and the runway occupancy time for departures. Secondly, this stopbar checks whether this time duration fits within the runway's arrival schedule, taking the runway occupancy time for arrivals into account. An additional time margin $t_{margin}^{stop\ bar}$, set to 120 seconds, between the end of the departure's occupancy time and start of the arrival is included, to be sure the arrival is not delayed.

In case a suitable gap is found in the runway schedule, all aircraft scheduled in advance of this gap are analysed in terms of the runway entry or exit they intent to use. If entering the segment towards the stopbar agent does not create any conflict with the found intentions, the departing aircraft is allowed to move forward. Otherwise, the departing aircraft receives a stop command until the criteria is met. This feature partly acts as a diode: aircraft leaving the runway can always pass the stopbar agent, while entering aircraft cannot. Only if all conditions are specified, the stopbar "temporarily" opens for the departing aircraft.

To enable runway crossings, a small adjustment in the taxiway infrastructure is required. In earlier stages of the model, segments leading towards a runway could only be used for entering or leaving the runway. This has been modelled by a directed link between the endpoint and stop bar agent. At particular locations, these segments are used for runway crossings, meaning two stop bars agents are indirectly connected with an endpoint agent in between. As an endpoint agent is missing the required capabilities and can be omitted in the crossing process, an additional segment is drawn between the two stop bar agents. Consequently, when an aircraft reaches the stopbar agent, it can either take the taxiway to the endpoint agent to take off or take the taxiway towards the stopbar agent on the other side of the runway. In the remainder of this report, this latter segment is called the *stop bar segment*.

When an aircraft agent wants to cross the runway, it has to follow a similar procedure as for an aircraft willing to take off. A crossing request is sent to the stopbar agent for permission to proceed. This request has been altered to also include the aircraft's current route, such that the stopbar agent is able to determine the type of usage: arrival, departure or runway crossing. For the latter case two conditions have to be met. Firstly, the crossing aircraft has to fit within the runway's schedule for which travel time and runway occupancy time are included. This latter parameter, t_{cross}^{runway} , is estimated to be 180 seconds. If the runway is used in departure mode, traffic is handled according to the FCFS principle. Secondly, similar to the runway exit reservation for arriving aircraft, a crossing aircraft is only allowed to proceed, if it could free the runway with a distance equalling at least the reservation distance d_{res} .

At an actual airport, a Runway Safety Area (RSA) is defined as a "safety zone surrounding the runway prepared or suitable for reducing the risk of damage to aircraft in the event of an undershoot, overshoot, or excursion from the runway" [8]. As the stop bar agents offer a protective boundary around the runway, it is decided that these agents act as the RSA. Subsequently, for both arriving and crossing aircraft the reservation distance d_{res} has to be cleared from the last stop bar agent it crosses, instead of the runway centre line.

Effect

The implementation of the stopbar agent has led to two key features. First of all, the stopbar agent enables

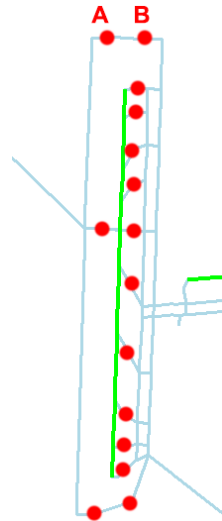


Figure 5.11: Location of stopbars surrounding runway 18C – 36C

runway crossing. Besides, this new agent protects the arriving aircraft by making sure that take offs and runway crossings are only allowed in case delays for arriving aircraft are unlikely.

5.4.6. Taxiway Closure

Issue

Aircraft agents can reach runway 18R – 36L, see Figure 2.1, in three different ways: by crossing runway 18C – 36C in the middle, or by taxiing across one of the taxiways located north or south of runway 18C – 36C. Two of these three options are closed depending on the direction and mode of operation of runway 18C – 36C. Activation of this runway prohibits runway crossings. Taking off in northerly direction or landing in southerly direction requires closure of the northern taxiway route, while the southern taxiway route is closed when taking off in southerly direction or landing in northerly direction. Consequently, there is a need to adapt traffic flows in case runway 18C – 36C is to be used.

Implementation

The responsibility to remove taxiway segments is given to the stopbar agent, since runway crossing takes place along a segment that connects two stopbars. Before the runway becomes operational, it is checked whether no runway crossing is being executed. Stopbar agents have to remove all outgoing segments towards other stopbar agents, the stopbar segments, such that runway crossings can no longer take place. To close either the northern or southern direction route, it is decided to add stopbar agents along these routes. All stopbar agents related to runway 18C – 36C are shown in Figure 5.11.

In the agent initialisation file, information has to be specified if and for what runway usage a stopbar segment has to be closed. This should include both runway mode and direction of use. As an example, stopbars A and B have to be closed when the runway is used for arrivals heading 180° , or when departing towards heading 360° . Before simulation takes place, a stopbar stores in its memory all outgoing taxiway segments that are directly linked to other stopbar agents of the same runway. Every time step, a stopbar agent checks whether the related runway agent is being operated in a way for which the stopbar agent has to be closed. If this is the case, it makes sure that all outgoing segments in its memory are safely removed. Otherwise, it checks whether the segment can be added back if it is not in place yet. By removing the segments from the taxiway layout, other ATC agents can no longer include these prohibited segments in an aircraft's route.

Effect

By removing these taxiway segments, it is ensured that taxi movements no longer take place along segments that interfere or conflict with the runway operation. Since the removed segments can no longer be included in a route, the ATC agents adapt their routing strategies to the runway configuration in use.

Issue

As long as runway 18C – 36C is inactive, aircraft heading towards or away from 18R – 36L cross this runway

since it results in the fastest route. As ATC agents do not have insights into the future state of the taxiway system, they are not able to foresee sudden taxiway closures. During simulations, it has been found that activating runway 18C – 36C results in a disturbed and chaotic traffic state as many route updates are required to adapt to the actual taxiway infrastructure. In case runway 18C – 36C is activated for departures in northern direction, the crossing segment is closed. For aircraft coming from one of the southern aprons, this could lead to the situation where aircraft are close to crossing the runway and suddenly have to turn around and use the southern taxiway to reach runway 18R – 36L. Consequently, sudden closures of taxiway segments has a large impact on the operational efficiency.

Implementation

As long as runway 18C – 36C is inactive, aircraft heading towards or away from 18R – 36L cross this runway since it results in the fastest route. As ATC agents do not have insights into the future state of the taxiway system, they are not able to foresee sudden taxiway closures. During simulations, it has been found that activating runway 18C – 36C results in a disturbed and chaotic traffic state as many route updates are required to adapt to the actual taxiway infrastructure. In case runway 18C – 36C is activated for departures in northern direction, the crossing segment is closed. For aircraft coming from one of the southern aprons, this could lead to the situation where aircraft are close to crossing the runway and suddenly have to turn around and use the southern taxiway to reach runway 18R – 36L. Another scenario limiting the ground operation is when an aircraft is entering the segment towards the stopbar controlling the crossing segment, when the runway is suddenly being activated for arrivals. Where ATC agents initially thought crossing the runway would reduce the aircraft's travel time, it turns out it is actually being delayed as it has to wait for a gap in the arrival traffic.

It is found that penalising the crossing segments in advance, leads to different routes that are preferred in terms of shortest path, as runway crossings are no longer included. The responsibility of penalising weights is given to the stopbar agent, which can adapt the weight of the outgoing segments that have to be removed soon. Since the estimated start time of the future RMO is also found in the runway schedule, a stopbar knows when it has to close the segment. The segment's weight is increased from its original value w_{original} up till w_{new} using Equation (5.1). The multiplication factor consists of two parts. The latter part is the penalty factor f_{penalty} , being the maximum factor by which the segment's weight is multiplied. The former factor is the time dependent factor, which makes sure the segment's weight increases in time. A stopbar has insights in all runway reconfigurations that take place within a time horizon $t_{\text{hor}}^{\text{runway}} = 300$ seconds. When the stopbar agent realises the runway reconfiguration takes place in 300 seconds, this factor is zero. At the moment the runway reconfiguration takes place, $t = t_{\text{reconf}}$, this factor is 1, meaning that the segment reaches its maximum weight.

$$w_{\text{new}} = w_{\text{original}} \cdot \left(1 + \frac{t_{\text{hor}}^{\text{runway}} - (t_{\text{reconf}} - t)}{t_{\text{hor}}^{\text{runway}}} \cdot f_{\text{penalty}} \right) \quad (5.1)$$

Effect

Stopbar segments are able to increase a segment's weight in advance of its closure, to make sure that other ATC agents are informed indirectly about the upcoming changes in the taxiway layout. Based on the increased weight, ATC agents are able to decide themselves whether the segment can still be included in a route.

5.5. Model Analysis and Improvement

The baseline model, as well as the developed model features and extensions, have been verified and analysed. This section discusses the improvements that resulted from this analysis. The improvements for the reservation mechanism and drawing up of the runway schedule are listed in Subsection 5.5.1 and 5.5.2 respectively. Subsection 5.5.3 describes the improved aspects of the auction protocol, followed by the routing algorithm in Subsection 5.5.4. The conflict solving abilities are improved as explained in Subsection 5.5.5. Subsection 5.5.6 provides an analysis of the taxiway graph, after which Subsection 5.5.7 lists some additional adjustments that have been made.

5.5.1. Improvements Reservation Mechanism

Partial responsibility in speed and routing commands

To improve efficiency and reality of the operation, it is required that a reserved intersection agent requires partial responsibility and control over the aircraft it holds the reservation for. This aspect is clearly shown by two examples. In the current model, only the next ATC agent is able to control the aircraft's speed by giving

speed commands. These speed commands are based on congestion around the intersection, or speed reduction commands when it has to take a turn. In case the aircraft goes straight and enters a short segment before making the turn, it might realise that the segment's length is too short to reduce speed from maximum speed v_{\max} to maximum turn speed v_{turn} . This situation is shown in Figure 5.12, where aircraft *AC1* is controlled by agent *A* while node *B* is reserved for this aircraft. To be sure the speed of aircraft *AC1* is at most v_{turn} at node *B*, a timely speed reduction command has to be given.

Every time step, an ATC agent checks whether an improved route can be found for all aircraft under its control. No routing updates are given while an aircraft is being handed off, to prevent last minute changes that are prone to taking wrong turns. Normally, aircraft receive a hand off clearance when they are on an inbound segment and a separation distance d_{sep} away from an ATC agent. However, on short segments this hand off clearance is given the moment an aircraft enters the inbound segment, which is why the node has to be reserved in advance. However, as node *B* is involved in multiple auctions, it can both win and lose auctions. As route updates might be required as a result of their performed auction. A simple example is given in Figure 5.12, for which *AC1* was originally routed along the southern taxiway. However, node *B* might lose the auctions along this route, such that a route command should be possible to reroute *AC1* along the northern taxiway. The number of route updates an aircraft can receive per time step is set to one, to improve route stability and limit the amount of communication between agents. Additionally, it is important to mention that for route changes d_{res} is measured from the aircraft's current position, as no conflict should be created by switching route. For hand off d_{res} is measured from the aircraft's next node, as it is required that a hand off can take place without causing problems for the upcoming ATC agents.

Stationary aircraft

For both hand off and route changes, all routes within the reservation distance d_{res} from the aircraft's next node are reserved. However, at this point the aircraft is still a distance away from its next ATC agent, meaning the total distance that is reserved can be up to $d_{\text{res}} + d_{\text{sep}}$ metres away. In case the aircraft is stopped ($v < 0.05$), e.g. due to congestion on any of the ATC's other outgoing links, it is decided that all reservations further away than d_{sep} are released. This is done to block as little segments as possible, such that the congestion can be solved more easily.

Auction process

If a reservation is placed on an intersection node, also the outgoing link the aircraft intends to use next is being reserved. As the complete reservation relies on these two aspects, it is decided that the reserved agent cannot lose this segment in an auction. Additionally, when a reserved node initiates an auction, the value of the reserved aircraft is taken into account, even if the aircraft is not under the control of the node yet. As it is known which segment the aircraft will use, the full aircraft's value is added to the bid. In case this bid spreads further throughout the network, a path is freed for the reserved aircraft.

5.5.2. Improvements Runway Schedule

Penalty factor

The penalty factor f_{penalty} has been tested for different values. ATC agents have to be timely stimulated to create routes that no longer cross the runway, but drive around it instead. As the segment connecting two stopbars is only a few hundred meters long, a large factor is needed to make up for the extra kilometres required for a detour. Smaller values of f_{penalty} resulted in route changes that are made too late, which is why this parameter is set to 500.

Time horizon departing aircraft

In an earlier stage, all arriving aircraft as listed in the runway's source schedule and all departing aircraft heading towards the runway were taken into account to determine the runway's future usage. An approximation of the remaining taxi time is used to make an estimation when the departing aircraft reaches the runway. This estimation is based on the segments' weights in the taxiway graph and thereby on the historic taxi speeds on these segments. Since insights into future congestion are unknown, this taxi time estimation is rather unreliable. To limit uncertainty, it is decided to only include departing flights with a taxi time below 10 minutes. Although this positively effects the schedule with future runway use, more schedule stability is required. Analysing all relevant traffic at every time step dt resulted in many tiny time changes in the schedule. Additionally, stopbar agents were constantly adapting the stopbar segment's weight to match the start time of the runway's departure mode of operation. The updating algorithm is now triggered every 10 seconds.

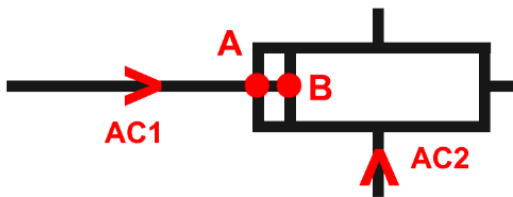


Figure 5.12: Example of operation involving short segments

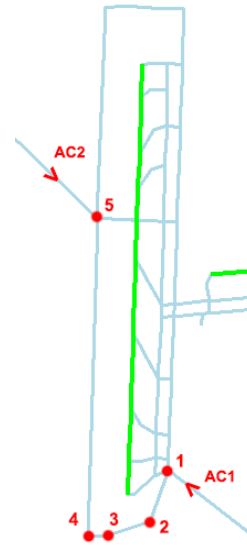


Figure 5.13: Auction process around surrounding runway 18C – 36C

Direction of use

The direction of runway usage is determined based on the flight's runway entry or exit: departing aircraft are likely to take a runway entry in the vicinity of the runway's starting point, while the runway exit of arrivals is expected to be close to the runway's endpoint. For a few runways, the first runway exit is closest to the runway's starting point. Some of the smaller aircraft types at AAS use this exit, which is why these exits are also the origin of the simulated counterpart. This issue caused many runway direction changes, which also led to stopbar segments being added and removed regularly. It is decided that the most common direction of the next 10 flights is selected as main direction for the upcoming blocks of operation

5.5.3. Improvements Auction Protocol

An auction protocol allows ATC agents to coordinate about the future use of resources. Appendix C describes the baseline protocol, as well as a few bugs have been identified and corrected.

Scope of coordination

The layout of AAS has multiple long streaks of single lane taxiways without any side branches. Two clear examples are the northern and southern taxiway paths surrounding runway 18C – 36C as seen in Figure 5.13. Safety measures are in place to make sure aircraft can only enter from one side simultaneously. In case a ATC agent has only one outgoing segment left, this outpath is forced by removing the opposing link. As this will trigger the subsequent agent to do the same, this outpath propagates up to the intersection where multiple taxiway segments come together. This is done to make sure no gridlocks occur.

The implemented auction protocol does not propagate this way, as the baseline model used to work with a fixed scope of coordination $x_{\text{coord}} = 2$. This means that the auction process travels up to an agent at a geodesic distance of 2, which is found to be insufficient for the layout of AAS. It is possible that two two agents are successfully performing auctions on two sides of a long stream of single taxiways, without realising the intention of the other agent.

An example is shown in Figure 5.13. The moment aircraft AC1 receives a hand off to enter segment 1 – 2, the forced outpath propagates until segment 4 – 5. So far, there are no auctions added in case the third auction takes place on segment 1 – 2. However, this could mean that the bid of an aircraft coming from the left, AC2 wins segments 5 – 4 and 4 – 3, while an aircraft coming from the right wins segment 1 – 1 and 2 – 3. Both aircraft believe they can use the shortest route towards their destination and move forward accordingly. Only the moment one aircraft gets close enough to cause forced outpaths along all segments between ATC agents 1 and 5, the other aircraft realises it is too late. This means coordination is lacking along a lengthy streak of single lane taxiways.

Based on the example discussed above, it is clear why one would like to bid on the full single lane taxiway to be sure it is being used in the most preferred direction. It is therefore decided that the fixed scope of coordination is being used as bare minimum. In case the last auction of the scope of coordination is with respect to an ATC agent with only two neighbours (so a corner or stopbar), it is decided to extend the auction process up until the moment an auction takes place with an ATC agent with at least 3 neighbours. Auctions on earlier segments are only won in case the previous auction is won, otherwise all earlier segments are considered to be lost. By allowing auctions to take place on one more segment after the long single lane taxiway, the protocol makes sure aircraft can continue their route.

Expiration lost auction

In case the agent that initiated the auction wins a link, it is added to its list of won segments and removed from the list of lost auctions if necessary. The inverse steps are repeated for the auction receiving agent. However, if the auction initiator loses, it is removed from its win list as well as from the lost list of the receiver. The decision to not add the segment ID to the loss list of the initiator and won list of the receiver, is taken from the point of view that it is unsure whether the auction receiving agent is actually willing to use it. Especially in congested parts of the airport layout, it is possible that an ATC agent loses multiple auctions. If all lost auctions are registered, all ATC intersection would have limited freedom to find suitable aircraft routes.

It is also possible that segments are won by the auction initiator, but not used in the end. This auction loss remains registered in the auction receiver, until this agent wins the link in an auction that takes place in the future. For each aircraft under control, the ATC agent is allowed to win only one auction. If this auction is always won on one of the other outgoing segments this means the lost segment cannot be used for future route updates, even when the segment is not used in the opposite direction.

A time stamp is added to the registration of lost and won auctions. Every time the ATC agent loses an auction on a particular segment, the time stamp is being updated. In case a lost auction has not been updated during the past thirty seconds, the segment is removed from the lost list in order to free it for future usage.

Taxiway closures

As explained in Appendix C, each ATC agent possesses a dictionary called *link_times* which knows the unimpeded taxi times towards each of the destination. This dictionary is used to determine an agent's bid, and is created during the simulation initialisation.

A feature has been implemented in Subsection 5.4.6, which enables stopbar agents could close taxiway segments due to the RMO in use. Since the taxiway layout is altered due to the long-term removal of a segment, some of the stored taxi times are no longer representative. Consequently, it was observed that ATC agents placed large bids in the direction of closed segments. Although Dijkstra's algorithm would recognise the closed segments and thereby select a different route, it is a loss of auction power. That is because an ATC agent is only allowed to win one auction per aircraft it has under control. The stopbar feature which is responsible for managing taxiway closures will now also trigger each ATC agent to update their *link_times* dictionary. For this update, the original dummy graph is used in which the closed stopbar segments are removed. Due to these *link_times* adjustments, more effective auctions take place.

5.5.4. Improvements Routing Algorithm

Forbidden turns

While simulating, it is found that aircraft made turns that are impossible or not allowed in practice. Looking at Figure 5.14, it is seen that runway exits are often connected to the main taxiway layout via a segment that is placed under a sharp angle. Aircraft leaving the runway via one of these exits are not allowed, or capable, to take this first sharp turn. A few of these turns are highlighted in red. A list of forbidden turns has been prepared and entered into the simulation. Each of these turns is being removed from the taxiway graph to be sure they can no longer be included in routes.

Starting point route

An ATC agent is responsible for finding the fastest routes for all aircraft under its control. In the baseline model, an ATC agent focused specifically on the fastest route to get from itself towards the aircraft's destination; no safety nets or conditions were in place to stop an impossible or unpractical route from being sent towards an aircraft. The most important examples are explained briefly.

The most striking example are routes that require an aircraft to turn around and move in the opposite direction. Due to congestion or other aircraft blocking particular segments, the fastest way to go from the ATC

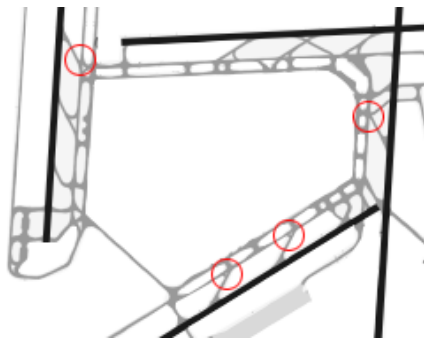


Figure 5.14: Sharp turns that are present in the airport's taxiway layout

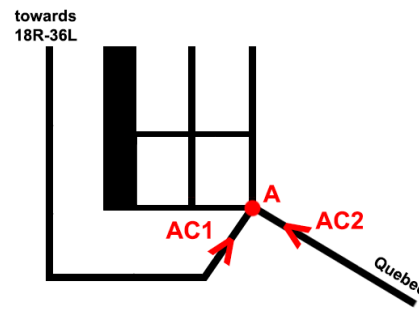


Figure 5.15: Aircraft willing to use each other's link

agent towards the destination is via the aircraft's current segment, requiring the aircraft to turn 180° degrees. Normally this segment should be removed since the aircraft is taxiing across it, but it is temporarily added in case the ATC agent has won an auction for future usage.

Whenever an ATC agent calculates the fastest route, it sets itself as origin and the aircraft's destination as end point. However, without taking the aircraft's current segment into account, vital information is missing:

- The forbidden turns, as explained above, cannot be excluded.
- Based on the aircraft's current segment and location, the ATC agent has to decide whether the aircraft is potentially able to slow down and make a turn. This aspect is required when considering which routes are feasible.
- The original taxiway graph has been transformed into a dummy graph, as can be read in Appendix D. Penalties are assigned to segments in case it requires an aircraft to make a turn, resulting in a more realistic weight assignment for each route. Routes are no longer based on the shortest route, but on a simple estimation of taxi time instead. If an aircraft's next ATC agent calculates the fastest route from itself towards the point of interest, no potential turns are included to arrive at the route's first segment.

Based on all these reasons, it is decided that an ATC agent should not initiate a new route from itself, but from the aircraft's previous ATC agent instead. In this way, all earlier mentioned issues are solved.

5.5.5. Improvements Conflict Solving Using each other's route

At crossroads, multiple aircraft might be waiting at different segments to receive their hand off call. Regularly, it happens that an aircraft has to wait until an other aircraft has been handed off and thereby frees its segment. In rare cases, two aircraft want to use each other's segment, which will not come available as they are waiting for each other. Beforehand, a feature was implemented with the aim to prevent such situations from happening: after an auction, the ATC agent checks which segments are won and which aircraft handed in a bid to win this segment. If there is already an incoming aircraft on a won segment, which has the intention to use one of the segments where a bid came from, this won segment is being removed from the list. By doing so, no route updates are given that would create a conflict with two aircraft waiting endlessly for each other.

Still, this situation occurred rarely during simulations, due to earlier route commands which cannot be updated. An example is shown in Figure 5.15. A runway reconfiguration requires runway 18R – 36L to change from an arriving mode, to a departing mode of operation. Aircraft AC1 is the last arriving aircraft coming from the runway and has the intention to taxi in northern direction to reach its apron. Simultaneously, AC2 is the first aircraft that has to depart from runway 18R – 36L. It is currently planning to cross runway 18C – 36C as there is temporarily no traffic on this runway. Just before both aircraft reach node A, the traffic situation is changed: runway 18C – 36C will be used soon, causing the weight of the crossing segment to increase. Simultaneously, AC1 cannot use the northern route due to congestion, so it has to take the southern route using taxiway Quebec. So the moment the auctions take place, there is no conflict of interest for the won bids. However, new routes are determined, requiring both aircraft to use each other's segment. Unless a faster route is found, which allows one aircraft to leave the conflict, both aircraft will be waiting endlessly.

A new feature is implemented, which allows ATC agents to recognise these conflicts. When such a conflict occurs, the ATC agent aims to clean one of the blocked segments with incoming traffic, by forcing the aircraft

to taxi the smallest circle possible before returning to the ATC agent. This means the agent forces a small routing in advance of the aircraft's current route to deconflict traffic, such that both aircraft can follow their intended route afterwards. By doing so, room is made for the other aircraft to pass and thereby the conflict is solved. As the routing algorithm aims at finding the fastest path for individual aircraft, it does not consider detours or driving circles to solve a conflict between different aircraft routes.

Aircraft movement

Aircraft are partly responsible for preventing conflicts with other aircraft. They can recognise different sorts of conflicts, including situations where it trails another aircraft, crosses a node, or when another aircraft is heading towards the same node. Initially, an aircraft checked whether it was still able to come to a full stop without infringing the separation distance, using the comfortable deceleration. If not, the aircraft would brake, otherwise the aircraft could accelerate.

When trailing another aircraft, the conflict trade-off is based on both their own and the other aircraft's current speed. In case no potential conflict is noted at time t , the other aircraft might decelerates while itself decides to accelerate. Consequently, it might be too late to make the decision to brake at time $t + dt$. It is therefore that aircraft should always consider the worst case will take place in the upcoming time step. This means the aircraft itself accelerates with acc_{com} , while the other aircraft decelerates with dec_{max} .

Aircraft at standstill were only allowed to accelerate in case they met the condition they could stop in time when travelling at full speed. Especially when aircraft are waiting in a queue, this results in larger spatial gaps than required. In accordance with the previous adjustment, it is decided that aircraft are allowed to increase speed from standstill in case they are able to stop in time when travelling at a speed of $acc_{max} \cdot dt$.

5.5.6. Improvements Taxiway (Dummy) Graph

The taxiway graph is a representation of the taxiway layout, which stores the state of the network. A dummy graph is derived from it, and is used as input to determine the fastest route. Both graphs are explained in detail in Appendix D, while this subsection describes an important improvement that has been made.

Maximum weight

The weight assigned to an edge in the dummy graph is an estimation of the expected taxi time. Each ATC agent is responsible for determining the average speed over the last minute on each of its incoming segments. This is done by taking snapshots of the aircraft's speed on each of these segments, with an interval of 10 seconds. In case there are no aircraft at a segment, the speed is set to v_{max} . In case the average speed is zero due to stationary aircraft, the segment's weight is set to infinity. As a result, this segment is excluded when determining the fastest route, unless no other options are available.

While simulating, it is found that occasionally aircraft are queuing up on taxiway Quebec (see Figure 2.1) while waiting for take off on runway 18C – 36C. Consequently, this segment's weight increases up to infinity. This causes route updates requiring other aircraft to travel counterclockwise around AAS to reach runway 18C – 36C, even for aircraft that are about to enter taxiway Quebec. In case the queue starts moving forwards, the segment's weight decreases to an acceptable level, initiating a new set of route updates in which taxiway Quebec is included. Due to this form of route instability, aircraft taxi in circles as route commands send them back and forth. This issue is resolved by making sure weights never reach infinity, which is achieved by setting the speed of segment i equal to $v_i = \max(\min_speed_frac \cdot v_{max}, \bar{v})$. This minimum speed fraction \min_speed_frac is set to 0.1.

Turn speed

For the dummy graph, different weight calculations are in place as explained in Appendix D. The distinction is made whether a turn is required to get from one segment to another. For the baseline layout as shown in Figure B.2, it is sufficient to state that a turn is required in case the heading of the two segments deviates more than 1 degree. This is done to exclude rounding errors. However, AAS's consecutive segments do not line up perfectly. As a result, aircraft are forced to slow down to turn speed v_{turn} when two consecutive segments deviate only a few degrees in heading. It is therefore decided that no slow down to v_{turn} is required in case the heading deviates less than specified by the parameter No_turn , which has been set to 20° .

5.5.7. Additional Adjustments

Handing off sequence

To decide which aircraft is to be handed off first, an ATC agent initially sorted all aircraft under control based on their current taxi time. Priority is given to aircraft that are close to their destination, which corresponds with ATC's line of reasoning to have as few aircraft in the network as possible, in order to reduce workload. However, this led to major negative consequences around runways, as arriving aircraft had to wait a longer time before leaving the dangerous runway area. This directly impacted arrival delay, as the runway had difficulties to reserve space required for arriving aircraft to touch down. Instead, it has been decided to let ATC agents monitor the time an aircraft is under its control and to sort aircraft based on this duration.

Forced outpath

ATC agents are able to recognise when all but 1 segment are either in use by incoming aircraft or reserved by approaching aircraft. It decides to claim its last available outgoing segments which is called a *forced outpath*. By doing so, the opposing edge is removed and thereby the ATC agent makes sure that no gridlock is being created due to an aircraft entering this last available segment. Consequently, all incoming aircraft are always able to continue their route. However, it is found that the feature that checks whether an outpath has to be forced is activated only once every time step. It occurred that during one time step, multiple neighbours either reserved, or handed off an aircraft on one of the ATC agent's incoming segments. Consequently, the next moment the *forced_outpath* was triggered, it realised no more options were available. This implementation has been altered, such that the function is triggered every time a segment is reserved, or a hand off from one of its neighbours takes place.

Normally, an ATC agent would wait to hand off an aircraft until the next segment has at least twice the separation distance available. This would mean the aircraft can enter the segment and free the ATC agent by a distance of at least the separation distance, such that it does not block the crossing for others. This condition is also in place when an outpath is forced. It has been decided that in case of a forced outpath there is no other way or direction to move in apart from the forced outpath. Consequently the first aircraft in line is allowed to enter this segment whenever there is room available. As congestion is often the cause of a forced outpath, a more compact traffic scenario is beneficial as it leaves room for other aircraft to bypass the congestion.

State of taxiway network

Each ATC agent has to be omniscient in terms of the current state of the taxiway network. This aspect is required due to the complexity and dimension of AAS. As a clear example, aircraft departing from an apron on the right side of the taxiway layout towards runway 18C – 36C or 18R – 36L should know whether they should taxi via the northern taxi routes or southern ones. Without (limited) information of the direction of the single lane taxiway Quebec, aircraft agents could be guided towards this taxiway as they believe it is available. In case taxiway Quebec is used in opposite direction, this requires the aircraft to make a full detour surrounding AAS. Another example is due to congestion surrounding a runway entry, as it has been mentioned that the model does not create nicely structured runway queues. As a consequence, aircraft might be guided towards a congested area, which could take a long time to be resolved.

5.6. Simulation Parameters

A broad range of parameters are taken over from the baseline model, while many more have been added while extending and improving the model. This section provides an overview of the most important parameters that have been used in the simulations.

There are a few parameters that are used by all aircraft agents, ATC agents, and the environment. An overview of these parameters is shown in Table 5.1. One parameter that requires more information is *min_speed_frac*. As explained in Subsection 5.5.6 it is decided to make sure a segment's weight is capped by introducing the applicable speed v_i in determining a segment's baseline weight. This weight is calculated by $w_i = \frac{s_i}{v_i}$, where s_i is the segment's length, and v_i equals the maximum of the average speed \bar{v} and $min_speed_frac \cdot v_{max}$. The parameter *min_speed_frac* defines which fraction of the speed v_{max} is used as minimum speed when calculating the segment's weight. Higher values of this parameter leads to higher values of v_i and thereby a lower weight penalty.

Table 5.1: Specification of the general parameters

Symbol	Description	Value
dt	Simulation time step	1 second
d_{sep}	Separation distance	150 metres
no_turn	Turn angle for which aircraft do not have to slow down	30 degrees
min_speed_frac	Fraction of the taxi speed that should be taken into account when calculating a segment's weight	0.10

Table 5.2: Specification of the aircraft dynamics

Symbol	Description	Value
v_{max}	Maximum taxi speed	30 knots
v_{turn}	Maximum turn speed	10 knots
acc_{com}	Comfort acceleration	0.5 kts/s
dec_{com}	Comfort deceleration	-1.5 kts/s
dec_{max}	Maximum deceleration	-10 kts/s

Table 5.3: Specification of the Air Traffic Control agents

Symbol	Description	Value
x_{coor}	Scope of coordination	2
t_{arr}^{runway}	Runway occupancy time for arrivals Runway arrival throughput of 50 aircraft/hour	72 s
t_{dep}^{runway}	Runway occupancy time for departures Runway departure throughput of 60 aircraft/hour	60 s
t_{cross}^{runway}	Runway occupancy time for crossings Runway crossing throughput of 20 aircraft/hour	180 s
t_{arr}^{apron}	Apron occupancy time for arrivals Apron arrival throughput of 120 aircraft/hour	30 s
t_{dep}^{apron}	Apron occupancy time for departures Apron departure throughput of 120 aircraft/hour	30 s
t_{hor}^{runway}	Time horizon within which a runway includes departing aircraft in its schedule of future modes of operation	300 s
t_{future}^{runway}	Defines the frequency for which the runway agent estimates its future usage	10 s
t_{margin}^{runway}	Safety margin added before and after a flight's expected time of runway usage	30 s
$t_{between_modes}^{runway}$	Defines the maximum time allowed between two flights of same runway mode of operation to consider it as one block	300 s
t_{reset_bids}	Time after which lost bids are removed	30 s
$t_{margin}^{stop\ bar}$	Safety margin used by stop bars to determine whether departing aircraft can enter the runway	120 s
$f_{penalty}$	Penalty factor used to increase weight of stop bar segment in advance of a runway becoming active	500

The aircraft dynamics are characterised by a set of 5 parameters, which are shown in Table 5.2. Two types of deceleration are defined, namely the comfort deceleration dec_{com} and maximum deceleration dec_{max} . The former is used for normal operations, e.g. slowing down to make a turn, while the latter is used for emergency scenarios in which immediate and extreme actions are required to maintain a safe operation. All five constants are taken from Udluft's research [73], of which the former 4 are based on position data at AAS airport and the latter is set to the maximum pedal braking as listed by the Flight Safety Foundation ALAR [61].

Multiple parameters are defined for the different types of ATC agents, including the occupancy times for both the apron and runway. Additionally, parameters related to the auction process and removal of the stop bar segment are shown in Table 5.3.

6

Model Analysis and Validation

The previous chapter explains how the model has been developed based on observations of, and comparisons with the actual operation. The traffic samples that have been used so far, consist of small subsets in which different scenarios in terms of runway (re)configurations are included. To find out how well the model behaves in terms of simulating the ground operations, a case study is performed to validate the model. Information regarding the set-up of the experiments is given in Section 6.1, after which the results are presented in Section 6.2. The robustness of the model is analysed by means of a sensitivity analysis in Section 6.3. The overall performance of the model is discussed in Section 6.4. Section 6.5 provides a few recommendations regarding further improvements and extensions.

6.1. Experiment Set-up

To gain insights of the model's performance, a large set of operational conditions has to be simulated. A case study of Amsterdam Airport Schiphol (AAS) is performed, as this gives this research the opportunity to determine the model's strengths and limitations based on a detailed comparison of the simulated ground operations with respect to the actual operation. This section provides information regarding the simulation set-up, as well as the data of AAS that is used to perform this case study. A description of the traffic scenarios and runway reconfigurations per day is given as well.

To simulate the ground operation for a large variety of traffic scenarios, it is decided to use 14 consecutive days of traffic. Not only does such a data set includes a large variety of weather conditions and thereby runway (re)configurations, it also means that different traffic demand is covered in terms of weekdays. The first 14 days of May 2016 (1st of May - 14th of May) are selected, as this time period does not include the runway maintenance that is performed in the weeks before and after it [24, 62]. Additionally, these two weeks are within AAS's summer period, meaning that a busy period is selected. As AAS allocates slots for the complete period, the flight schedule for the period of May and e.g. July are not that much different.

It is decided to focus only on the time period 05:00-23:00 for each of the 14 days, as the number of flights taking place during the night is limited. Consequently, simulating the traffic during the night is not of interest, as almost all of these aircraft are able to travel their optimal route.

An overview of the characteristics of these days is found in Table 6.1. The number of flights corresponds to the amount of flights that have been filtered, cleaned and processed successfully, as described in Section 4.3. It can be concluded that the average number of flights found is around 885. However, the average number of flights per day in May 2016 has been 1,400 [25]. This difference is caused by aircraft that do not have an Automatic Dependent Surveillance-Broadcast (ADS-B) transmitter, as is explained in Subsection 4.3.5.

It is found that almost no flights take place between 23:00 - 05:00. As it is so quiet during the night, almost all aircraft can travel the shortest path towards their destination. It is therefore decided to only include the flights that take place between 05:00-23:00. The number of flights that are being simulated is shown in the fourth columns, which results in an average of 850 flights per day. The small decrease in flights from 885 to 850 justifies the assumption to exclude the night flights.

Table 6.1: Characteristics of selected days

Date	Number of flights	Number of simulated flights	Number of runway reconfigurations
01-05-2016	826	782	19
02-05-2016	923	879	18
03-05-2016	960	914	14
04-05-2016	855	824	17
05-05-2016	910	888	22
06-05-2016	963	932	12
07-05-2016	899	853	20
08-05-2016	947	910	17
09-05-2016	847	817	13
10-05-2016	853	826	13
11-05-2016	845	822	17
12-05-2016	815	794	17
13-05-2016	837	809	19
14-05-2016	876	846	18

The included dates provide a wide range of traffic scenarios. A large variety of winds is included, causing each of the 6 runways at AAS to be used at least once for both take off and landing. For 11 out of the 14 selected days, the wind conditions are stable. This means that the direction in which the runways are being used remains fixed, while the number of runways and runway mode of operation is altered. The other three days (4th, 10th and 11th of May) have a changing wind condition from North to South, or vice versa. The result is that runways that are used as landing in one configuration, might be used for take off after the reconfiguration. Former air traffic controllers have listed this type of reconfiguration as one of the most challenging ones, due to the conflicting traffic. The operational disturbance is strongest on the 4th of May, as 4 reconfigurations take place within 40 minutes to cope with the changing weather conditions. An other very interesting case to test the model's performance is related to the operation that takes place on the 3rd of May, when AAS performs a mixed mode operation on one of its runways. This means that a runway is used for both arriving and departing aircraft simultaneously.

The last column in Table 6.1 provides insights in the number of runway reconfigurations that have been applied at a particular day between 05:00-23:00. It can be concluded that the ground operation is disrupted many times during each of the included days.

Four Key Performance Indicators (KPIs) have been identified to analyse the performance of the simulated ground operation with respect to the actual operation. The selected KPIs are:

- Taxi time: the taxi time provides an indication of the efficiency of the operation.
- Taxi distance: the taxi distance is an important parameter to evaluate the performance of decentralised control. As the local agents have more freedom to guide an aircraft towards its destination, differences in the distance can be found as a result of shortcuts, or detours to avoid congestion.
- Average taxi speed: the taxi speed is included for two reasons. First of all, the aircraft dynamics are one of the most vital steering parameters of the operation. It is therefore that a sensitivity analysis is performed including a comparison with respect to the actual operation in Subsection 6.3.3. Additionally, using speed as a performance indicator provides insights up to what extent decentralised control is able to clear a path for the aircraft, or to avoid congestion.
- Average density: density is selected as it provides insight into the general state of the model.

Operational efficiency is often mentioned when discussing the results. However, it is difficult to give one definition for operational efficiency. For example, Air Traffic Control (ATC) aims at maximising the efficient use of runway capacity, meaning that the distance between two consecutive departing aircraft is the required minimum and there is no queue of aircraft waiting at the runway. However, there is no preference for a particular aircraft. For aircraft, an operation is said to be efficient in case it has no delay. Often, the operational efficiency refers to taxi time. This means that efficiency is lost, in case an aircraft is delayed while alternatives options or routes are available. Thereby, efficient taxiway usage refers to route commands and aircraft hand off, for which no alternative option could be selected that resulted in a lower delay on system level.

6.2. Results

This section presents the results of the simulated ground operations. Subsection 6.2.1 discusses the performance at system level. The differences between, and similarities of, the routing strategies of the model and ATC are given in Subsection 6.2.2.

6.2.1. Performance at System Level

This subsection provides insights in the simulated ground operations in comparison to the actual operation. As discussed in the previous section, ground operations are being simulated for a total of 14 days from 05:00-23:00. The numbers in Table 6.1 add up to a total of 11,896 flights to be simulated, during a variety of traffic scenarios and operational circumstances.

It should be noted that the goal of this research is not to mimic the actual operation. Instead, the overall objective is to obtain a better understanding of the mechanisms of decentralised control. By identifying and analysing differences, important insights are gathered regarding the differences between decentralised control and actual ATC, as well as the regarding the Agent-Based Modelling (ABM) technique. After all, differences are even expected, since decentralised control is able to use the resources in a more flexible manner.

Successful flights

One of the key characteristics to test whether the simulation model has been able to simulate all traffic is by looking at the total numbers of aircraft that the model is able to spawn into the network, as well as the number of aircraft that arrived at their destinations. Since the actual flight schedules also include aircraft that start their flight a few seconds before 23:00, it is likely that there are still aircraft on the taxiways at the moment the simulation shuts down.

In the simulated operation, 11,883 aircraft (99.89%) made it to their destination, which is 3 aircraft more than in the actual operation. For four days, the simulation was able to guide one more aircraft effectively towards its destination, while for one day the ground operation had one more completed trajectory. This difference can be explained by the routes and taxi speeds of the last (few) aircraft.

Statistical analysis

To determine whether operational differences between the simulated and actual operation have significance, statistical tests are performed. As the same flight schedule in terms of aircraft, origin, destination and spawn times is used for both operations, paired tests are considered. However, complete data for the simulated and actual operation are required. This means that aircraft agents which do not reach their destination in time in at least one of the two scenarios are removed from the statistical analysis.

It has been checked whether the paired t-test could be used, based on the conditions of normality and equal variances [38]. These conditions are tested using the Kolmogorov-Smirnov test and Brown-Forsythe test respectively. For all tested scenarios, at least one condition is not met, which is why the Wilcoxon signed-rank test is used as statistical test.

Apart from statistical significance, also practical relevance is considered. As the sample size, around 850 aircraft per day, is large and operational differences are likely, statistical significance is probable. To gain insights whether the outcome also has practical relevance, the effect size r is analysed. This measure gives an indication of the magnitude of the results found. Cohen's criteria state that the effect is small, medium or large for values of r larger than 0.1, 0.3, and 0.5 respectively. Any value below 0.1 is considered to be a trivial effect [12].

Box plots

All successful flights (11,883 for simulator, 11,880 for ADS-B) have been taken into account to draw box plots of the taxi time, taxi distance, and taxi speed as is shown in Figure 6.1 until Figure 6.3. Additionally, the aircraft density in the network has been measured every second during the period 05:00-23:00. A box plot of the densities is shown in Figure 6.4.

From a quick look at these figures, it can be concluded that the taxi time in the simulated ground operation is lower than the one of the actual operation. While the mean taxi distance for the simulated and operation is similar, the mean taxi speed is substantially higher in the simulation (+14.8%). The density is slightly lower in the simulated operation. Although it seems logical that for a similar taxi distance and lower taxi time the taxi time might also be lower, a more profound explanation of the individual graphs can be given by playing back the simulations.

Taxi time

The taxi time box plot is shown in Figure 6.1. A Wilcoxon test was conducted to evaluate whether the simulated operation shows similar taxi time compared to the actual operation. The results indicate a significant difference, $z = 47.65$, $p < .05$, $r = 0.31$. The median of the ranks in favour of simulated operation is 4061.5, while the mean of the ranks in favour of the actual operation is 6832.

When analysing the simulations in terms of taxi time, a few remarks can be made. First of all, as the ground operation does not include the apron area, relevant information is unknown. Gate occupancy is an example, as the aircraft's gate is assumed to be available in the simulation model. Every once in a while, an actual aircraft is being directed towards a buffer position, in order to wait for an occupied gate to become available. As a consequence, the taxi time of these aircraft is (substantially) higher, as can be seen from the outliers in Figure 6.1.

It is already noted in Subsection 4.3.5 that the ADS-B tool is not able to differentiate between an aircraft that has to wait on the taxiway as a consequence of its push back, and an aircraft that is stopped in a congested area soon after it has left its apron. In the former case in the actual operation, an aircraft is standing still waiting for the push back truck to disconnect and to receive a taxi clearance, while it can taxi freely in the simulation.

Also, taxi time is related to the applied runway separation time. In the actual operation, this time varies according to the Wake Turbulence Category (WTC) two consecutive aircraft belong to. However, in the simulation model a constant separation time is maintained as no distinction between aircraft types is introduced. When comparing the simulation with the actual operation, it is noted that the runway queues are smaller in the simulation. This indicates that the set separation time is an underestimation of the actual separation time, resulting in a lower taxi time.

Taxi distance

A Wilcoxon test was conducted to evaluate whether the simulated operation shows similar taxi distance compared to the actual operation. The results indicate a significant difference, $z = -29.85$, $P < .05$, $r = 0.19$. The median of the ranks in favour of simulated operation is 3838, while the mean of the ranks in favour of the actual operation is 4442. So although statistically significant, the practical relevance is only found to be trivial.

The simulation model aims at guiding the aircraft along the shortest path (Dijkstra's algorithm), for which the weights of the segments depend on the average taxi speed of the past minute. As a consequence, to bypass slow moving aircraft, the ATC agents try to guide aircraft along routes which might be longer in distance but shorter in estimated time. An example of one of the longest routes in the simulated operation is given in Figure 6.5: multiple aircraft are waiting on taxiway Quebec, to line up on runway 18C – 36C. The weight of this taxiway is very high due to the low average speed on taxiway Quebec. Consequently, the ATC agents guide a departing aircraft travelling from its apron (red dot) towards the runway (green dot) along taxiway Alpha (A) or Bravo (B) as that seems to result in the fastest route.

The implemented route algorithm aims at finding the shortest path by penalising the taxiway segments with a low average speed. As a consequence, Dijkstra algorithm combines characteristic of finding the fastest route, with the characteristic to be sensible for slow moving aircraft. Looking at Figure 6.2, it seems as if these two model characteristics together mimic the actual ground path. However, this is not the case as is found by analysing the routing strategies in Subsection 6.2.2.

Average taxi speed

It can be concluded from Figure 6.3 that the range of average taxi speeds in the actual operation is broader than that of the simulated ones. A Wilcoxon test was conducted to evaluate whether the simulated operation shows similar average taxi speed compared to the actual operation. The results indicate a significant difference, $z = -47.97$, $P < .05$, $r = 0.31$. The median of the ranks in favour of simulated operation is 6981, while the mean of the ranks in favour of the actual operation is 4167. It is therefore concluded that there is a significant difference, for which the practical relevance is medium. The low taxi speeds in the actual operation can be explained by aircraft that are waiting for a long time at a parking spot before their gate becomes available, resulting in a very low average taxi speed for that flight. The high average speeds are related to flights that have a higher speed at the runway exit, while the models has limited the maximum taxi speed to 30 knots. This difference in speed is clearly visible for flights whose gate is close to the runway.

Overall it can be concluded that aircraft have a higher speed in the simulation than in the actual operations, for which different model decisions play a role. One possibility is that the parameters describing the aircraft's dynamics (see Section 5.6) are set too optimistic. This aspect is analysed in the sensitivity analysis in

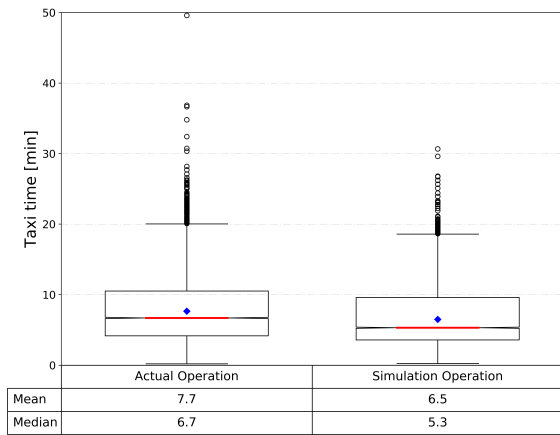


Figure 6.1: Box plot of taxi time

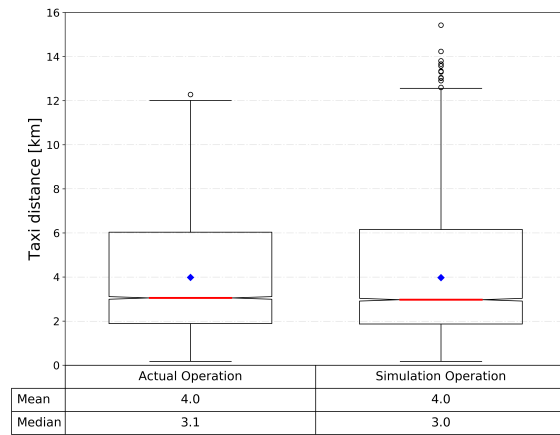


Figure 6.2: Box plot of taxi distance

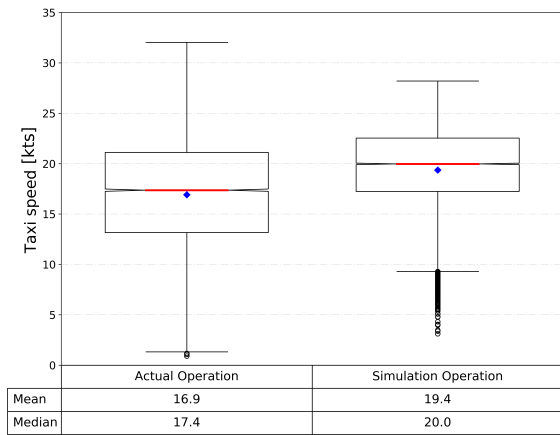


Figure 6.3: Box plot of average taxi speed

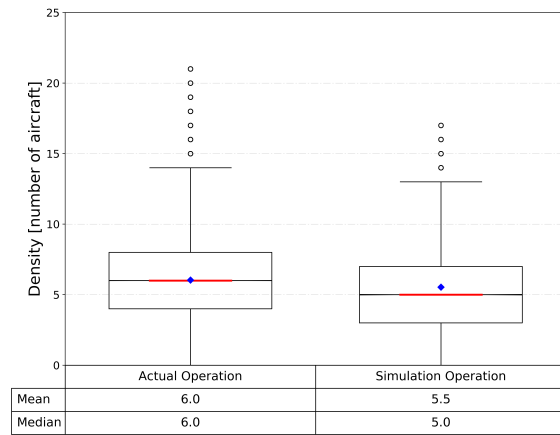


Figure 6.4: Box plot of average density

Section 6.3. As also mentioned earlier, the ATC agents aim at guiding the aircraft along the fastest path which depends on the taxiing speed of all traffic. Consequently, aircraft might take a longer route if this means the aircraft can travel at a higher speed. Additionally, it is specified in the model that aircraft only brake when the required deceleration is above a specified minimum (dec_{com}). So for required deceleration that are small, actual pilots might stop accelerating (or even start decelerating), while the aircraft agents keep on trying to accelerate to v_{max} .

Average density

When analysing the density box plot in Figure 6.4, it can be found that the density in the simulated ground operation is lower than that of the actual operations. A Wilcoxon test was conducted to evaluate whether the simulated operation shows similar average density compared to the actual operation. The results indicate a significant difference, $z = -78.41$, $P < .05$, $r = 0.51$. The median of the ranks in favour of simulated operation is 2505.5, while the mean of the ranks in favour of the actual operation is 6561. The practical relevance of this difference is found to be large. This difference caused by a combination of multiple factors listed before; smaller runway queues, lower taxi times, and the fact that aircraft agents do not wait at a parking spot as the model assumes the gate is always available.

Statistical results

The results of the statistical analysis are shown in Table 6.2, in which the simulated operation is compared with the actual operation in terms of taxi time and taxi distance, as well as the average taxi speed and density. The sample size N represents the number of aircraft agents that arrived in time at their destination, while P and r represent the P -value and effect size respectively. From this table it can be concluded that all performance indicators in the simulated operation are statistically different from the actual operation ($\alpha < 0.05$), apart from the taxi distance for the 3rd of May. All significant parameters are shown in green, where effect

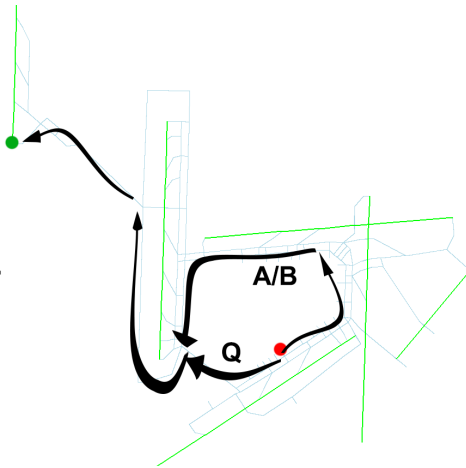


Figure 6.5: Travelling along a bypass, instead of taking the shortest route

Table 6.2: Statistical analysis of the simulated operation

	<i>N</i>	Taxi time		Taxi distance		Average taxi speed		Average density	
		<i>P</i>	<i>r</i>	<i>P</i>	<i>r</i>	<i>P</i>	<i>r</i>	<i>P</i>	<i>r</i>
01-05-2016	781	0.00	0.31	0.00	0.14	0.00	0.34	0.00	0.51
02-05-2016	877	0.00	0.37	0.00	0.22	0.00	0.34	0.00	0.57
03-05-2016	914	0.00	0.24	0.38	0.02	0.00	0.26	0.00	0.44
04-05-2016	822	0.00	0.34	0.00	0.21	0.00	0.33	0.00	0.55
05-05-2016	886	0.00	0.31	0.00	0.24	0.00	0.30	0.00	0.49
06-05-2016	932	0.00	0.30	0.00	0.25	0.00	0.28	0.00	0.53
07-05-2016	852	0.00	0.33	0.00	0.28	0.00	0.31	0.00	0.54
08-05-2016	910	0.00	0.30	0.00	0.21	0.00	0.28	0.00	0.49
09-05-2016	815	0.00	0.33	0.00	0.32	0.00	0.33	0.00	0.51
10-05-2016	826	0.00	0.29	0.00	0.17	0.00	0.31	0.00	0.49
11-05-2016	820	0.00	0.31	0.00	0.24	0.00	0.30	0.00	0.51
12-05-2016	793	0.00	0.34	0.00	0.23	0.00	0.34	0.00	0.54
13-05-2016	807	0.00	0.28	0.00	0.10	0.00	0.34	0.00	0.42
14-05-2016	844	0.00	0.30	0.01	0.06	0.00	0.32	0.00	0.52
Total	11879	0.00	0.31	0.00	0.19	0.00	0.31	0.00	0.51

sizes ≥ 0.3 are considered to be practically relevant. This value is selected based on Cohen's statement that "*a medium effect of is visible to the naked eye of a careful observer*" [13]. It is noted that the average density is both statistically and practically relevant for all 14 days. Additionally, when combining all 14 days of operation, the effect size is even considered to be large, $r \geq 0.5$. Apart from the 9th of May, the practical relevance in terms of taxi distance is only considered to be trivial. This indicates that the route distances travelled in the simulated and actual operation are closely related. This confirms the results as shown in Figure 6.2. Section G.1 lists the hypotheses that have been tested to evaluate the differences between the simulated and actual operation.

6.2.2. Routing Strategies

It is found in previous subsection that the box plot of the average distance travelled looks quite similar for both the simulated and actual operation. This subsection goes into more detail regarding the routing strategies, as applied by both the simulation model and ATC. The routing strategies are determined based on replaying the ground operations and analysing the agents' behaviour. Four different tools are used to provide insights regarding scenarios of interest: routing patterns, performance monitoring, replaying of routes, and measuring route complexity

1. Routing patterns

The routing strategies of both the simulation model and actual ATC are visualised by plotting the most fre-

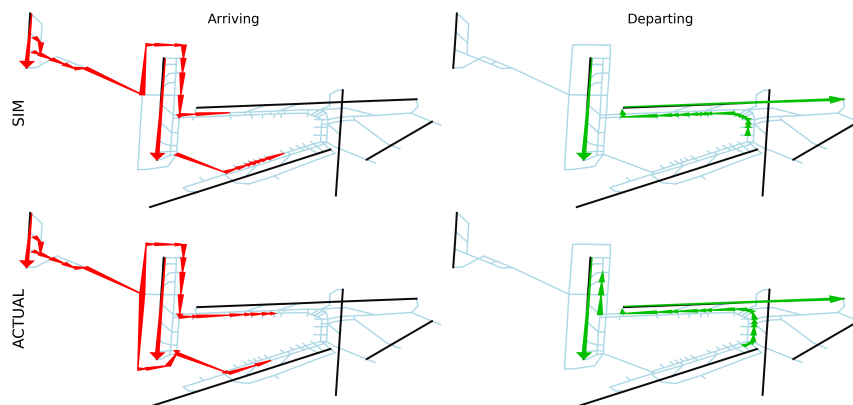


Figure 6.6: Most frequently used routes part 1

quently travelled routes. For both the simulated and actual operation, routes are selected that are used by at least 25% of the arriving and departing traffic respectively. To clearly show how the routing patterns change according to the runway configuration in use, it is decided to create a plot with the routes of aircraft that are spawned during the last two hours, with an update interval of 10 minutes. It was found that smaller windows, e.g. 1 hour, are too limited for more quiet scenarios. Reason for this is that an aircraft with a small delay in one of the two operations might create a false image of the routing differences between the operation. By creating a movie of the individual plots, clear insights are obtained regarding the evolution of the routing strategies. Two interesting examples are shown in Figures 6.6 and 6.7, for the 8th of May and 7th of May respectively. The simulated operation and actual operation are shown in the top row and bottom row respectively. Each red arrow in the left column represents a segment that is travelled by at least 25% of all arriving aircraft. The same holds for departures in the right column.

1a. Initial value problem

In Figure 6.6, it can be seen that the strategies are opposite of each other: where ATC uses the outer taxiway (Bravo, see Figure 2.2) for departing traffic and the inner for arriving traffic, the opposite is visible in the simulated operation (indicated by *SIM*). It is found that the model strategies for particular scenarios are sensitive to the starting variables in terms of the route of the first aircraft that is spawned. The cause of the model's routing strategy as shown in Figure 6.6, is due to one arriving aircraft. The aircraft is spawned during a quiet period, meaning that the aircraft could travel the shortest path towards its apron involving the outer taxiway. As occupied segments are removed when ATC agents determine a route, the next aircraft, which is departing, is guided along the inner taxiway. Once these patterns emerge, ATC agents apply these routing strategies until the field is cleared again, and a new aircraft "creates" the next routing strategy. As can be seen from Figure 6.6, the routing strategy in the simulated operation makes least sense, as the departing aircraft lining up for the horizontal runway have to cross the flow of arriving aircraft. This situation is a clear example of the initial value problem.

1b. Efficient use resources

It is noted that decentralised control is able to make better use of the limited available resources, especially for particular runway reconfigurations. Figure 6.7 shows a clear example, for which the overall traffic flow is going from West to South due to the runway configuration in use. As no traffic is expected to move from right to left, aircraft are able to use both parallel taxiways in the northern region of the airport: one stream for each of the arriving runways. Also, the shortest path algorithm is clearly visible, as the shortest path is used in the southern region of the airport.

2. Performance Monitoring

A second tool used is a graph that plots the taxi time and taxi distance over time. This helpful tool provides insights in spotting scenarios where simulated and actual operation differ from each other. Additionally, the tool helps to determine moments of interest where taxi time is constant, while the taxi distance suddenly rises or drops, or vice versa. Two examples are shown in Figure 6.8 and Figure 6.9, of which the insights resulted in

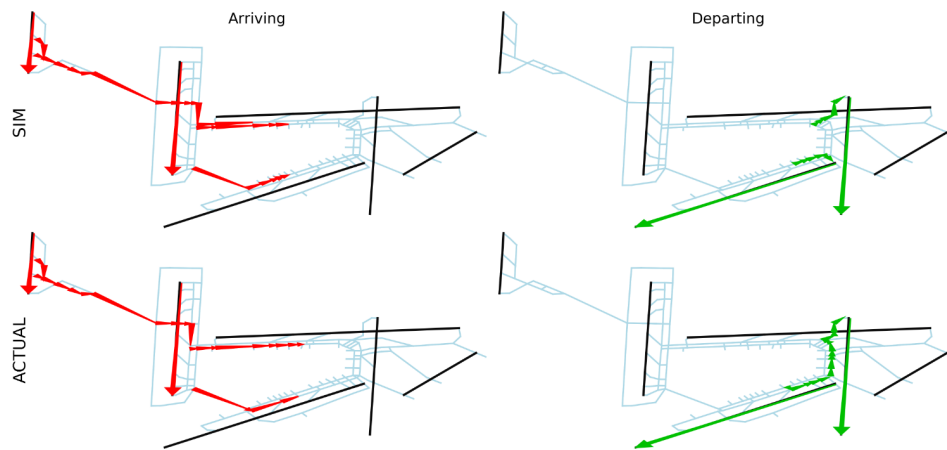


Figure 6.7: Most frequently used routes part 2

two recommendations for future research. The left and right axis show the average taxi time (blue) and taxi distance (red) respectively, for all aircraft that are spawned into the network during the last hour. The solid and dotted line represent the simulated and actual operation respectively, while the vertical lines indicate when a runway reconfiguration took place.

2a. Local goal

From Figure 6.8 it can be concluded that around 07:30-09:00 the taxi time of the simulated ground operation increases rapidly. Figure 6.10 shows departing aircraft (in green) which have to wait in a queue before lining up on the runway above. In the actual operation, a runway queue is created on the taxiway closest to the runway, where aircraft wait (perfectly) in line. However, in the simulated operation each ATC agent tries to send an aircraft as quickly as possible towards the runway entry, which is the local goal of an agent. The global goal of an efficient operation without blocking the arrivals (in red) is not taken into account. Due to the inefficiency in lining up the aircraft, congestion is created, which increases taxi time.

2b. Similar patterns

Figure 6.9 shows the taxi time and taxi distance plot of the simulated and actual ground operations of the 3rd of May. It can clearly be noted that both patterns of the actual and simulated operation are very similar. Consequently, it can be said that the simulation model is able to approach the actual operation. The actual operation has a higher taxi time, which is mainly caused by the aircraft that have to wait on a buffer location until their gate becomes available.

2c. Missing future information

There are two moments in Figure 6.9 when the simulated taxi distance is clearly above that of the actual operation. More insights regarding the difference around 13.30 - 14.30 is shown in Figure 6.11. As no data is shared regarding the future operation in the simulation, departing aircraft in green are routed according to the operation taking place at that time. For the simulation this means aircraft are being directed around the runway 18C – 36C, since the runway is still occupied for the final arrival (red aircraft). In the actual operation, aircraft are waiting in line to cross the runway, once it becomes available, to save taxi time and distance. Overall, it can be concluded that the simulation model lacks information regarding future operational states, which limit the efficiency of the modelled operation. This same explanation can be given to the taxi distance difference early in the morning (05:00-07:00).

3. Replaying of routes

A tool is designed to replay the travelled routes. When analysing the replays of the simulated ground operation next to the actual operation, all the previously highlighted elements are found regularly.

3a. Timely information There is a lack of timely information regarding the availability of crossing runway 18C – 36C, causing the aircraft in the simulation to travel an extra distance to go around the runway. Additionally, it is noted that congestion is created due to aircraft that are guided towards the same runway (or

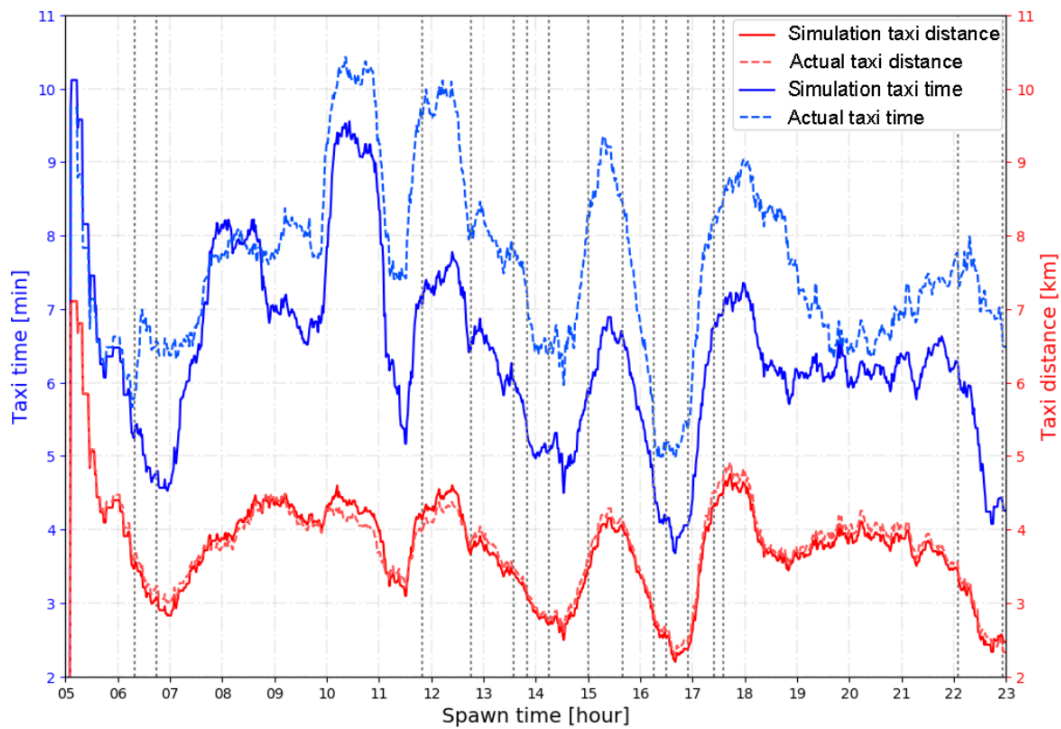


Figure 6.8: Taxi time and taxi distance plot of 9th of May

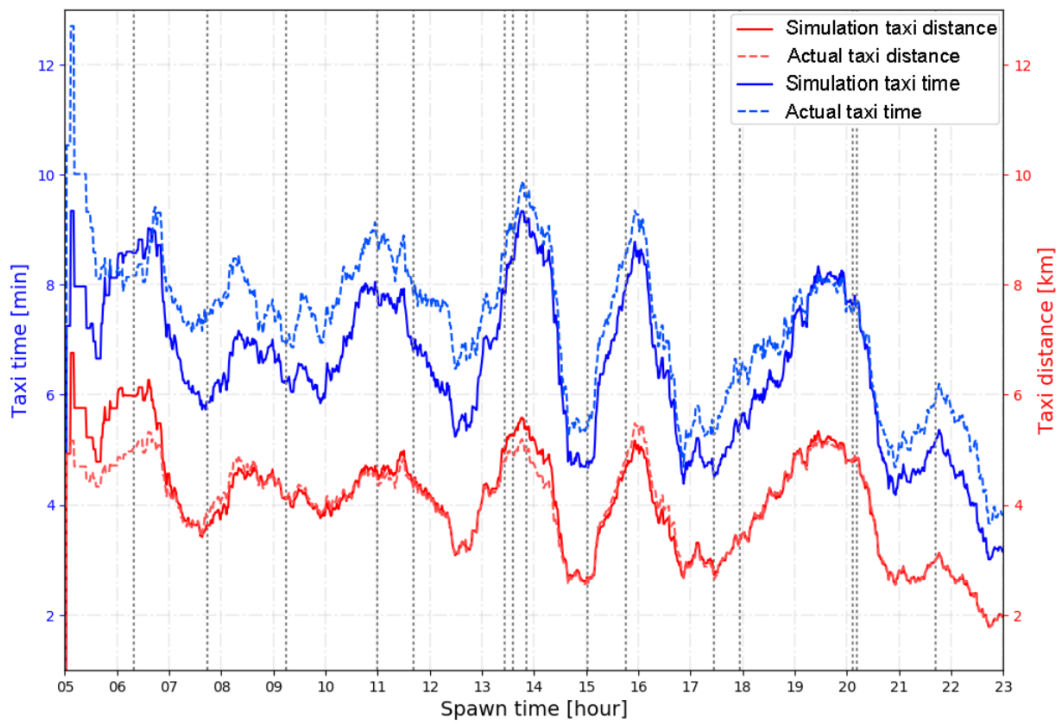


Figure 6.9: Taxi time and taxi distance plot of 3rd of May

apron) node from different sides. Continuing on this latter aspect, the region around runway 18C and taxiway Quebec (see Figure 2.2) is particularly vulnerable for this deficiency. Congestion is expected due to aircraft waiting in line to depart from runway 18C – 36C. However, this also blocks the traffic flow at Quebec that would like to go in southern direction towards runway 18R – 36L. However, the bottleneck created at this location holds aircraft from doing so.

Also, ATC agents have no insights in when a taxiway is planned to be used in a particular direction. Only

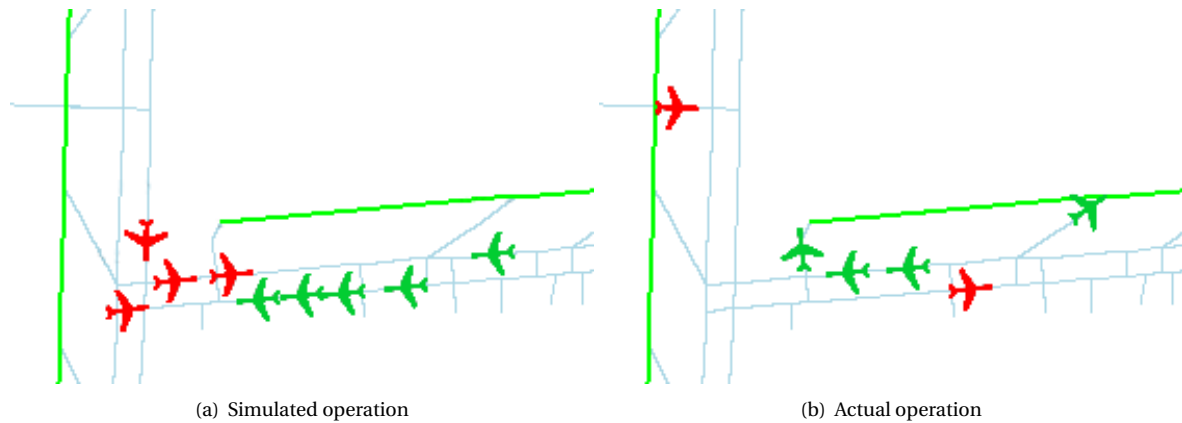


Figure 6.10: Bad runway queue due to missing global goal

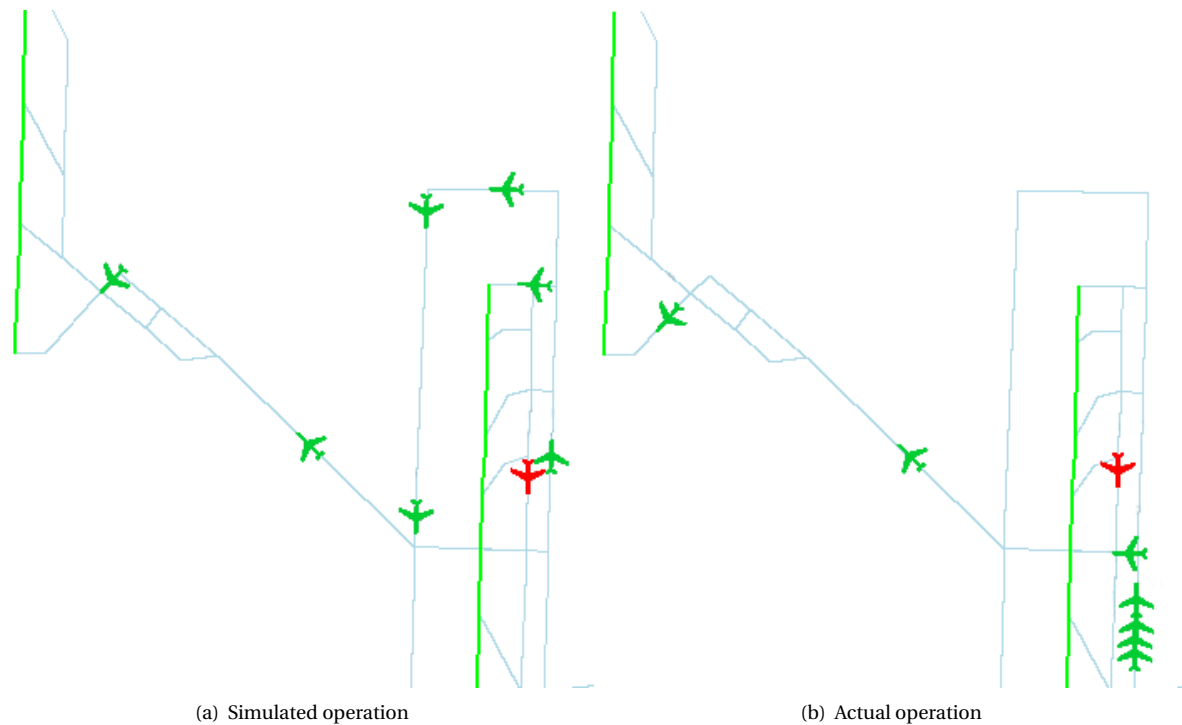


Figure 6.11: Unknown insights into future operation

close by agents have some insights in the expected usage of a taxiway, as they are involved in the auctions on adjacent segments. However, an auction is only triggered in case an aircraft is close by and willing to use it. This means that multiple ATC agents might plan a route using this Quebec taxiway, even if it conflicts with other routes. Whenever an aircraft reaches Quebec and uses it in a particular direction, the ATC agents are notified this segment is no longer available. As a consequence, some aircraft receive a new route which is in the opposite direction of their current route, meaning they have to travel back and thereby travel longer routes than required when more information was known beforehand.

3b. Sensitive stationary aircraft

The model is designed in such a way that aircraft receive routes created from the 'fastest route' perspective, where a Dijkstra algorithm is used in combination with a graph, whose weights depend on the historic taxi speeds. It is explained in Subsection 5.5.6 that a feature is added which should maximise the weight of a segment. Unfortunately, it is found that the shortest path algorithm is still sensitive to slow moving traffic, which rarely results in aircraft making a loop before continuing their path. An example is shown in Figure 6.12. During the simulation, a queue of aircraft is waiting on taxiway Quebec, either to line up on runway 18C – 36C or

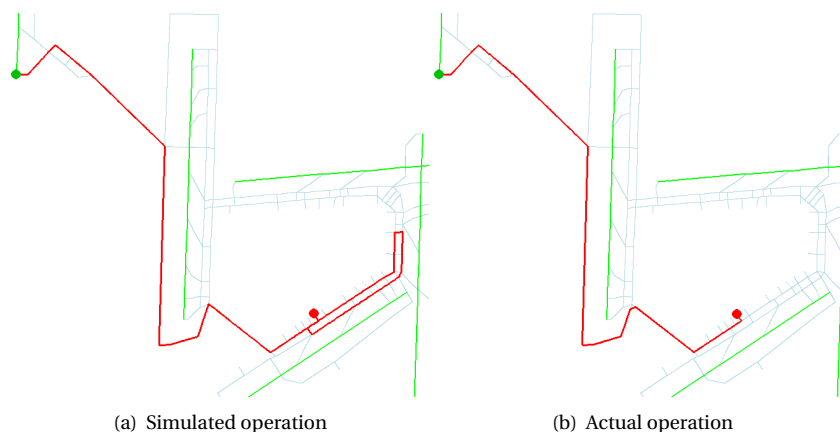


Figure 6.12: Loop in aircraft's ground path

hold due to the aircraft in front. While the actual ATC guides an aircraft from its gate (red dot) to the runway (green dot) via taxiway Quebec (see Subfigure 6.12(b)), the ATC agents put a high weight on taxiway Quebec due to slow moving traffic. Instead, the aircraft receives a (much longer) route to travel around the airport via taxiway Bravo. However, halfway the model realises that aircraft are moving again along taxiway Quebec, such that the best possible route is now via Quebec. This means the aircraft has to turn around, before taking the same route as suggested in the actual operation. It can be seen in Subfigure 6.12(a) that the travelled path is much longer in the simulated operation, due to the fact that the model has no information regarding the expected waiting time of other aircraft. Consequently, information is missing to select an optimal path which creates deficiencies in the network. This is also the main reason why the maximum distances in the taxi speed box plot in Figure 6.2 is higher for the simulated operation; the system behaviour emerges in aircraft making a loop before continuing their path.

Both of the identified limitations can be solved, by altering the way in which a segment's weight is determined. Instead of making them dependent on speed only, they should represent the dynamics over time. If a segment's weight could acclimate to the future traffic state, other agents can create routes that have been adapted for future dynamics.

4. Route Complexity

Another way to look at routing complexity, is by analysing the entropy state of the assigned routes. Two well known entropy measures are Sample Entropy (SampEn) and Approximate Entropy (ApEn), which both measure the complexity of a series of N points [54, 59]. Fundamentally, these complexity measures analyse the probability that two sequences within the serie that are equal for m points, will remain similar for the next point (within a specified tolerance r). It has been decided to use SampEn as developed by Richman and Moorman to analyse the assigned routes in terms of structure and regularity. The benefit of using this complexity measure instead of using the well known ApEn [54] is due to the fact that ApEn is biased due to self-matching. Additionally, SampEn has a clear reference level of zero, while negative values are possible for ApEn.

Each of the assigned routes consists of a series of nodes it has passed. For a particular moment in time, the routes of all aircraft that are spawned during the last hour are grouped and pasted behind one another. By analysing SampEn with values $n = 2$ and $r < 1$, the probability is analysed that if two aircraft travel along the same segment, also their next segment is similar. In case it is, it is more likely that aircraft follow similar routes and thereby enhance route structures route stability. Limited parts of the routes are taken into account, as only the main taxiways are included as shown in Figure 6.13. This means that the runways and its entries/exits, and apron entries/exits are removed, as well as some of the taxiways that lead directly to an apron or runway. These segments are excluded as they do not add any freedom to ATC in terms of routing strategies. For each of the aircraft routes, only the segments are selected that are included in the layout found in Figure 6.13.

Table 6.3 presents the average complexity SampEn for the simulation and actual operation respectively. Overall, it can be concluded that the SampEn is on average 4.3% higher in the simulated operation, meaning the routes are more chaotic. This statement is also confirmed by analysing the most frequently travelled route

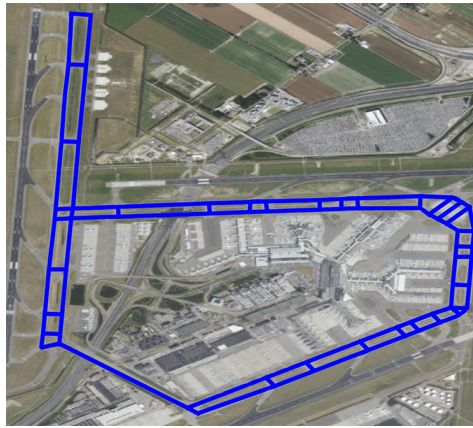


Figure 6.13: Layout used for complexity measure Sample Entropy

graphs in Figures 6.6 and 6.7, as the length and occurrence of these routes are smaller in the simulated operation. This indicates that more variety is present between the different routes in the simulated operation. It can therefore be concluded that the decentralised ATC as implemented in the model is able to handle a more complex ground operation. Although most values on day level look rather similar, clear differences occur during the day. Below multiple examples are given to provide insights in the differences that are found.

For each day, a SampEn graph has been drawn which analyses the route complexity for all aircraft that were spawned during the last hour, with a calculation time interval of 10 minutes. An example of such a graph is shown in Figure 6.14. The orange line represents the SampEn of the simulation, while the blue line is the actual operation. The vertical lines represent a runway reconfiguration, which is derived from the runway database as explained in Section 4.1. It should be noted that there could be a delay between the moment ATC (de)activates a runway and the moment the first (or last) aircraft uses it. According to procedures, a departure runway first has to be activated before an aircraft can receive its push back clearance. Consequently, it takes both push back time and taxi time before the first aircraft takes off from the newly activated runway. Vice versa, when a departure runway gets deactivated, there might still be aircraft that just received a push back and are still on their way to the runway. Depending on the distance between apron and runway, this "time lag" might be more than half an hour. The same holds for landing runways.

4a. More complex departure peak

The most remarkable conclusion that is drawn from analysing the SampEn graph is that the simulated operation tends to be more complex during departure peaks, while the actual operation is more complex during landing peaks. For the departure peak, it might be required to create a runway queue on the main taxiways, depending on the runways in use (e.g. for runway 09, 36C, 24). The chaos in the simulated operation as seen in Figure 6.10 is clearly present in the SampEn graph shown in Figure 6.14 between 07:30 and 11:00. Figure 6.15 shows the effect of stationary aircraft on taxiway Quebec. As mentioned before, the model is sensitive for slow moving vehicles on taxiway Quebec, as the model believes that faster routes might be possible. As long as aircraft are moving on Quebec, including this link seems to result in the fastest route for some origin-destination pairs. However, whenever aircraft have to stop due to congestion, other routes seem more optimal. Loops in aircraft as shown in Figure 6.12 cause SampEn to increase, as is clearly visible between 14:30 – 16:00, 17:00 – 18:00, and 21:00 – 22:30 in Figure 6.15.

4b. Less complex arrival peak

Figure 6.16 shows how the SampEn of the simulated and actual operation alternate depending on the departure and arrival peaks. Reasoning behind the higher SampEn values for the simulated operation during departure peaks has been explained above. However, it is interesting to note that the actual operation is characterised by a higher entropy during the arrival peaks that take place between 08:00 – 10:00, 13:30 – 16:00, and 18:00 – 20:30. Two factors that play a role in the increased SampEn are given in Figure 6.17. Subfigure 6.17(a) shows that there is a clear routing structure in the Southern part of AAS for the arriving aircraft in red, namely the shortest path as programmed in the simulation model. However, such a routing structure is missing for the actual operation. Normally aircraft are routed along the outer taxiway in anti-clockwise direction, however an exception is made for arriving aircraft with one of the blue dots as their destination. As long as

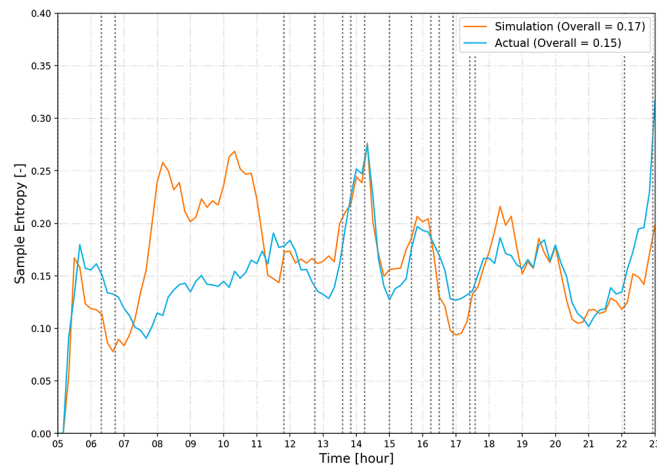


Figure 6.14: Rolling hour of Sample Entropy on 9th of May

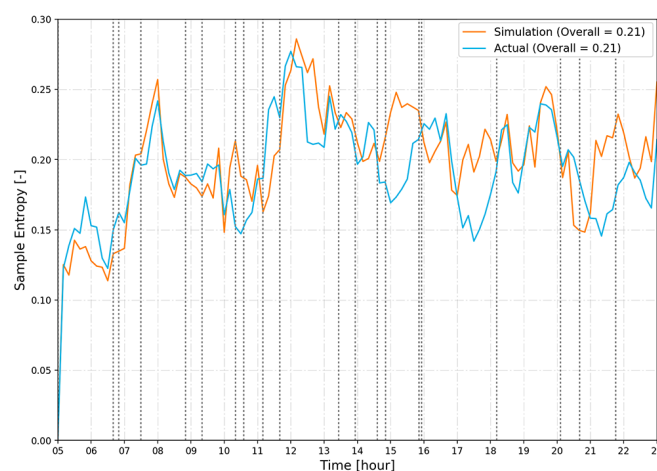


Figure 6.15: Rolling hour of Sample Entropy on 1st of May

this routing can be performed safely without any conflicts, this operation is preferred by both pilots and ATC. Another cause of the higher SampEn values can be seen in Subfigure 6.17(b). As the push back process of some gates or aircraft is on the main taxiways, this segment might be unavailable for some time. ATC can decide to deviate from standard routing strategies to bypass this segment, as shown in Subfigure 6.17(b). As the effect of push backs is not included in the simulation model that same segment is not blocked in the simulated operation, meaning no bypass is required.

4c. Largest differences

From Table 6.3, it can be seen that the SampEn of the simulated operation is much higher with respect to the actual operation on the 3rd and 8th of May. For the 3rd of May, this is caused by the mixed mode operation that is performed on runway 18C – 36C. As a consequence of this mode of operation, there are many aircraft travelling towards, and away from this runway, meaning that there is no optimal direction for taxiway Quebec. As aircraft coming from both sides of the airport are aiming to use Quebec in their preferred route, it happens often that they are just too late and therefore have to travel back and go around the AAS layout. This creates chaos on the taxiways of the simulated operation, which increases the entropy measure.

Regarding the 8th of May, the higher entropy values are explained by the fact that runway 09 – 27 is used as departure runway in eastern direction, between 13:28 and 20:18. As explained before, this requires runway queues to build up on the main taxiways. The deficiencies in terms of sub optimal creation of queues in the simulated operation and related chaos are the cause of the higher overall SampEn.

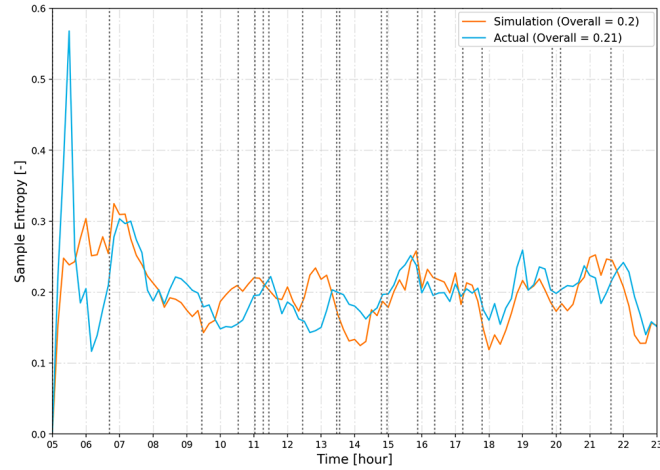


Figure 6.16: Rolling hour of Sample Entropy on 2nd of May

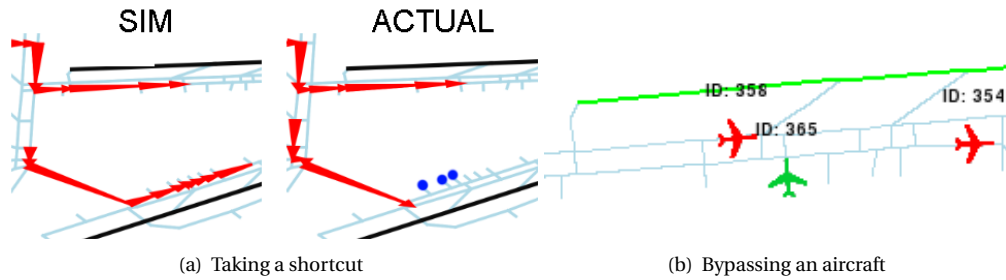


Figure 6.17: Deviation from procedures of Air Traffic Control

Table 6.3: Sample Entropy value per day

Date	Total SampEn Simulation	Total SampEn Actual	Difference w.r.t. Actual
01-05-2016	0.21	0.21	+0.0%
02-05-2016	0.20	0.21	-4.0%
03-05-2016	0.26	0.22	+16.2%
04-05-2016	0.22	0.22	+1.3%
05-05-2016	0.21	0.20	+6.7%
06-05-2016	0.20	0.20	+1.0%
07-05-2016	0.21	0.20	+3.8%
08-05-2016	0.21	0.18	+15.6%
09-05-2016	0.17	0.15	+8.8%
10-05-2016	0.18	0.18	-2.1%
11-05-2016	0.21	0.21	-0.6%
12-05-2016	0.20	0.20	+0.9%
13-05-2016	0.21	0.20	+3.2%
14-05-2016	0.23	0.22	+6.5%
Average	0.21	0.20	+4.3%

6.3. Sensitivity Analysis

The results of fourteen days of simulation are presented in the previous section. A sensitivity analysis is performed, to gain a better understanding of the influence of individual parameters and the robustness of the simulation model. Subsection 6.3.1 explains the method that is applied to perform the sensitivity analysis. Two sets of independent parameters are selected, namely the operational parameters and aircraft dynamics. The performed sensitivity analysis is covered in Subsection 6.3.2 and 6.3.3 respectively.

6.3.1. Process Sensitivity Analysis

Multiple operational parameters have been changed in value to determine the individual impact on the performance of the overall operation. This performance is expressed in terms of the dependent parameters:

- Taxi time
- Taxi distance
- Average taxi speed
- Average taxi density: number of aircraft in the taxiway network

The individual 14 days have been analysed in terms of the above-mentioned 4 performance indicators to determine which operational days are selected for the sensitivity analysis. The 7th of May is selected based on the fact that the simulated and actual operation are quite similar, as well as being an average day in terms of number of aircraft spawned. Additionally, the 3rd of May is selected as it is one of the more complex days due to the mixed mode operation on runway 18C – 36C. Evaluating the impact of individual parameters on both an average and complex day of operation, provides interesting insights in the robustness of the model.

Statistical tests are performed to determine whether the observed operational differences between the baseline scenario and scenario under investigation have significance. Since the normality and equal variance conditions of the paired t-test are not met, the Wilcoxon signed-rank test is used as statistical test. Additionally, the effect size r is calculated to draw conclusions regarding the practical relevance. A few hypotheses have been tested, which can be found in Section G.2.

Three parameters are used to present the statistical results. The sample size N represents the number of aircraft agents that arrived in time at their destination, while P and r represent the P -value and effect size respectively. These three parameters are given for each of the two simulated days, as well as the result of the combined data sets. Each of these aspects are tested with respect to the baseline scenario. Apart from that, the combined results from simulating the ground operation on the 3rd and 7th of May is tested with respect to the actual operation (ADS-B). It should be noted that the sample sizes used with respect to the baseline might differ from the one with respect to ADS-B. Due to the fact that different aircraft have not reached their destination in the selected scenario, baseline scenario or actual operation, different aircraft pairs are removed to conduct the paired tests. Statistical significant results, where $\alpha < 0.05$, are marked in green. Additionally, it is decided to highlight medium and large practical relevance, $r \geq 0.30$, in green as well.

6.3.2. Operational Parameters

Four operational parameters have been selected for the sensitivity analysis:

- $D_throughput$: departure throughput, which is the number of aircraft that can take off per hour.
- No_turn : maximum angle between two segments for which an aircraft does not have to slow down.
- Min_speed_frac : minimum fraction of the speed is used to determine a segment's weight (see Subsection 5.5.6).
- x_{coord} : scope of coordination between the ATC agents, indicating how far auctions are being propagated.

Different values have been selected as input for the sensitivity analysis as can be seen in Table 6.4. Each of these settings have been simulated for both the 3rd and 7th of May. The results of these simulations are presented in Table E.1 until E.4 for taxi time, taxi distance, average taxi speed and average density respectively.

Appendix G provides insights in a subset of hypotheses that has been tested. Since no hypotheses have been defined in advance of the data collection, a Bonferroni correction had to be applied for each tested hypothesis. It should be noted that the results as presented below do not have this correction.

General findings

Before discussing the individual parameters in more detail, general conclusions are drawn regarding the sensitivity analysis of the operational parameters. When analysing the results in Table E.1 until E.3 for taxi time, taxi distance, and taxi speed, it is found that none of the tested scenarios results in a difference, with respect to the baseline scenario, that is practically relevant. The latter is especially clear for the taxi distance, where the practical relevance of all results are considered to be less than trivial. For the average density in Table E.4, it is found that 7 out of the 10 tested cases are statistically significant, while four of them are practically relevant.

A few interesting points can be deduced from the Spearman's Rank Correlation Tables as presented in Tables E.5 until E.8. First of all it can be noted that only 5 of the 16 cases have statistical significance, of

which none of them is related to taxi distance. As a result, it can be concluded that taxi distance is not influenced by any of the four selected operational parameters. Additionally, it is noted that r equals 0.00 for both min_speed_frac and x_{coord} for all four parameters.

Departure throughput

When looking specifically at the departure throughput $D_throughput$, it is found that changes in the throughput are statistically significant on taxi speed, taxi time, and density, but only practically relevant (≥ 0.3) for the latter two. An increase in departure throughput decreases the time between two aircraft agents that are willing to take off from the runway, such that runway queues are expected to be shorter. Consequently the taxi time is likely to be lower, as well as the number of aircraft (density) in the network. Both statements are confirmed by the Spearman's Rank Correlation in Table E.5 and E.8. Additionally, as aircraft agents are spending less time waiting in the queue, taxi time is higher as seen in E.7. No significance is found for the effect of a lower/higher departure throughput on taxi distance, as only a limited number of aircraft agents have to adapt their routings to the longer/shorter queue of waiting aircraft. It should be noted that it is assumed that this finding is not applicable to all runway configurations. This is because for some runways the related runway queue is located on the main taxiways, which thereby has a much larger impact on the overall operation.

When comparing the different scenarios with the actual operation (the final three columns in Table E.1 until E.3), it is seen that each scenario is still significantly different from the actual operation. However, a lower departure throughput results in a practical relevance that is lower than the baseline scenario, providing indications that the time to take off should be increased to get closer to reality.

Turn angle

The parameter No_turn specifies up to what angle a turn can be made for which an aircraft agent does not have to slow down. From Tables E.1 until E.3, it is found that changes in this parameter have statistical significance, but no practical relevance. Practical relevance is only observed for the impact of increasing the allowed turn angle with 10° on density. Comparing the different tested scenarios with the actual operation shows that increasing No_turn , results in practical relevance for both taxi time and taxi speed. The fact that aircraft agents can travel at higher speeds across larger parts of the taxiway layout result in an operation that deviates more from reality. It should be noted that this parameter closely relates to the aircraft dynamics, as the turn penalty also depends on the aircraft's capability to brake. As taxi speeds have such a large effect on the operation, different speed profiles are tested in Subsection 6.3.3.

As a result of the fact that a higher speed is allowed on particular corners, the average speed increases. This correlation is found to be significant as can be seen in Table E.7. Also a reduction in density is found to be significant. The higher taxi speed causes a reduction in taxi time, although this effect is not significant. As a consequence of the fact that higher speeds are allowed on some turning points, some segments are reduced in weight in the Dijkstra algorithm calculations. Although this might result in faster routes as "shortcuts" are more beneficial, it could also stimulate aircraft agents to change a route to bypass slow moving aircraft. This latter aspect explains the minimal increase in taxi distance, although it has no significance (see Table E.6).

Minimum speed fraction

Min_speed_frac refers to the parameter that limits the maximum edge weight. From the Spearman's Rank Correlation in Tables E.1 until E.4, it is found that Rho is zero and insignificant for each of the four indicators. Also from the sensitivity analysis, it can be concluded that the impact of changing min_speed_frac is minimal and not significant. Subsection 5.5.6 explains that this parameter is introduced to remove the effect of loops in an aircraft's path, which occurs for a limited number of flights. These loops are caused by slow moving traffic which causes the weight of the respective segment to increase rapidly. Increasing the value of min_speed_frac decreases the effect but does not completely preclude these loops.

Scope of coordination

The fourth parameter is the scope of coordination, x_{coord} . From the Spearman's Rank correlation in Tables E.5 until E.8 it is found that there is no significant impact on any of the four performance indicators. Although a higher scope of coordination results in statistically significant differences in taxi time, taxi distance and density, the practical relevance is found to be only trivial. This confirms the findings of Udluft, who stated that the effect of communication only becomes visible for highly congested taxiway networks [72]. Also, when comparing the tested scenarios with the actual operation it is found that changing the scope of coordination only has a limited effect on the practical relevance, although it remains trivial.

Table 6.4: Values of operational parameters as tested by the sensitivity analysis

	Baseline	Option 1	Option 2	Option 3
<i>D_throughput</i> [aircraft/hour]	50	40 (-20%)	45 (-10%)	55 (+10%)
<i>No_turn</i> [°]	30	20 (-10°)	40 (+10°)	
<i>Min_speed_frac</i> [-]	0.10	0.05 (-50%)	0.15 (+50%)	
<i>x_{coor}</i> [agents]	2	0	1	3

6.3.3. Aircraft Dynamics

Visual inspection and analysis of the 14 days of simulated operation showed that the taxi speed of the aircraft agents deviated from the ones in the actual operation. The selected speed profile is one of the key elements of the model, as it directly impacts the taxi speed and thereby the taxi time and density on the taxiway layout. Additionally, it is the main input of determining a segment's weight, meaning it directly influences the assigned routes. Therefore, the aircraft dynamics, in terms of speed profile, have to be tested in detail.

The aircraft dynamics are characterised by 5 parameters:

- v_{\max} : maximum speed of an aircraft.
- v_{turn} : maximum turn speed of an aircraft.
- acc_{com} : comfort acceleration of an aircraft.
- dec_{com} : comfort deceleration of an aircraft, used for regular deceleration.
- dec_{\max} : maximum deceleration of an aircraft, used to prevent conflicts.

From the taxi speed box plot in Figure 6.3, it can be concluded that the speed profile in the simulation model might be set too optimistic as the mean value is 15% higher in the simulated operation. Based on some test scenarios with reductions in speed, it is found that reducing the speed by 20% allows for a better representation of the actual operation. First of all, the aircraft dynamics parameters are changed individually. The standard values and selected values for the sensitivity analysis are shown in Table 6.5. Additionally, different speed profiles are created and analysed, to explore which subset of the aircraft dynamics might have to be altered. An overview of the created speed profiles is presented in Table 6.6. *Profile1* and *Profile2* refer to speed profiles in which all parameters (speed, acceleration and deceleration) are decreased and increased by 20% respectively. *Profile3* and *Profile4* have similar speeds as in the baseline scenario, but have a decrease and increase of 20% in terms of acceleration and deceleration. Additionally, *Profile5* is a speed profile in which the speeds are increased by +20%, while the acceleration and deceleration are reduced by 20%. The opposite approach results in *Profile6*. Only a change in speed parameters is made to obtain *Profile7* and *Profile8*. These eight speed profiles are tested in addition to the scenarios in which the parameters are changed individually.

General findings

The results of the sensitivity analysis of the aircraft dynamics are shown in Table E.1 until E.4, for the performance indicators taxi time, taxi distance, average taxi speed and average density respectively. Additionally, Appendix G provides an example of hypotheses that could be tested using the data as presented in these tables. Looking at a global level at these results, it can be concluded that none of the tested scenarios has a practical relevance compared to the baseline scenario or the actual operation in terms of taxi distance, as is seen in Table E.2. This is also confirmed by the Spearman's Rank Correlation in Table E.6.

Focusing on the taxi time, taxi speed and density in comparison with the actual operation, it is found that an increase in any of the individual parameters results in a larger practical relevance, while this significance is lower for decreased values. This difference can be large, e.g. lowering v_{\max} or v_{turn} reduces the practical relevance from large to trivial. *Profile1* confirms the finding that reducing all the variables of the aircraft dynamics by 20%, results in a simulated operation much closer to the actual operation.

It should be noted that the sample size for *Profile6* on the 3rd and 7th of May, and $dec_{\text{com}} + 20\%$ on the 3rd of May is much smaller than the size of the original flight schedule (914 and 853 for the 3rd and 7th of May respectively). In these three simulations an operational gridlock was created, which blocked part of the taxiway network. More information regarding the causes of these gridlocks is given in Subsection 6.4.1. To make sure the effects of these gridlocks are not included in the sensitivity analysis, it is decided to only include flights that are spawned into the network up to 1 hour before the gridlock occurred.

Speed profiles

Analysing the different speed profiles, it is found that the effect on taxi time, average taxi speed and average

Table 6.5: Values of the aircraft dynamics as tested by sensitivity analysis

	Baseline	Option 1	Option 2
v_{\max} [kts]	30	24 (-20%)	36 (+20%)
v_{turn} [kts]	10	8 (-20%)	12 (+20%)
acc_{com} [kts/s]	0.5	0.4 (-20%)	0.6 (+20%)
dec_{com} [kts/s]	1.5	1.2 (-20%)	1.5 (+20%)
dec_{max} [kts/s]	10	8 (-20%)	12 (+20%)

density is statistically significant compared to the baseline scenario. Additionally, these results all have an effect size $r \geq 0.3$, meaning they are considered practically relevant as well.

Comparison with the actual operation reveals that both *Profile1* and *Profile5* are statistically significant in terms of the 4 performance indicators, but are only considered to be trivial in terms of practical relevance. Both speed profiles possess the same characteristic of a 20% reduction of in terms of v_{\max} and v_{turn} . Also *Profile7* is characterised by the reduction in speeds, but its effect on the average density is found to be statistically significant (Table F.4). For *Profile5*, the practical relevance in terms of taxi speed is found to be lowest of all tested scenarios, while *Profile7* results in the lowest value for both taxi time and average density. This is an indication that reducing both speeds is required to get closer to the actual operation.

Taxi speed

Regarding the individual parameters maximum speed v_{\max} and maximum turn speed v_{turn} , the Spearman's Rank Correlation shows that an increase in any of these parameters results in a decrease in taxi time and increase in average taxi speed. These effects, as well as the fact that the average density is lower, are statistically significant when compared with the baseline scenario. Due to the limited congestion on the taxiways, aircraft are able to travel at maximum speeds along large parts of their routes. The higher maximum speeds allow aircraft to travel faster, reach their destination sooner and thereby the density on the taxiways decreases. When comparing the scenarios with the actual operation, it is observed that a reduction of 20% in either one of the two speed variables results in a practical relevance that is only trivial for taxi time, taxi speed and density.

Acceleration

An increase in comfort acceleration acc_{com} results in higher average speeds as confirmed by Table F.7, as aircraft reach their maximum speed quicker after slow downs caused by making a turn or congestion. The higher average speeds are linked with the shorter taxi times and lower aircraft density, which are both significant compared to the baseline scenario. The change in value of acceleration resulted in a significant difference in terms of taxi time, average taxi speed and average density compared to the baseline scenario, both statistically and practically. Comparing the different scenarios with the actual operation, it is found that the simulated operation gets closer to the actual operation for lower values of acc_{com} in terms of practical relevance for each of the 4 performance indicators.

Deceleration

For the deceleration dec_{com} it is concluded that it only has a statistically significant effect on average speed when compared to the baseline scenario, which increases for higher values of dec_{com} . As the deceleration distance decreases, aircraft agents can travel larger distances at a higher speed. Although the sensitivity results in Table E.1 state that changes in dec_{com} cause differences in taxi time that are significant, no significant correlation is found in Table E.5. The maximum deceleration dec_{max} is mainly used to react quickly to (potential) conflicts that arise, as well as for the required number of short segments that has to be reserved. No practical relevance is found in Table F.1 until F.4 and no significant correlations are found in Table E.5 until F.8. Although all performance indicators of each scenario are statistically significant when compared to the actual operation, the effect size (slightly) decreases with decreasing values of dec_{com} and dec_{max} .

6.4. Discussion of Results

This section discusses the knowledge that has been gained from the development process and the assessment of the emerging behaviour of decentralised control with respect to the actual operation. Subsection 6.4.1 covers the operational gridlocks that occurred in three simulations of the sensitivity analysis.. Subsection 6.4.2 describes the general findings and goes into more detail regarding the emergent behaviour in terms of routing strategies of the agents. An overview of the model limitations is given in Subsection 6.4.3.

Table 6.6: Speed profiles for sensitivity analysis

	v_{\max} [kts]	v_{turn} [kts]	acc_{com} [kts/s]	dec_{com} [kts/s]	dec_{\max} [kts/s]
Baseline	30	10	0.5	1.5	10
Profile1	24 (-20%)	8 (-20%)	0.4 (-20%)	1.2 (-20%)	8 (-20%)
Profile2	36 (+20%)	12 (+20%)	0.6 (+20%)	1.8 (+20%)	12 (+20%)
Profile3	30 (-)	10 (-)	0.4 (-20%)	1.2 (-20%)	8 (-20%)
Profile4	30 (-)	10 (-)	0.6 (+20%)	1.8 (+20%)	12 (+20%)
Profile5	24 (-20%)	8 (-20%)	0.6 (+20%)	1.8 (+20%)	12 (+20%)
Profile6	36 (+20%)	12 (+20%)	0.4 (-20%)	1.2 (-20%)	8 (-20%)
Profile7	24 (-20%)	8 (-20%)	0.5 (-)	1.5 (-)	10 (-)
Profile8	36 (+20%)	12 (+20%)	0.5 (-)	1.5 (-)	10 (-)

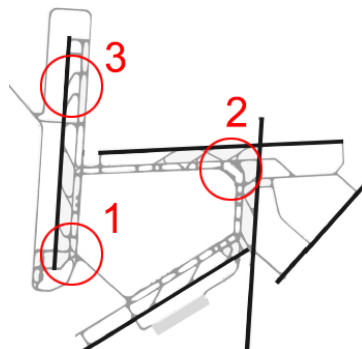


Figure 6.18: Location of gridlocks

6.4.1. Operational Gridlock

Unfortunately, an operational gridlock occurred in 3 of the total 56 simulated ground operations that have been run for the sensitivity analysis, meaning that the simulation model was not able to solve a conflict that occurred. The gridlocks occurred on three different locations in the airport layout, as can be seen in Figure 6.18, and were caused by two different aspects. This subsection gives more insights regarding the causes of the gridlocks, as well as recommendations of implementations to prevent future gridlocks from occurring.

Shortage of knowledge

The gridlocks that occurred for the scenario *Profile6* on the 3rd and 7th of May is the consequence of a shortage of knowledge of an ATC agent regarding the position of aircraft agents that are under control of other ATC agents. The gridlocks occurred at location 1 and 2 in Figure 6.18. Figure 6.19 provides insights of how such a gridlock is created. It should be noted that the scenario is drawn in such a way that it clearly visualises the issue and thereby it is not according to scale.

Figure 6.19 sketches a scenario in which node *A* is in its decision making process. Three aircraft are present; aircraft agent *AC1* has point 1 as destination (runway 18C – 36C), while *AC2* and *AC3* want to travel towards runway 18R – 36L via node 2. *AC1* and *AC2* are both heading towards ATC agent *A*. *AC3* is halted in front of node *D*, to maintain a safe separation with aircraft *AC2*. ATC agent *A* checks whether an aircraft hand off is possible, starting with *AC1* as it is waiting longest. One of the conditions is to check whether all segments and nodes within separation distance are available. For the scenario outlined, segments *AB* and *BC*, and node *B* has to be reserved. However, a few moments later when aircraft *AC1* has moved to the location as presented in Figure 6.19(b), it cannot receive a hand off from node *B*, as node *B* cannot handle *AC1* safely as long as *AC3* is too close to node *C*. The same holds for *AC2* being too close to node *D*, as well as *AC1* being too close to node *A*. This latter means that *AC2* is not able to pass node *A* meaning that these three aircraft are stuck, while waiting for each other to move.

The conflict could have been prevented or solved in different ways. The easiest one would have been to let aircraft *AC1* go straight and use a different runway entry to line up, instead of going via node *C* towards entry 1. Currently the model is designed in such a way that it uses the exact same origin and destination as used in the actual operation, in order to have a fair comparison. ATC should be free in selecting the best suitable option, to make sure the route destination can be reached, even if one of the runway entries is blocked. In

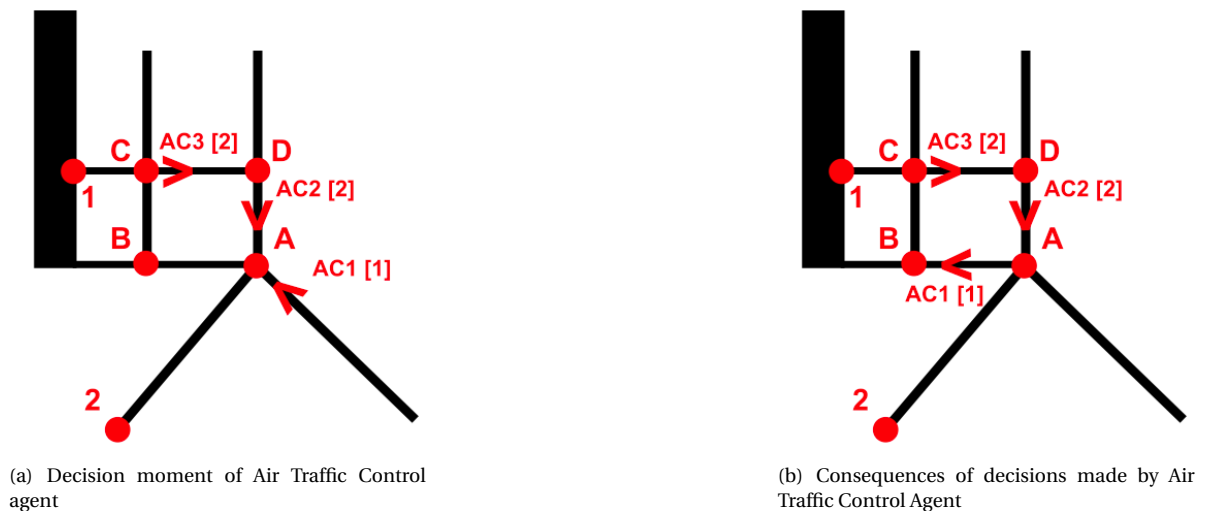


Figure 6.19: Operational gridlock caused by insufficient information of aircraft positions

future stages of the model development, conditions can be chosen like aircraft weight category or aircraft type to get closer to the actual operation.

Another option would be to allow more communication between ATC agents regarding the position of aircraft. If node *A* would have known that handing off aircraft *AC1* would push him into a congested area which required him to halt within short distance, the ATC agent would have handed off *AC2* first.

To determine which aircraft agent should be handed off, ATC agents go through a list of aircraft while checking if all handing off conditions are met. This list is currently being sequenced based on the amount of time an aircraft is waiting to receive a hand off. Alternatively, a factor could be included that takes into account the likeliness it can travel freely or away from congestion. For the scenario as shown in Figure 6.19, it is most beneficial if *AC2* is being handed off first as it can travel freely after it and (part of) the congestion is resolved, while handing off *AC1* first will amplify the congestion.

Invalid route

A second type of gridlock occurred while simulating the scenario *Dec_com_+20%* for the 3rd of May, at location 3 in Figure 6.18. The gridlock is shown in Figure 6.20. The thick line represents runway 18C – 36C, which is used for arrivals in northern direction. All aircraft agents apart from *AC1* are moving towards node 2, in order to travel around runway 18C – 36C towards runway 18R – 36L. To safely spawn *AC1* into the network, the availability of the runway exit has to be guaranteed. As mentioned in Subsection 5.4.2, it is required that all nodes within the reservation distance d_{sep} to be reserved. The initial route of *AC1* is towards node 1 and includes ATC agents *A* – *C* – *D*. ATC agent *A* cannot find a different route as both segment *CD* and *BE* are unavailable due to traffic in the opposite direction. Therefore, node *A* could be reserved in the scenario at stake, as well as segment *AC* due to the following reason: although aircraft agent *AC2* is moving along segment *DC*, ATC agent *C* still believes it has two outgoing edges: one towards the runway and one towards node *A*. Consequently, it is not removing one of the incoming edges to secure a forced outpath. As a consequence, reserving node *A* is possible. ATC agent *A* therefore guides *AC1* towards node *C* as it believes node *C* can handle the aircraft. The effect of this decision is shown in Subfigure 6.20(b), where both *AC1* and *AC2* are heading towards node *C* and the only outgoing segment is towards the runway which is not a destination of any of the two aircraft. As the two opposing aircraft are not able to escape from the conflict, a gridlock occurred.

Also this conflict could have been prevented from happening in two different ways. One option would be to put a condition on the fact that a segment could not be reserved in case it is lost. Looking at the situation in Subfigure 6.20(a), segment *AC* – *CA* is won by node *C* as the bid value coming from *AC2* is much higher than *AC1* as it has been taxiing for a much longer time. In the normal case, the nodes and segments that require a reservation depend on the latest route update of an aircraft, which is based on the segments that are lost or won by the upcoming ATC in the route. However, as both segments *AC* and *BE* are lost, no new path can be determined, meaning that its initial route via *A* – *C* – *D* will remain. Not being able to reserve node *A* and



Figure 6.20: Operational gridlock caused by missing forced outpath

segment AC should result in the situation that $AC1$ is not spawned yet until a safe operation is possible.

Another simple option might be to not take segments towards runways and aprons into account when analysing the number of outgoing links an intersection node has. In this specific scenario this would mean that the moment aircraft $AC2$ enters segment DC , segment CA is labelled as forced outpath as it is the only outgoing edge of node C . As a consequence, segment AC is removed from the graph to prevent aircraft opposing each other and thereby creating a gridlock. One could make it more complex by putting conditions on links. For example, when $AC2$ is entering the runway via node 3, $AC1$ can safely enter link AC . Or if $AC1$ wants to reach the runway at node 3, also no issue is created. This condition can be summarised in the statement: "an aircraft is allowed to enter segment x towards segment's destination y , if at least one outgoing edge towards an intersection agent can be created, based on the current outgoing edges available". So only if aircraft $AC2$ is travelling towards node 3 an outgoing edge can be (re)created for node C , namely CD . In all other scenarios, aircraft $AC1$ is not allowed to enter segment AC . For further development, the segment conditions can be further expanded. As an example, aircraft weight categories or aircraft types can be included as a condition whether particular aircraft are allowed to enter a specific segment.

6.4.2. Overall Findings

From Section 6.2 it is concluded that the simulation model has been able to successfully simulate the ground operations for a test setup consisting of 14 consecutive days. Each of the 6 runways of AAS has been used at least once in both arriving and departing mode to handle some of the almost 12,000 flights that have been spawned into the network. Such a large set of operational variety has made it possible to test the model for a large variety of different scenarios, different conditions, and complexities.

Overall performance

Comparing the operation as managed by decentralised control with respect to the actual operation, it is found that the performance in terms of taxi time and distance shows very similar patterns. The simulation model has been able to guide a few more aircraft in time towards their destination, in a taxi time that was lower on average. This difference can mainly be explained by the fact that some aircraft in the actual operation had to wait at a parking bay until their gate became available. As this research assumed that the gate and apron aspect are not taken into account, all arriving aircraft can travel directly to their apron in the simulation. Additionally, the performed sensitivity analysis revealed that part of this difference can be explained by the aircraft's speed that is set too high.

Taxi routes

It is found that the average taxi distance travelled in the simulation differs less than 3% from the average distance in the actual operation. Additionally, from the sensitivity analysis it is concluded that a small change in any of the selected parameters had no significant effect on the taxi distance travelled. Still, the individual

paths travelled could differ quite a lot. Analysing the operations in terms of routing strategies, it is noted that the routes travelled in the simulated operation are more complex in terms of entropy measures due to the fact that routing patterns are less evident than in the actual operation where ATC adheres to predefined procedures. As ATC agents only have local information and a limited number of aircraft they have to control safely and efficiently towards their destination, these agents have much more flexibility in which route to select. The entropy measure confirms this aspect, from which it can be concluded that distributed control is able to handle a more complex and chaotic operation.

Initial value problem

A few remarkable routing aspects in terms of emerged behaviour have been found, which caused differences in routing strategies. First of all, the routing strategies applied for a particular runway configuration is heavily influenced by the route the first aircraft agent travels, the so-called initial value problem. While this first aircraft is travelling the taxiway network, all ATC agents observe that the aircraft's current segment is used in a particular direction. As a result, ATC agents can temporarily not include this segment in a route update for an other aircraft that is willing to travel in the opposing direction. Consequently, the ATC agent has to find an alternative path to get the aircraft to the other side of the airport. This means two streams of traffic are created, often used by arriving and departing aircraft respectively, which depend on the initial aircraft's route.

Lack of timewise information

From observing the simulated operation it is clearly visible that vital timewise information is missing, meaning that the simulation model is showing reactive instead of proactive behaviour. An apparent example is the scenario where runway 18C – 36C will become available to cross, once the last aircraft has landed/taken off. In the actual operation a small queue is lining up to cross the runway the moment it is deactivated, while the aircraft agents in the simulation are already travelling around 18C – 36C and thereby taxi a much longer distance. Additionally, if an aircraft is waiting on taxiway Quebec (see Figure 2.2) due to congestion, ATC agents have no idea when the taxiway will become available or when the stationary aircraft will start moving. Consequently, other aircraft are being guided around the airport as it seems most optimal for that particular situation. However, it occurs often that the stationary aircraft is already moving before the other aircraft would even have come close if it was guided along taxiway Q. This lack of knowledge results in deficiencies in terms of taxi distance and taxi time as far from optimal routes are selected based on an ATC agent's insights in the current operation at stake.

It is recommend to allow more timely information to be shared. This includes future information regarding the future (expected) traffic state. One example is to make the weight calculation of taxiway segments more dynamic. In case a particular segment is included in many routes, it should receive an increased weights to encourage aircraft to taxi via alternative routes. Thereby, the traffic load is already distributed across the airport layout, the prevent future bottlenecks from occurring. Also, expected directions of taxiway use could be included, in order to let ATC agents proactively adapt to upcoming changes.

Local goal

Also, the local goal of the agents is dominating the global goal of the operation. A clear example is the runway queuing strategy that is missing in the simulated operation. Each ATC agent aims at guiding the aircraft agent under control as soon as possible towards their destination. Close to a runway this occasionally results in parallel taxiways being blocked by departing aircraft. The created congestion and fact that streams of arrival aircraft agents have difficulties to pass this congestion reduces the operational efficiency. The first operational gridlock, covered in Subsection 6.4.1 and visualised in Figure 6.19, is also an example of how a lack of information (in terms of aircraft positions) means that ATC decisions are made, based on local level instead of global level.

One recommendation would be to add a second layer of control, e.g. ATC agents that are responsible for a small subset of local agents. This "central" layer acts as a helicopter view, which could identify (potential) bottlenecks, or prioritise particular local agents for a better conflict solving.

Also, it could be an option to identify the taxi highways, depending on the runway configuration in use. The weights of these segments should represent a scenario in which aircraft are penalised for standstill, but promoted in case an aircraft could follow the highway for a longer period of time.

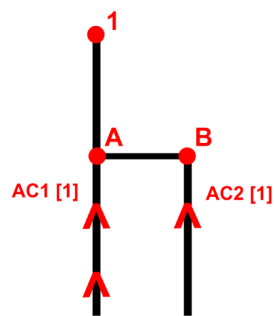


Figure 6.21: Adding extra layer of auctions

6.4.3. Model Limitations and

Although the model has been able to successfully simulate the ground operations for 14 consecutive days, some model limitations came forward while analysing the model's performance with respect to the actual operation. This subsection covers the main operational limitations of the model that are found.

Runway queue

Aircraft agents do not create a runway queue. For runways close to the main taxiways, this causes a large congestion that blocks the surrounding taxiways. As both arriving aircraft agents, and aircraft with a different runway as destination have difficulties travelling across this congestion, operational deficiencies are created.

To improve this aspect, the model requires a strategy to create a runway queue at specific locations in the taxiway system. This also requires an adapted routing algorithm that takes into account whether aircraft should be guided towards the runway entry, or towards the end of the runway queue.

Auction to use segment next

The communication between ATC agents is currently based on the interaction to guarantee a safe operation, as well as the communication required to perform the auction process. This latter aspect focuses on determining the *optimal* direction of a segment. However, a different kind of auction might result in a fairer usage of the segments. An example is shown in Figure 6.21. Both aircraft *AC1* and *AC2* want to travel towards node 1 via node *A*. The only auction that currently takes place, is between *A* and *B*, and *A* and 1 to determine the direction of usage. The current implementation requires the ATC agents to make their decision in numerical order. As a result, ATC agent *A* can take its decision first, meaning that if segment *A* – 1 becomes available, node *A* has the first option to claim it. So if more aircraft are trailing *AC1*, this means that all of them have priority above the aircraft waiting at node *B*. An option to improve this aspect would be to introduce an auction between agents *A* and *B* on who is allowed to use the segment *A* – 1 next. This means that the operation is improved based on the fact that aircraft with higher value get higher priority.

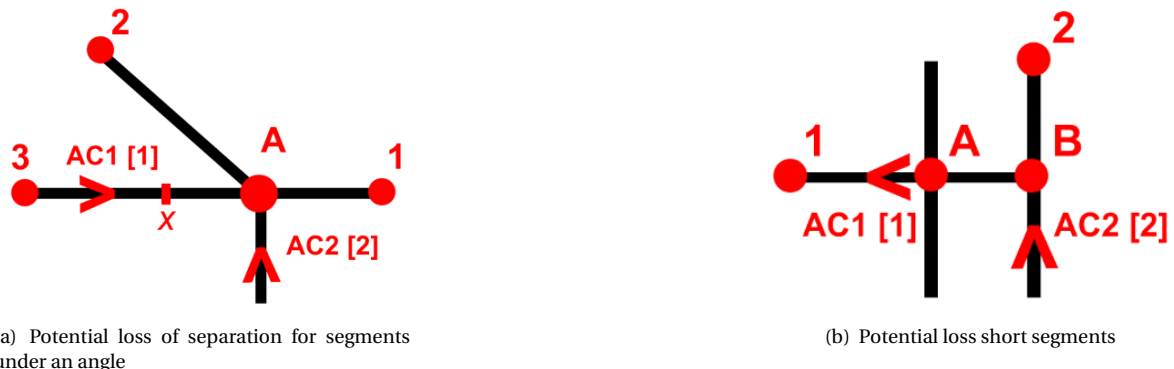
This aspect is also related to the gridlock scenario as presented in Subsection 6.4.1. The implemented ATC agents make a decision based on the observations of the neighbourhood as well as the aircraft under control. However, this means that decisions are based on obtaining the local instead of global goal. Consequently, it occurs regularly that the complexity in terms of congestion becomes worse, or is solved less efficiently.

Forced outpath

As explained in Subsection 6.4.1, a second type of conflict is caused by ATC agents that do not force an outpath. Each ATC agent makes sure it has at least one outgoing segment. However, it could occur that the only remaining outgoing edge is towards an apron or runway, which is no destination of all incoming aircraft agents. Consequently, the conflict at stake cannot be solved. A potential solution is given in Subsection 6.4.1 and refers to implementing a more detailed set of conditions on which aircraft is allowed to enter a particular segment. Additionally, the segments towards a destination can be excluded when analysing the number of outgoing segments.

Closest point of approach

ATC agents are mainly responsible for facilitating an aircraft agent from origin to destination, providing a safe operation. Apart from ATC, aircraft agents are also accountable for a safe operation, as they have to make sure that they do not collide with aircraft on the same link or the next link. Aircraft agents have to adapt their



(a) Potential loss of separation for segments under an angle

(b) Potential loss short segments

Figure 6.22: Closest point of approach

speed when two aircraft agents are heading towards the same ATC agent from different segments. It is already stated in Section 5.1 that for this model it is assumed that the distance between two aircraft agents equals the taxiway distance. It is one of the reasons why one of the layout simplifications involves the straightening of corners, as mentioned in Section 4.2. However, not all corners could be simplified to have an angle of 90° without drastically changing the airport layout. Below, two different scenarios describe the necessity of implementing the concept of closest point of approach in future model developments, in order to guarantee a safe operation.

An example of a scenario involving long segments (\geq separation distance) is shown in Subfigure 6.22(a). Both aircraft agents AC1 and AC2 are heading towards the same ATC agent A, although both on their way towards a different node (1 and 2 respectively). As AC2 is closest to node A, it gets the first shot at receiving a hand off. In the current model, AC2 will receive its hand off while AC1 waits at location x , separation distance away from node A, until AC2 is at least a separation distance away from node A. However, for all cases where the angle between segments 3 – A and 2 – A is less than 90° , the distance between the two aircraft agents will be less than the set separation distance. Especially for sharp corners, the separation between the two aircraft will be small. In a preferred operation and before handing off AC2, ATC agent A will check whether AC1 can stop in time such that AC2 can safely travel along segment A – 2 taking the closest point of approach into account. If it is possible, AC1 receives a message at what distance from node A it has to stop. If not, AC1 gets priority and will be handed off first.

Although a large fraction of the model development has focused on guiding aircraft agents across short segments, as explained in Subsection 5.4.2, the concept of closest point of approach is not integrated yet. An example given in Subfigure 6.22(b) shows how aircraft agent AC2 receives a hand off from ATC agent B towards segment B – 2, as it cannot discover any other aircraft agent on its outgoing segments. Even when AC1 is within separation distance from node B, it is not labelled unsafe as it is not on an outgoing segment of node B. Consequently, the closest point of approach between AC1 and AC2 could be within the separation distance. A detailed analysis is required to implement the concept of closest point of approach, in order to exclude this "unsafe" aspect. Reason for this is the fact that e.g. for the AAS layout, the two main parallel taxiways are separated by a distance that is less than the set separation distance of 150 metres. Based on the concept of closest point of approach, two aircraft will conflict if they are travelling in opposite direction along the two parallel taxiways causing a complete gridlock in the operation. It is therefore of importance to distinguish longitudinal separation from perpendicular separation.

Static separation distance

Continuing on this latter aspect, the separation distance plays an important factor in the simulated operation. To have a safe operation, both ATC agents and aircraft agents try to maintain a minimum separation distance between two aircraft agents. This separation distance is set to 150 metres, and it is assumed that the distance between two aircraft agents equals the taxiway distance between them. Additionally, this minimum separation applies to all aircraft agents, no matter whether travelling at full speed or waiting in a (runway) queue. Especially the latter case results in a large deficiency in comparison with the actual operation where aircraft maintain a much smaller separation distance at low (or zero) speed.

An example is shown in Figure 6.23 where the simulated and actual operation are visible in Subfigure 6.23(a) and 6.23(b) respectively. In the simulated operation, a queue is lining up on taxiway Quebec (lower right corner) with aircraft being separated by 150 metres. The first aircraft in the queue is not allowed to be



Figure 6.23: Separation between slow moving vehicles

handed off by its ATC agents, as the distance between this agent and the aircraft waiting at the runway is less than the required separation distance. The most common aircraft at AAS is the Airbus A320, having a total length of 38 metres [1, 27]. This means two of these aircraft agents maintain a separation of at least 110 metres, even when they are waiting in a queue. In the actual operation, the aircraft line up much closer.

The larger separation minimum maintained by the simulation model for slow moving or stationary aircraft has one large consequence. As the stationary aircraft occupy a larger area of the airport layout, more aircraft are affected by this congestion. As an example, in case the second aircraft in the queue on taxiway Quebec in Subfigure 6.23(a) wants to move in southern direction instead of towards the runway, it has to wait until the aircraft in front has moved. If a smaller separation minimum is applied for slow moving aircraft, the leading aircraft on taxiway Quebec could move forward and thereby give room to the second aircraft to continue its route. Summarised, a function has to be developed that adapts the required separation distance to an aircraft's speed. Once again, it is of importance to distinguish longitudinal separation from perpendicular separation

Unrealistic estimation of taxi times

Whenever the runway 18C – 36C is activated in either arrival or departure mode, at least one of the stop bar agents has to remove a segment on which aircraft agents can no longer travel as they will interfere with traffic that is using the runway. Every 10 seconds, the runway agent tries to make an estimation of the traffic willing to use the runway within a time horizon of 10 minutes. The source agent of each runway has an arriving schedule, such that it knows exactly when the runway is used for arrivals. Although a runway knows if and which departing aircraft agent is travelling towards it, it has no departure schedule with timings when the aircraft agent reaches the runway. Instead, the runway agent communicates with the ATC agent that is controlling the departing aircraft at that time, asking for a time estimation based on the assigned route and current speeds at the different segments. However, as future traffic scenarios and conflicts are not taken into account, this time prediction is often too optimistic. As a consequence, stop bar agents are closing segments too early which results in aircraft agents having to travel longer routes.

For further research, it is recommended to implement a new protocol to set a segment's weight, depending on the expected state of the network in the future.

6.5. Areas of Improvements

From the development phase itself, as well as from comparisons between the simulated and actual operation, a few model limitations are identified in Subsection 6.4.3. The main areas of improvement are listed in this section.

Future information

There is a lack of future information, meaning that the ATC agents show reactive instead of proactive behaviour. An example is when an aircraft has to make a detour, because the ATC agent does not know that a shorter route will be available soon. Also, the future taxiway usage in terms of direction and time are not known, causing some large operational deficiencies.

The only type of future information that is implemented relates to the increasing weight of segments in advance of a runway becoming active. As these segments can no longer be used for the upcoming runway mode of operation, ATC agents make sure that no more aircraft are guided along these routes. However, more

future information is required to improve the operational efficiency. One option could be to allow a reservation system on the most restrictive or important taxiways, of which single lane taxiway Quebec is an example. A reservation system would give all ATC agents insights in the time duration and direction in which the taxiway is being used. Based on this information, an agent can decide whether it wants to use this taxiway or whether a more suitable option exists. This might lead to a better use of the limited resources and prevent aircraft from making detours in case the segment is suddenly used in the opposite direction.

Additionally, it could be the case that based on the current state of the taxiway system, all ATC agents decide to guide their aircraft along the same route. Currently, there is no feedback to the ATC agents that warns them of expected congestion. Future work should focus on the implementation of anticipatory routing, to improve route stability and prevent future congestion from occurring. One way to implement it is by making sure that the weights of segment anticipate to future situations. In case the route of many aircraft is planned along the same resources, the weight could be increased to encourage ATC agents to look for alternative options.

Global goal

It has been noticed that the individual goal of an agent surpasses the overall goal, which has led to operational deficiencies or even gridlocks. A new type of agent could be introduced, which is responsible for controlling a group of local controllers. By adding centralisation on a small scale, conflicts might be solved more easily, and prevented from occurring. A trade off should be performed to determine the required level of centralisation.

It is expected that adding a small form of centralisation helps to solve two limitations that are mentioned earlier. An agent that possesses some central responsibilities, is able to oversee the communication of the local agents under its control, it could encourage the creation of neat runway queues. Additionally, this agent could solve the first gridlock, as it might warn the local agents for upcoming congestion. Thereby, the local controllers can be supported in their decision making process, on which aircraft to hand off first.

Safe operation

the simulation model applies separation based on taxiway distance instead of closest point of approach. To increase the safety level of the operation, it is required to implement the closest point of approach protocol, in which a distinction is made between longitudinal separation and perpendicular separation.

Also, the speed aspect should be included, to make the separation distance dependent on the relative speed between aircraft. Aircraft waiting in a runway queue should be allowed to get much closer compared to two trailing aircraft taxiing at full speed.

One of the created gridlocks is caused by inadequate application of the forced outpaths. Normally, an outpath would be forced in case an ATC agent has only one outgoing segment remaining, either due to inbound traffic or reserved segments. However, segments towards an apron or runway are not excluded from the number of available segments. This could lead to the situation that the only available outgoing segment is towards a runway, which is not the destination of any of the incoming aircraft.

Although it would be possible to exclude segments towards a destination, when counting the number of outgoing segments, this is not preferred as it leads to many unnecessary forced outpaths. Instead, it is recommended to place conditions on the hand off process: there should be communication between agents, to determine whether an ATC agent could still process all incoming aircraft in case the new aircraft receives its hand off. Only if this new condition is satisfied, the aircraft receives its hand off.

7

Conclusions and Recommendations

Airports in Europe have difficulties to meet the rising demand in air transportation. The average growth of 3.7% per year results in operational bottlenecks at airports. As a result, airport capacity problems are responsible for 14% of the delays in Europe [17, 30]. The high costs that expanding infrastructure brings along, is one of the main reasons why alternative ways are explored to improve the operation within the available capacity. Over the past few years, multiple studies have focused on the human aspect, which is Air Traffic Control (ATC). A variety of tools has been developed with the aim to support ATC in their decision process and thereby ease a controller's workload [4, 66]. However, the operation remains dependent on a centralised controller, who is responsible for many aircraft and tasks in a large area of the airport. Due to the constrained processing ability of humans, only limited local information can be absorbed and used for future decisions, while still maintaining a safe operation. To structure taxi flows, ATC makes use of procedures, protocols and standard taxi directions for each runway. ATC can deviate from these standards if the situation requires it.

Runway reconfigurations disturb the operation as multiple new traffic streams have to be integrated in the existing traffic structure. As these traffic flows are potentially conflicting with the already existing traffic, ATC has difficulties to use the available runway capacity effectively. As the operational standards limit ATC's option space, ATC does not have the means to solve the disturbance quickly. Consequently, the effects of the reconfiguration remain noticeable for a longer period of time, causing operational delays to increase.

Decentralised control shifts decision making from a global level to a local level by placing fictitious air traffic controllers on taxiway intersections. Each of the controller agents is responsible for guiding an aircraft effectively and safely towards its destination. Instead of including limited global information to make a decision that seems best on system level, detailed information from the surroundings is used to guide an aircraft. Since local controllers can focus on solving local congestion and conflicts, they can adapt quicker and more efficiently to changing conditions. Consequently, it is expected that the dynamics, uncertainties and disturbances of an airport's ground operation, are better managed when using decentralised control.

So far limited research has focused on this promising type of control. Most studies related to the field of ground operations stick with simplified airport layouts, making validation impossible. Consequently, the results cannot be translated to practice.

This research has taken the unconventional path by deciding to implement decentralised control from the operational needs, instead of the theoretical view most researchers use. Countless hours of observing the actual ground operation, and interviews with (former) air traffic controllers preceded the development phase. The direction and requirements of the simulation model have been defined along the development phase, by iteratively comparing the simulation results with the actual operation. This approach allowed the researcher to acquire practical knowledge, and a better understanding of the principles and mechanisms, to apply agent-based modelling and decentralised control to real-life cases.

Section 7.1 describes the contribution of this research, followed by the main findings in Section 7.2. Recommendations for future research are provided in Section 7.3.

7.1. Contribution

This research continues on work done by Udluft [73]. His research has put a lot of effort in developing an agent-based modelling toolbox, called Open Source Simulator for ATM Research (OSSAR), that could be used to simulate ground operations. This toolbox has been the starting point of this research. Contribution to this model, as well as to science, is covered in this section.

Demonstrating feasibility

This research implemented a decentralised control infrastructure for an airport's ground operation, which is able to successfully guide aircraft agents towards their destination. Udluft accomplished this same step for a simple traffic set on a basic airport layout. This research took the level of decentralisation to a new level, by enabling decentralised control on an actual layout consisting of 218 nodes and 266 segments. Apart from additional features to the model, new types of agents had to be introduced and developed to cope with the challenging aspects of an actual airport infrastructure. This research also demonstrates the feasibility of applying agent-based modelling techniques to a full scale and complex operation.

Research approach

Additionally, this approach is a contribution to science on how to draw practically useful conclusions. Most studies in the air transportation domain cannot be applied in practice, as they do not understand the airport dynamics, oversimplify the operation, or aim to find the theoretical optimum.

Agent-based modelling is known for its ability to model socio-technical processes. However, the performance of this technique heavily relies on the level of operational knowledge, which is why it is of uttermost importance to acquire a detailed understanding of the ground operation. This research is a one of a kind in terms of the research approach that has been followed. Instead of focusing on following a predefined methodology that is based on fixed requirements, this research has been guided by assessing the simulation results with the actual operation. Countless hours have been dedicated to observing the actual ground operation, interviewing (former) air traffic controllers, and comparing simulated and actual operation. All these steps led to a detailed understanding of the complexities and dynamics involved in an actual ground operation.

Agent-based simulation model

As this study is part of a research cluster focusing on resilience, one of the requirements was to keep a toolbox mindset. Consequently, it was decided to design a modular model, for which building blocks can easily be added or removed to suit the researcher's needs. A well documented and extended version of the OSSAR simulator has been one of the main deliveries of this research. When starting this research, only a few guidelines were added as comments to OSSAR's hundreds of lines of code. To acquire a clear understanding of all work done before, as well as for the future of this toolbox, this research has added detailed information to every line of code. This step revealed a number of small bugs that may not have come forward based on observing the simulation itself. Additionally, new agent types as well as many new features have been implemented. All of these implementations resulted in a ground operation toolbox, of which all tools should be in place to easily change airport layout or experiment with new types of coordination.

Industry

Follow-the-greens is an innovative system to guide aircraft across an airport layout, by means of lights that are installed in the taxiways [28]. The developed agent-based simulation model enables the opportunity to test different forms of autonomous behaviour. Additionally, the understandings and insights acquired in this research could be of help for further developments of the follow-the-greens concept. Overall, the combination of follow-the-greens and decentralised control enables a strong system to autonomously manage future ground operations.

7.2. Main Findings

This section covers the main findings of this research, which aims at achieving the following research objective:

"the objective is to create an understanding of the principles and mechanisms of a decentralised air traffic control, by systematically comparing the emergent behaviour of an agent-based model to the actual ground operation"

To meet the research objective, two research questions have been established:

1. *How could decentralised air traffic control be implemented in an agent-based model to simulate actual ground operations?*
2. *Up to what extent does the emerging behaviour of decentralised air traffic control match the behaviour of centralised air traffic control at Amsterdam Airport Schiphol (AAS)?*

Decentralised control is successfully implemented by means of seven types of agents: three types of intersection agents, a runway agent, an apron agent and two agents responsible for adding and removing aircraft to the taxiway layout. The main requirement of an decentralised control agent is the ability to communicate with all agents within a specified radius in order to safely hand off an aircraft. This research emphasises the need to have basic insights in the overall state of the taxiway network, as well the future runway usage.

The developed agent-based model has been tested for a large variety of operational challenges to answer the second question. Comparisons between the simulated and actual operation revealed that the performance of decentralised control has similar patterns in term of taxi time and taxi distance. Differences in routing strategy are found, as decentralised control is using the taxiway layout in a more flexible manner. This is supported by the results, which show that routing patterns are less visible and less structured in the operation managed by decentralised control. The behaviour of the actual ATC to adapt their routing strategy proactively to upcoming operational changes, is missing in the simulated operation due to the lack of timely information. The main findings are explained below.

Scope of information

It is found that **only limited information is required to allow safe operations**, which confirms findings by Udluft [73]. Although his simple layout needed information from only two consecutive segments, he mentions that this number is likely to increase with airport complexity. However, this research found that there is no fixed scope of information, as this number differs for each ATC agent. One condition is that an ATC agent has to be able to determine whether all segments within a specified distance can be reserved, to safely hand off an aircraft. As each taxiway segment differs in length, the scope of information is linked to distance instead of a fixed number of agents.

However, **only local information is not sufficient to accomplish a good performance** in terms of decentralised control. The main taxiway layout of AAS consists of a main circle, characterised by the presence of single lane taxiways, of which the direction of usage has a direct impact on the routing structures. The taxi direction along this segment, as well as congested areas have to be known by an ATC agent, to make a decision whether the aircraft should take the northern or southern route to reach the other side of the airport. It is therefore decided that each ATC agent has access to the current state of the taxiway network. This statement conflicts with Udluft, who stated that limited information is sufficient to obtain good results when using decentralised control [73].

Additionally, this research demonstrates the **need for information regarding the future state of the taxiway network**, as large operational deficiencies are caused by ignorance. Examples are single lane taxiways that are “suddenly” being used in a particular direction, and runways that are being activated and thereby require taxiway segments to temporarily shut down.

Operational complexity

Fourteen days of simulated ground operations have been compared with respect to the actual operation, to draw conclusions based on the applied routing strategies. Entropy measures show that decentralised control is able to **successfully handle a more complex and chaotic operation**. As decentralised control does not have to adhere to preferred driving directions, it has a larger option space to avoid stationary traffic and absorb operational disturbances as caused by runway reconfiguration.

It is to be noted that this conclusion is mainly applicable to arrival peaks. Most of AAS’s runways require departure queues to be formed on the main taxiways. As the model is less efficient in creating these queues, parallel taxiways could be blocked by departing aircraft. Consequently, arriving traffic has issues to circumvent this kind of congestion.

Scope of coordination

The sensitivity analysis shows that varying the scope of coordination between 0 and 3 agents has no practical relevance in terms of taxi time, distance or speed. This confirms the findings of Udluft, who stated that **the**

effect of coordination only becomes visible for highly congested taxiway networks. The actual flight schedule results in a relatively quiet operation, allowing the formation of traffic flows. Since the planes encounter few other aircraft, coordination does not result in more efficient routes.

7.3. Recommendations

This research started with a model that demonstrated the feasibility of decentralised control, using a simple layout and departing traffic only. Months of observations, comparisons, and development resulted in a model that is able to simulate fourteen days of traffic on the layout of AAS. Although a model is in place that is a realistic representation of the actual operation, more research is required to use it in an operational context.

Model improvements

Based on the comparisons between the simulated and actual operation, a few model shortcomings and suggested improvements have been identified in Section 6.4.3 and 6.5 respectively. By enhancing the model in terms of anticipatory routing, prioritisation of the global goal and the maintaining of safety, a widely applicable model is created.

Coordination techniques

The importance of communication and information sharing has been noted multiple times by this research. It would be very interesting to use this existing model for testing different coordination technique to determine the impact of different protocols. Additionally, different types of auctions can be investigated. Currently, the only type of auction taking place is to determine the future direction of a taxiway segment. However, it would be interesting to determine whether a fairer operation is created if an auction takes place to determine which aircraft is allowed to use a segment next, in case two aircraft want to use the same segment.

Learning element

Another interesting technique would be to add a learning element to the ATC agents. If an evaluation takes place on the routing decisions made by the agents, they gain knowledge of good and bad decisions. By doing so, an understanding of the effect of decisions is created, as well as the traffic scenario that emerged from it. By encouraging good behaviour of agents, insights are gained of which decisions improve the operation. Eventually this could benefit the routing stability as applied by all agents, and the information of successful routing strategies could be used to improve the actual operation.

Apron operation

One of the assumptions of this research states that the apron operation is not included. When comparing the actual and simulated operation, it has been found that this assumption is noticeable. In the actual operation, aircraft had to wait on a buffer location as their gates were still occupied. Additionally, some departing aircraft receive a push back on one of the main taxiways, which temporarily blocks it and thereby creates congestion. Enabling apron operations also allows the integration of towing operations. Each of these examples are of interest to evaluate the ability of decentralised control to handle these kinds of operational disturbances. A recommendation for future research would be to expand the model with the operation that takes place at the gate and on the aprons.

Research approach

The unique aspect of this research is the parallel approach of an empirical study and model-based study. Including an empirical study makes it possible to obtain a clear overview of the actual operation. Also, comparisons of the simulated and actual operation lead to the understanding of the mechanisms of decentralised control, as well as the possibilities of the modelling technique Agent-Based Modelling (ABM). The importance of approaching a research question from an operational perspective is highly undervalued. When developing socio-technical models, it is recommended to follow a similar approach, which is to support the model-based study with an empirical study. This allows the researcher to assess the decision making process of the simulated operation with respect to the actual operation. The knowledge gathered from this comparison reinforced the development process, as it was made very clear on what aspects the model had to be improved. The operational foundation of this approach makes it possible to take the simulation model to the requested level of operational detail, and to draw practically useful conclusions.

Bibliography

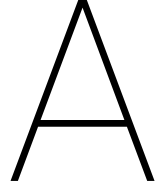
- [1] Airbus. A320. <http://www.aircraft.airbus.com/aircraftfamilies/passengeraircraft/a320family/a320/>, n.d. [Online; accessed 7 November 2017].
- [2] H. Alders. Aldersadvies Nieuwe Normen en Hanshavigstelsel. Technical report, Omgevingsraad Schiphol, Amsterdam, 2013.
- [3] H. Alders. Aldersadvies Nieuwe Normen en Hanshavigstelsel Bijlage 3: Uitwerking Normen. Technical report, Omgevingsraad Schiphol, Amsterdam, 2013.
- [4] J. A. D. Atkin, E. K. Burke, J. S. Greenwood, and D. Reeson. On-line decision support for take-off runway scheduling at London Heathrow Airport. *Journal of Scheduling*, 323(11):1–44, 2008.
- [5] H. Balakrishnan and Y. Jung. A Framework for Coordinated Surface Operations Planning at Dallas-Fort Worth International Airport. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, pages 1–19, 2007. ISBN 978-1-62410-015-4. doi: 10.2514/6.2007-6553. URL <http://arc.aiaa.org/doi/abs/10.2514/6.2007-6553>.
- [6] E. Bonabeau. Agent-based Methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences of the USA*, 99(supplement 3):7280–7287, 2002. ISSN 00278424. doi: 10.1073/pnas.082080899.
- [7] Z. K. Chua, M. Cousy, F. André, and M. Causse. Simulating Air Traffic Control Ground Operations: Preliminary Results from Project Modern Taxiing. In *4th SESAR Innovation Days 2014*, 2014.
- [8] International Civil Aviation Organization. Ninth Meeting of Directors of Civil Aviation of the Central Caribbean (C/CAR/DCA/9). Technical report, International Civil Aviation Organization, June 2007.
- [9] International Civil Aviation Organization. DOC 444 ATM/501 - Procedures for Air Navigation Services, 15 edition. Technical report, International Civil Aviation Organization, November 2007.
- [10] International Civil Aviation Organization. TEC/OPS/SEP – 08-0294.SLG - Wake turbulence aspects of Airbus A380-800 aircraft. https://www.eurocontrol.int/sites/default/files/field_tabs/content/documents/nm/airports/airports-wake-vortex-aspects-letter-a380-aircraft1.pdf, July 2012. [Online; accessed 10 November 2016].
- [11] International Civil Aviation Organization. AMOFSG/10-SN No. 14 - Aerodrome Meteorological Observation and Forecast Study Group (AMOFSG). <http://www.icao.int/safety/meteorology/amofsg/AMOFSG%20Meeting%20Material/AMOFSG.10.SN.014.5.en.pdf>, June 2013. [Online; accessed 14 November 2016].
- [12] J. Cohen. *Statistical Power Analysis for the Behavioral Sciences*. Lawrence Erlbaum Associates, New York, 2 edition, 1988. ISBN 0805802835.
- [13] J. Cohen. A power primer. *Psychological Bulletin*, 112, 1992.
- [14] The Dutch Safety Board. Take-off from Taxiway - Amsterdam Airport Schiphol. http://asndata.aviation-safety.net/reports/2010/20100210_B733_PH-BDP.pdf, December 2011. [Online; accessed 9 November 2016].
- [15] EEC/ECAC. Study on Constraints to Grow. Technical Report Volume 1, EUROCONTROL, Brussels, 2001. [Online; accessed 25 October 2016].
- [16] EUROCONTROL. All-Causes Delay to Air Transport in Europe - October 2017. <https://www.eurocontrol.int/sites/default/files/publication/files/flad-oct-2017.pdf>, November 2017. [Online; accessed 01 February 2018].

- [17] EUROCONTROL. Monthly Network Operations Report - overview - October 2017. <https://www.eurocontrol.int/sites/default/files/publication/files/nm-monthly-network-operations-report-overview-october2017.pdf>, November 2017. [Online; accessed 01 February 2018].
- [18] EUROCONTROL. Aircraft Equipage Requirements in the European Commission IRs 1207/2011, 1028/2014 and 2017/386. <https://www.eurocontrol.int/spi-ir>, March 2017. [Online; accessed 7 October 2017].
- [19] EUROCONTROL. Monthly Network Operations Report - Analysis March 2018. <https://www.eurocontrol.int/sites/default/files/publication/files/nm-monthly-network-operations-report-analysis-march-2018.pdf>, April 2018. [Online; accessed 20 April 2018].
- [20] EUROCONTROL. ATFM slots. <http://www.eurocontrol.int/articles/atfm-slots>, n.d. [Online; accessed 21 November 2016].
- [21] J. García, A. Berlanga, J. M. Molina, and J. R. Casar. Optimization of airport ground operations integrating genetic and dynamic flow management algorithms. *AI Communications*, 18(2005):143–164, 2005. ISSN 0921-7126. URL <http://portal.acm.org/citation.cfm?id=1218858>.
- [22] N. Gilbert. *Agent-Based Models*. SAGE Publications, London, 1 edition, 2007. ISBN 9781412949644.
- [23] J-B. Gotteland and N. Durand. Genetic algorithms applied to airport ground traffic optimization. In *Congress on Evolutionary Computation*, pages 544–551, 2003. ISBN 0-7803-7804-0. doi: 10.1109/CEC.2003.1299623.
- [24] Schiphol Group. Groot onderhoud aan de Buitenveldertbaan van Schiphol. <https://nieuws.schiphol.nl/groot-onderhoud-aan-de-buitenveldertbaan-van-schiphol/>, March 2016. [Online; accessed 7 October 2017].
- [25] Schiphol Group. Monthly Transport and Traffic statistics May 2016. <https://www.schiphol.nl/en/schiphol-group/page/transport-and-traffic-statistics/>, June 2016. [Online; accessed 7 October 2017].
- [26] Schiphol Group. 2016 Traffic and transport figures for Schiphol Group. <https://news.schiphol.com/2016-traffic-and-transport-figures-for-schiphol-group/>, January 2017. [Online; accessed 14 September 2017].
- [27] Schiphol Group. Vijf vragen over vliegtuigen. <https://www.schiphol.nl/nl/jij-en-schiphol/pagina/vijf-vragen-over-vliegtuigen/>, n.d. [Online; accessed 7 November 2017].
- [28] Sendobry A. Urvoy C. Klingauf U. Haus, S. Control theoretic concept for intuitive guidance of pilots during taxiing. In *2011 IEEE/AIAA 30th Digital Avionics Systems Conference*, page 6B3–1–6B3–14, 2011.
- [29] G. Hunter. Empirical Evaluation of Impact of Reconfiguration Events on Airport Capacity. In *12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference and 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, pages 1–14, 2012. ISBN 978-1-60086-930-3. doi: 10.2514/6.2012-5534.
- [30] IATA. IATA forecasts passenger demand to double over 20 years. <http://www.iata.org/pressroom/pr/Pages/2016-10-18-02.aspx>, October 2016. [Online; accessed 18 March 2018].
- [31] H. Idris, J.-P. Clarke, R. Bhuvra, and L. Kang. Queuing Model for Taxi-Out Time Estimation. *Air Traffic Control Quarterly*, 10(1):1–22, 2002.
- [32] H. R. Idris and R. J. Hansman. Observation and analysis of departure operations at Boston Logan International Airport. Technical Report November, MIT International Center for Air Transportation, Cambridge, 2000.
- [33] International Civil Aviation Organization. *Annex 2 to the Convention on International Civil Aviation - Rules of the Air*. International Civil Aviation Organization, 10 edition, 2005. ISBN 5149548022. URL [http://www.icao.int/Meetings/anconf12/DocumentArchive/an02_cons\[1\].pdf](http://www.icao.int/Meetings/anconf12/DocumentArchive/an02_cons[1].pdf).

- [34] International Civil Aviation Organization. *ICAO Doc 9871, Technical Provisions for Mode S Services and Extended Squitter*. International Civil Aviation Organization, 2008. ISBN 9789292311179.
- [35] Y. Jiang, Z. Liao, and H. Zhang. A collaborative optimization model for ground taxi based on aircraft priority. *Mathematical Problems in Engineering*, 2013:1–9, 2013. ISSN 1024123X. doi: 10.1155/2013/854364.
- [36] Y. Jiang, X. Xu, H. Zhang, and Y. Luo. Taxiing route scheduling between taxiway and runway in hub airport. *Mathematical Problems in Engineering*, 2015:1–15, 2015. ISSN 15635147. doi: 10.1155/2015/925139.
- [37] H. Khadilkar. *Analysis and Modeling of Airport Surface Operations*. Master, Massachusetts Institute of Technology, 2011.
- [38] T.K. Kim. T test as a parametric statistic. *Korean J Anesthesiol*, 68(6):540–546, 2015.
- [39] KLM. KLM schaft 17 Embraer vliegtuigen aan ter vervanging van de Fokker 70. <http://nieuws.klm.com/klm-schaft-17-embraer-vliegtuigen--aan-ter-vervanging-van-de-fokker-70/>, March 2015. [Online; accessed 7 October 2017].
- [40] S. G. Knijnenburg. *planning and simulation of taxiing aircraft using linear programming*. PhD thesis, Delft University of Technology, 1998.
- [41] G. J. M. Koeners and R. M. Rademaker. Ground control support functions to optimize surface traffic flow in a CDM environment. In *AIAA/IEEE Digital Avionics Systems Conference*, volume October, pages 1–11, 2012. ISBN 9781467316996. doi: 10.1109/DASC.2012.6382353.
- [42] G. J. M. Koeners, E. P. Stout, and R. M. Rademaker. Improving taxi traffic flow by real-time runway sequence optimization using dynamic taxi route planning. In *AIAA/IEEE Digital Avionics Systems Conference*, pages 1–12, 2011. ISBN 9781612847979. doi: 10.1109/DASC.2011.6095993.
- [43] F. Lancelot, M. Causse, N. Schneider, and M. Mongeau. Human-in-the-Loop Multi-agent Approach for Airport Taxiing Operations. In J. Bajo, J. Z. Hernández, P. Mathieu, A. Campbell, A. Fernández-Caballero, M. N. Moreno, V. Julián, A. Alonso-Betanzos, V. Botti, and M. D. Jiménez-Lopéz, editors, *Trends in Practical Applications of Agents, Multi-Agent Systems and Sustainability: The PAAMS Collection*, pages 235–236. Springer International Publishing, Heidelberg, 2015. ISBN 9783319196282. doi: 10.1007/978-3-319-19629-9.
- [44] H. Lee and H. Balakrishnan. Fast-time simulations of Detroit Airport operations for evaluating performance in the presence of uncertainties. In *AIAA/IEEE Digital Avionics Systems Conference - Proceedings*, pages 1–13, 2012. ISBN 9781467316996. doi: 10.1109/DASC.2012.6382349.
- [45] H. Lee and H. Balakrishnan. A Comparison of Two Optimization Approaches for Airport. In *31st Digital Avionics Systems Conference*, pages 1–15, 2012. ISBN 9781467319003.
- [46] H. Lee, I. Simaiakis, and H. Balakrishnan. A comparison of aircraft trajectory-based and aggregate queue-based control of airport taxi processes. In *AIAA/IEEE Digital Avionics Systems Conference - Proceedings*, pages 1–15, 2010. ISBN 9781424466160. doi: 10.1109/DASC.2010.5655526.
- [47] Luchtverkeersleiding Nederland. Voorschriften Dienst Verkeersleiding. Technical report, Luchtverkeersleiding Nederland, 2014.
- [48] W. Malik, G. Gupta, and Y. C. Jung. Managing departure aircraft release for efficient airport surface operations. In *AIAA Guidance, Navigation, and Control Conference*, pages 1–15, 2010. ISBN 978-1-60086-962-4. doi: 10.2514/6.2010-7696. URL <http://arc.aiaa.org/doi/pdf/10.2514/6.2010-7696>.
- [49] W. A. Malik, G. Gupta, and Y. C. Jung. Spot Release Planner : Efficient Solution for Detailed Airport Surface Traffic Optimization. In *12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference and 14th AIAA/ISSM*, pages 1–9, 2012. ISBN 9781600869303.
- [50] A. Marín and E. Codina. Network design: Taxi planning. *Annals of Operations Research*, 157(1):135–151, 2008. ISSN 02545330. doi: 10.1007/s10479-007-0194-0.

- [51] Á. G. Marín. Airport management: Taxi planning. *Annals of Operations Research*, 143(1):191–202, 2006. ISSN 02545330. doi: 10.1007/s10479-006-7381-2.
- [52] Y. Peng, G. Wei, S. Junqing, and S. Bin. Evaluation of Airport Capacity through Agent Based Simulation. *International Journal of Grid Distribution Computing*, 7(6):165–174, 2014.
- [53] B. Pesic, N. Durand, and J.-M. Alliot. Real-World Applications Aircraft Ground Traffic Optimisation using a Genetic Algorithm. In *GECCO conference*, pages 1–8, 2001.
- [54] S.M. Pincus. Approximate entropy as a measure of system complexity. *Proceedings of the National Academy of Sciences*, 88:2297–2301, 1991.
- [55] C. V. Rafegas. *Taxiway Management Through Multi Agent System*. PhD thesis, Universitat Autònoma de Barcelona, 2013. URL <https://ddd.uab.cat/pub/tfg/2013/113652/ValderramaRafegasCarolina-TFGAa2012-13.pdf>.
- [56] D. B. Rappaport, P. Yu, K. Griffin, and C. Davison. Quantitative Analysis of Uncertainty in Airport Surface Operations. In *9th AIAA Aviation Technology, Integration and Operations Conference*, pages 1–16, 2009. URL <http://arc.aiaa.org/doi/abs/10.2514/6.2009-6987>.
- [57] S. Rathinam, J. Montaya, and Y. Jung. An optimization model for reducing aircraft taxi times at the dallas fort worth international airport. In *26th International Congress of the Aeronautical Sciences*, pages 1–14, 2008. ISBN 9781605607153.
- [58] W. R. Richards, K. O'Brien, and D. C. Miller. New air traffic surveillance technology. *Boeing Aero Magazine*, pages 6–13, apr 2010. URL http://www.boeing.com/commercial/aeromagazine/articles/qtr_02_10/pdfs/AERO_Q2-10_article02.pdf.
- [59] J.S. Richman and J.R. Moorman. Physiological time-series analysis using approximate entropy and sample entropy. *American Journal of Physiology - Heart and Circulatory Physiology*, 278(6):2039–2049, 2000.
- [60] P. C. Roling and H. G. Visser. Optimal Airport Surface Traffic Planning Using Mixed-Integer Linear Programming. *International Journal of Aerospace Engineering*, 2008:1–11, 2008. ISSN 1687-5966. doi: 10.1155/2008/732828.
- [61] Flight Safety Foundation. FSF ALAR BRIEFING NOTE 8.4 - Braking Devices. Technical report, Flight Safety Foundation, Alexandria, 2009.
- [62] Schiphol Group. Onderhoud aan de Zwanenburgbaan. <https://nieuws.schiphol.nl/onderhoud-aan-de-zwanenburgbaan/>, May 2016. [Online; accessed 7 October 2017].
- [63] O.A. Sharpanskykh. Lecture notes in Agent-Based Modelling and Simulation in Air Transport, March 2018.
- [64] I. Simaiakis. *Analysis, modeling and control of the airport departure process*. Doctoral, Massachusetts Institute of Technology, 2013. URL <http://dspace.mit.edu/handle/1721.1/79342>.
- [65] I. Simaiakis and H. Balakrishnan. Queuing Models of Airport Departure Processes for Emissions Reduction. In *AIAA Guidance, Navigation, and Control Conference*, pages 1–26, 2009. ISBN 978-1-60086-978-5. doi: 10.2514/6.2009-5650. URL <http://arc.aiaa.org/doi/abs/10.2514/6.2009-5650>.
- [66] I. Simaiakis, H. Balakrishnan, and M. Sandberg. Dynamic control of airport departures: Algorithm development and field evaluation. *IEEE Transactions on Intelligent Transportation Systems*, 15(1):285–295, 2014. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6315592.
- [67] J. W. Smeltink, M. J. Soomer, P. R. de Waal, and R.D. van der Mei. An Optimisation Model for Airport Taxi Scheduling. Preprint submitted to Elsevier Science, 2004. URL <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.116.9774>.
- [68] S.H. Stroeve and M.H.C. Everdij. Agent-based modelling and mental simulation for resilience engineering in air transport. *Safety Science*, 93:29–49, 2017. doi: <https://doi.org/10.1016/j.ssci.2016.11.003>.

- [69] R. Tamilselvan and P. Balasubramanie. A Genetic Algorithm with a Tabu Search (GTA) for Traveling Salesman Problem. *International Journal of Recent Trends in Engineering*, 1(1):2–5, 2009.
- [70] Air Traffic Control the Netherlands. Record number of take-offs and landings at Schiphol on a single day. <https://news.lvn1.nl/record-number-of-take-offs-and-landings-at-schiphol-on-a-single-day/>, June 2016. [Online; accessed 14 September 2017].
- [71] Air Traffic Control the Netherlands. Meteorological influences. <https://www.lvn1.nl/en/environment/route-and-runway-use/meteorological-influences.html>, n.d. [Online; accessed 14 November 2016].
- [72] H. Udluft. Agent-Based Simulation of Decentralized Control for Taxiing Aircraft. <http://stefjanssen.com/AimsPresentations/udluft.pdf>, December 2016. [Online; accessed 2 October 2017].
- [73] H Udluft. *Decentralization in Air Transportation*. PhD thesis, Delft University of Technology, 12 2017.
- [74] H. Van Dyke Parunak, R. Savit, and R. L. Riolo. Agent-Based Modeling vs. Equation-Based Modeling: A Case Study and Users' Guide. In *Proceedings 1st International Workshop on Multi-Agent Systems and Agent-Based Simulation*, pages 10–25, Berlin Heidelberg, 1998. Springer. ISBN 978-3-540-65476-6. doi: 10.1007/10692956_2.
- [75] G.M. Wittenbaum, S.I. Vaughan, and G. Strasser. Coordination in task-performing groups. In Tindale R.S., Heath L., Edwards J., Posavac E.J., Bryant F.B., Suarez-Balcazar Y., HendersonKing E., and Myers J., editors, *Social psychological applications to social issues, Vol. 4. Theory and research on small groups*, pages 177–204. Plenum Press, New York, 1998.



Deriving the travelled Routes from ADS-B Data

This appendix provides an elaboration of a number of steps that have been, to determine the ground paths that have been travelled in the historical operation. Section A.1 covers the process to determine the nearest segment of each Automatic Dependent Surveillance-Broadcast (ADS-B) data point. Section A.2 explains the procedure that is followed in case multiple ADS-B data points have a similar time stamp. The exclusion of aircraft travelling backwards along their route is described in Section A.3.

A.1. Determine nearest Segment

The nearest segment of each ADS-B data point can be done by iterating over all taxiway segments in the airport layout. However, it is found that the performance of this strategy is way too slow due to the large number of segments, data points and flights. Consequently, an adapted algorithm is written, which limits the number of potential closest segments for each data point.

To create a limited set of potential segments, the centre coordinates of each segment are determined. Also, the segment length of the longest taxiway is found, which is 2388 metres for AAS. It is assumed that the maximum distance between an aircraft and the centre of the segment it is on, is half of the maximum distance. Adding a safety margin of 10% gives a radius of interest r_{interest} of $(0.5 \cdot 2388) \cdot 110\% = 1313$ metres. Subsequently, the algorithm selects all segments that have their centre within range of $p_j\%$ of the identified radius from the ADS-B data point. Only for this set of segments, the closest distance between segment and data point are calculated, while keeping track of the closest one. If one of the segments has a perpendicular distance smaller than 5 metres, the algorithm chooses this segment as the one the aircraft is on. If none of the segments in the set is within a 5 metres range, a larger percentage p_{j+1} is selected. For all segments that are in set $j + 1$ but not in j , distances are calculated and the closest one is stored. The set of percentages p_j that are checked are 10%, 25%, 50%, 75% and 100% of the identified radius of 1313 metres. If none of the segments is within a distance of 5 metres, the closest segment is stored. The algorithm has been changed to iteratively test the most promising segments based on the distance towards the segment's centre, instead of checking all segments at once. An overview of this process is given in Figure A.1

Additionally, from the second data point of each flight onwards, it is checked whether the segment identified for the previous node is within a distance of 5 metres. If this is the case, this segment is selected, otherwise the process continues as mentioned above.

This process is repeated for each of the flights. Equation (A.1) is used to determine the weight of segment i , based on the length of this segment s_i and the number of data points it is closest to ($x_{\text{closest},i}$). Consequently, the weight of a segment is inversely proportional to the number of times it is closest to an ADS-B point.

$$w_i = \frac{s_i}{x_{\text{closest},i} + 1} \quad (\text{A.1})$$

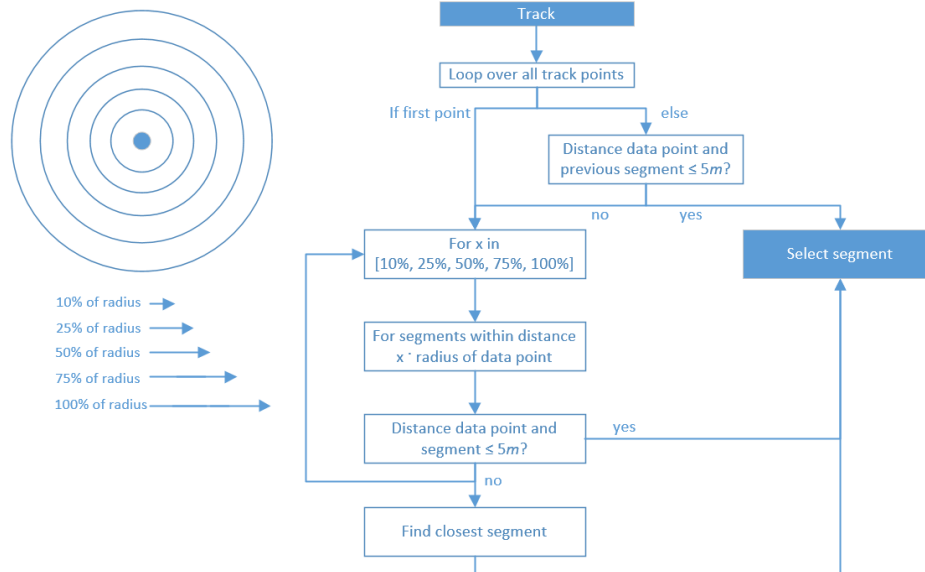


Figure A.1: Process to determine the closest segment

A.2. Shift ADS-B Points with similar Time Stamps

The time stamp of the stored ADS-B data point has a 1 second accuracy, allowing multiple data points to have a similar time stamp. Before the actual interpolation can take place, something has to be done about these data points that have similar time stamps. Procedures have been implemented, which aim at shifting some of the points to keep as much detail in the actual operation as possible. This procedure looks at the number of nodes that have the same time stamp, as well as whether there is a time gap between the set of double nodes and the ones in front or after it.

Insights in the procedure are shown in Figure A.2:

- The most simple process takes place when a set of double points occurs at the beginning of the track, or at the end of the track. In the former case, the last node of the set keeps its time stamp, while the others are shifted backwards in time, making sure the time difference between them is 1 second. The same strategy is applied for the latter case, where all nodes apart from the first one are shifted forward in time.
- If the set of double points occurs in the middle of the flight, it has to be checked what the time gap between the previous node and the set of double points is, and the gap between the set of double points and the next node. Depending on the number of points having the same time stamp, it is decided what strategy is applied.
- If both gaps in front of and behind the double set are > 1 second, it is checked which of the two time gaps is larger and one of the two nodes in the set is shifted in that direction. If there are two points, and the time gap in front is at least 2 seconds, the first point is shifted backwards. Otherwise, it is checked whether the time gap behind is 2 seconds, after which the last point is shifted forwards. If there is no gap in front or behind, the last point is removed.
- If there are three points, a similar line of thoughts is used. However, the middle point will always be fixed. For the first point of the double set, it is checked whether it can be shifted backwards. If not, it is removed. The same holds for the last point: if it cannot be shifted forward, it is removed.
- The only thing that changes for double sets of more than 3 nodes, is the fact that all nodes except for the first node, median node (rounded upwards) and last node are deleted straight away. For the remaining nodes the same strategy is applied as for the case in which there are three points with the same time stamp.

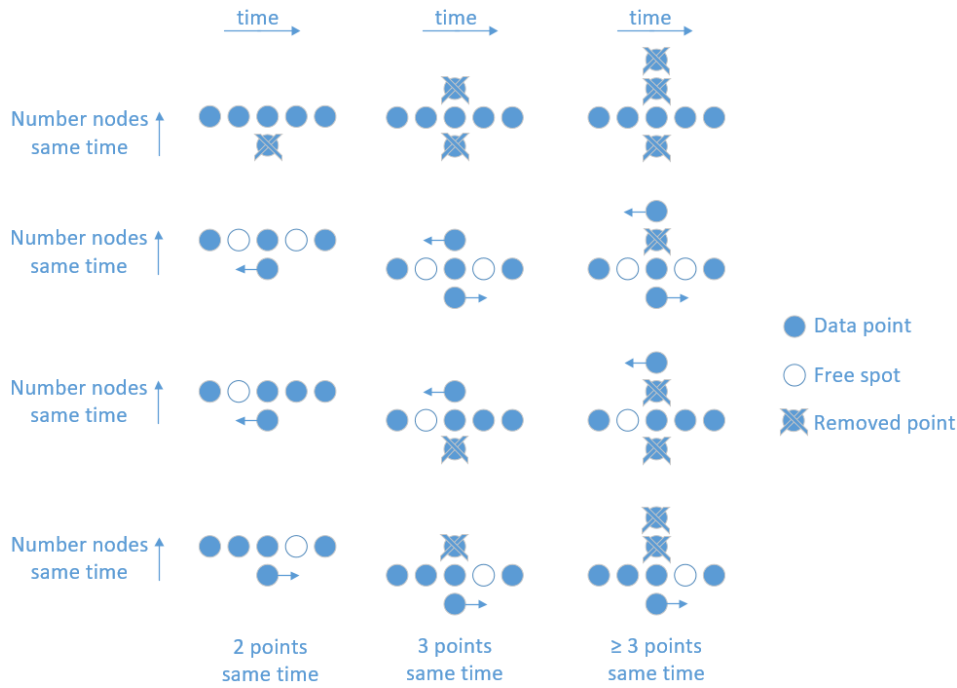


Figure A.2: Procedure to shift data points with similar time stamps

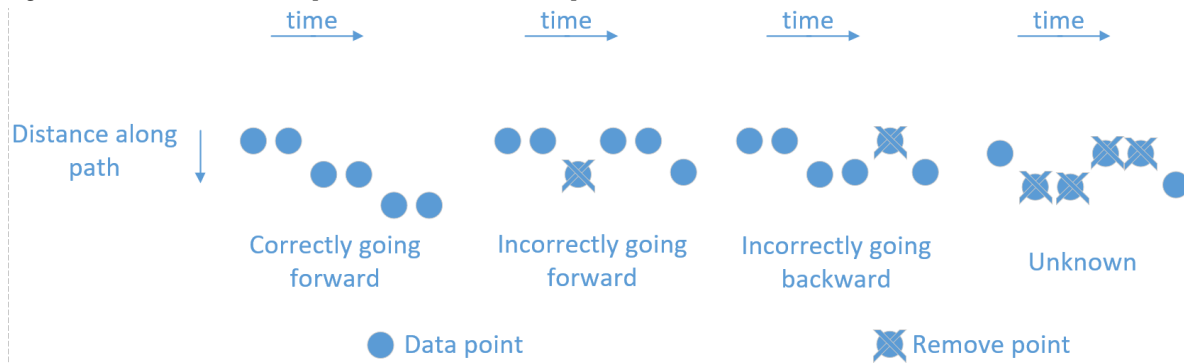


Figure A.3: Procedure to remove data points when going forward in time but backwards in distance

A.3. Prohibit Aircraft from going Backwards

A final step before interpolation can take place, is to project each of the ADS-B data points on the ground path based on the shortest orthogonal distance. It is assumed that an aircraft can only move forward along the path. This assumption is necessary, as it is found that for some flights a set of nodes is later in time, but backwards along the route, due to e.g. push backs on the taxiway or measuring errors. To determine which of the data points has to be removed, a procedure is created which is shown in Figure A.3. Looking at the nodes in chronological order, two sets of points are created: one set that is more forward along the path, but earlier in time, and one set that is more backward along the path but later in time. If the number of points in set 1 is larger than in set 2, the data points in set 2 are removed. If it is the other way around, the data points in set 1 are removed. In case both sets have the same size, all points in both sets are removed as no clear conclusion can be drawn.

B

In-depth Explanation Baseline Model

This research continues on earlier work done by Udluft [73], who designed an ABM tool to simulate ground operations. This appendix provides an in-depth explanation of the agent-based model that has been developed to demonstrate the feasibility of decentralised control. Information regarding the modelled operation and the layout used are found in Section B.1, as well as a summary of the main findings. As an agent-based model is specified by its environment, its agents, and the interaction among agents and the environment. Each of these aspects is covered in Section B.2, B.3, and B.4 respectively.

B.1. Model Operation and Main Findings

Udluft's aim was to demonstrate that decentralised air traffic control is a feasible technique to perform ground operations. He did so by placing local agents on each of the taxiway intersections, as well as on gates and runway nodes. A representation of the concept is shown in Figure B.1. More information regarding the individual agents can be found in Section B.3.

A simple airport layout is used to simulate the ground operations. The layout, visible in Figure B.2, consists of 3 gates, 18 intersections, 2 runways with two runway entries each, and 32 segments connecting it all. The main operation in this study consists of departing traffic only. Aircraft are spawned at a gate according to a flight schedule, which is randomly created in terms of departure time, origin (gate), and destination (runway entry). Once an aircraft has left its gate, intersection agents are responsible for guiding the aircraft agents safely and efficiently to their destination. The moment an aircraft agent reaches the runway entry, it is removed from the network.

An auction technique has been implemented as way of communication between agents. By letting ATC agents bid on segments depending on the aircraft they are responsible for, the model aims at supplying the limited resources to the intersection agent that needs them the most. Tests have been performed to analyse the impact of the scope of coordination, meaning that the bid propagated either to 0, 1, 2 or 3 agents. The former means that no bidding, and therefore no coordination, takes place regarding the utilisation of the resources. More information regarding means of communication between agents can be found in Section B.4. A summary of the auction protocol that is used to let ATC agents coordinate, is found in Appendix C.

The research analysed the impact of the scope of coordination, by running Monte Carlo simulations [73]. These experiments are based on simulating the ground operations for different spawn rates (number of released aircraft per hour) for each of the scope of coordination values. Main findings are that the scope of coordination has a positive effect on the operation as it reduces taxi time, as long as the spawn rate is above a certain limit. Below this limit, the number of aircraft in the network is too small, meaning that aircraft agents can travel along their optimal route without encountering other aircraft. Consequently, no intervention of ATC agents is required to distribute the usage of resources by means of auctions, and thereby the scope of coordination has no impact on the taxi time.

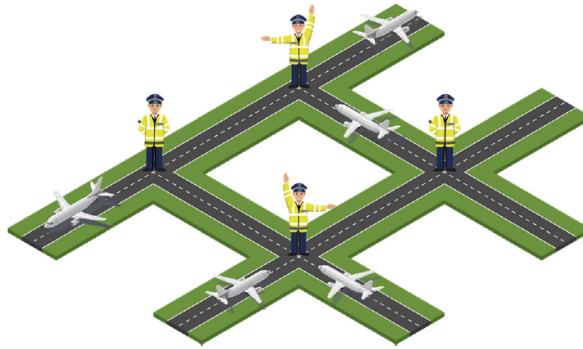


Figure B.1: Decentralised Air Traffic Control [72]

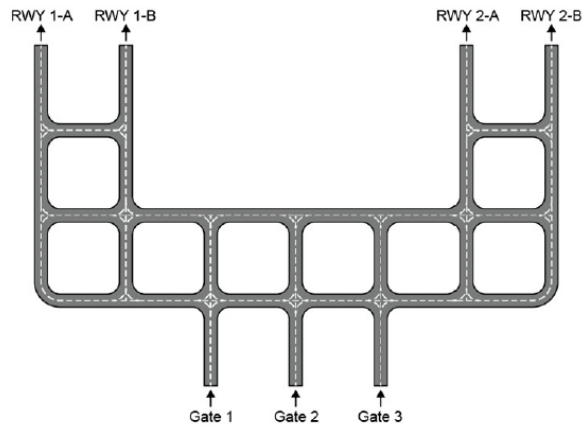


Figure B.2: Initial airport layout [72]

B.2. Model Environment

This section covers the modelled environment, which consists of non-agent objects. An environment is essential for ABM as it allows agents to observe this environment and possibly act upon it. It thereby offers an indirect way of communication between the individual agents.

The environment of this initial model consists of a directed graph illustrating a taxiway infrastructure. The gates, runway entries/exits, and taxiway intersections are represented by a node, while the graph edges define the taxiway segments. Each directed edge possesses a weight, which could be altered by the adjacent agents.

The environment is considered dynamic, as links can be removed and added by ATC agents to prevent conflicts from happening. Additionally, every ten seconds a ATC agent checks the speed that is driven along each of its incoming taxiways. This speed is used to determine the average speed of the last minute, which is an input to update the segment's weight. The ATC agent is responsible for altering the weight of all its incoming segments. The environment is accessible as each ATC agent has access to the complete directed graph at each moment in time. Based on knowledge of the previous state of the graph in combination with the actions of aircraft agents, the next state of the environment can be predicted. Since there are no external factors that can alter the environment, the environment is found to be deterministic.

B.3. Types of Agents

Six types of agents are introduced in the model, of which 5 are related to ATC and the other one is the aircraft agent. A summary of each agent is given in this section.

Starting with the aircraft agent, which is seen as a point mass that is allowed to move in two dimensional space. This aircraft follows commands from ATC agents, while making sure it maintains a safe distance from the other aircraft agents. The movements of an aircraft depend on its current location in Cartesian coordinates (x, y) , its heading h , current speed v , and acceleration acc . Its heading originates from the shortest

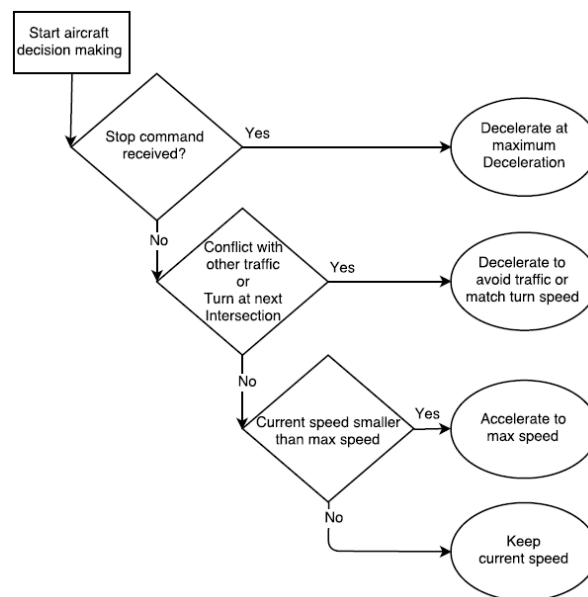


Figure B.3: Decision making process of an aircraft agent's acceleration [73]

path as found by the ATC agents, while the acceleration is set based on the following protocol:

1. An aircraft agent aims at travelling at maximum taxi speed v_{\max} and accelerates to this speed if the situation allows;
2. An aircraft agent follows stop and routing commands from ATC agents;
3. If an aircraft agent has to make a turn, its speed has to be equal to, or smaller than, the maximum turn speed v_{turn} ;
4. If an aircraft agent has received a stop command, it decelerates until it has zero speed;
5. An aircraft agent's position, destination, and route is being broadcast to other agents;
6. An aircraft agent is able to detect a conflict with other aircraft agents that are either taxiing on the same taxiway, on the taxiway it will taxi on next, or travelling towards the same node.

The decision making process applied by aircraft agents to determine its acceleration is shown in a flowchart in Figure B.3. Each aircraft agent performs this process for every time step.

Decentralised ATC is implemented in terms of five different types of agents. The first ATC agent is the intersection agent, which is placed at each taxiway intersection, runway entry point, and at gate exit point. This agent is responsible for controlling the aircraft agents and does so by sending routing updates to aircraft and guiding them accordingly. Additionally, this controller prevents and solves conflicts between aircraft agents by giving stop commands. Each intersection agent is responsible for all incoming aircraft agents entering from one of the adjacent taxiways. It has knowledge of the current state of the environment and thereby the taxiway system. Additionally, by means of radar functionalities the speed of the incoming aircraft could be determined, in addition to the location and current route.

For each aircraft agent under control, the intersection agent continuously determines the shortest paths towards its destination by applying the Dijkstra algorithm. For this calculation, the current state of the network is used, meaning that direct links are removed in case aircraft agents are moving in opposite direction. Additionally, aircraft can remove an incoming link in order to guarantee it always has an outgoing link allowing conflicts to be solved successfully. In case no route towards the destination is found, the intersection agent tries to find a path towards one of nodes included in the current route. This is done in an iterative manner, starting with the node closest to the destination. In case no path is found towards the node at distance 2, a stop command is given to that specific aircraft agent.

To maintain a safe operation, each intersection agent has to make sure only one aircraft agent can simultaneously be within a radius equalling the separation distance. If no aircraft is within this domain, the intersection agent will focus on handing off the first aircraft approaching the node. It does so by reserving itself as well as the next segment for that particular aircraft, while halting all other aircraft at the separation

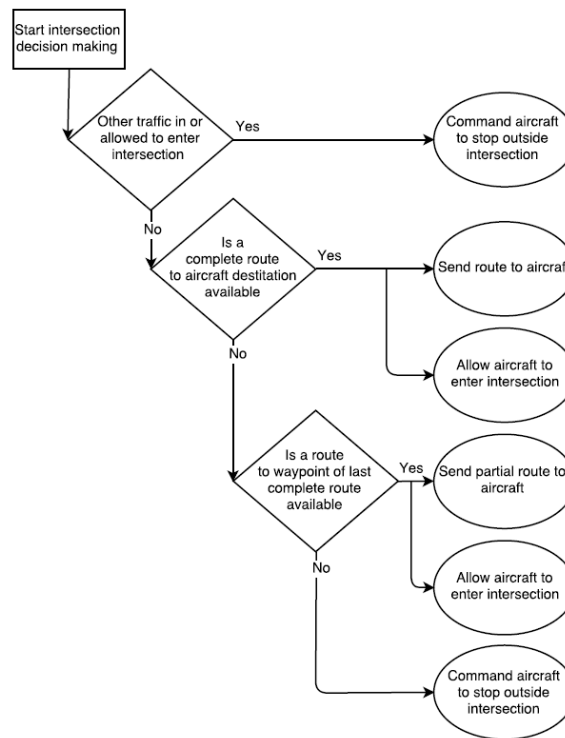


Figure B.4: Decision making process of an intersection agent's hand off [73]

distance. This decision making process is visualised in Figure B.4. This process is executed by each intersection agent at each time step.

Wherever a gate is located, a gate agent is placed on top of a gate intersection agent. The gate intersection agent is a simplified version of the intersection agent, as this agent does not have to attract aircraft since they all start at the gate. The former agent acts as a source, keeping a waiting list in which aircraft are added in case the simulation time is equal to the aircraft's departure time, the gate agent checks whether the outgoing taxiway segment is available and no other aircraft agent is within the required separation distance from the gate. If both conditions are met, the aircraft agent is being released and handed over towards the gate intersection agent below it. Otherwise, the gate agent attempts releasing the aircraft agent in the next step. This gate agent is seen as a source agent.

One runway agent is allocated per runway and is responsible for safely controlling the traffic willing to take off. The moment an aircraft agent arrives at a runway entry (runway intersection agent), it is added to the runway agent's waiting list. The runway agent allows the first aircraft in the queue to enter the runway, by telling the runway intersection agent that the aircraft can be handed off. The runway intersection agent acts as a sink as it is responsible for removing the aircraft from the network. This particular agent is a simplification of the intersection agent as it only has functionalities to attract aircraft agents. A set occupancy time is applied as minimum separation time between two take offs.

B.4. Interaction among Agents and with Environment

This section goes into more detail regarding the interaction that takes place between the different types of agents. Additionally, the interaction between each type of agent and the environment is discussed.

Figure B.5 provides a schematic overview of the interactions that take place. Each of these interactions is discussed below:

1. The gate agent communicates with its gate intersection agent to determine whether an aircraft agent is within separation distance from this later agent, to know whether a new aircraft can be released safely. If possible, the aircraft agent is being handed over towards the gate intersection agent.

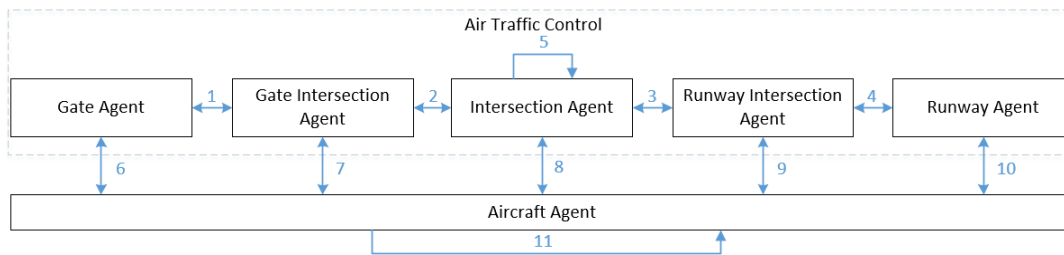
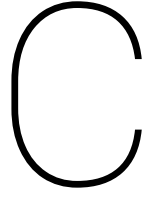


Figure B.5: Diagram with interaction between agents

2. The gate intersection agent and intersection agent communicate with each other about the state of the intersection agent to determine if an aircraft can be handed over safely towards the intersection agent. This includes the transmission of the location of surrounding aircraft.
3. Intersection agents communicate with each other to perform auctions on which agent is allowed to use the segment in between them next. The implemented coordination technique to perform the auction is covered in Appendix C. Additionally, the location of aircraft agents surrounding an intersection agent is communicated towards the other intersection agent, as well as the current activity of this agent, in order to determine whether hand off is possible.
4. The intersection agent and runway intersection agent communicate with each other about the state of the runway intersection agent to determine if an aircraft can be handed over safely towards the runway intersection agent.
5. The moment an aircraft agent arrives at the runway entry, this runway intersection agent informs the runway agent that this aircraft has to be added to the runway queue. The moment this aircraft is first in line and the runway is no longer occupied, the runway agent requests the intersection agent to hand over the aircraft.
6. Before spawning an aircraft into the taxiway network, the gate agent sends the aircraft agent a few parameter updates to change its status and to set the time the aircraft has left its gate.
7. When handing over the aircraft towards the next ATC agent, the gate intersection agent provides the aircraft with all information that is required to process the handing off process.
8. An intersection agent sends both stop commands and routing commands towards the aircraft agent, after which the aircraft confirms it has received the message. Radar capabilities allow an intersection agent to observe the aircraft's position and speed, while the aircraft agent broadcasts its current route and heading. Additionally, intersection agents request information from the aircraft agents like their taxi time. In return, the intersection agent sends updates towards the aircraft agent required for its administration.
9. The runway intersection agent is able to send a stop command to the aircraft agent in case the runway is still occupied. Additionally, when the aircraft can enter the runway, the runway intersection agent sends the aircraft agent the required information to process the handing off process.
10. The runway agent observes the aircraft agent until the moment it arrives on the runway, after which it sends the final information to the aircraft required to conclude the administration of its taxi phase.
11. Aircraft agents are able to observe one and another based on the fact that they broadcast their position, heading and route.

Regarding the interaction with the environment, it can be concluded that neither the aircraft agent, gate agent, or runway agent observe the environment. Only the 3 types of intersection agent are able to observe the environment, by reading the most recent directed graph. This graph represents the actual state of the taxiway structure, which is used to determine the shortest path for an aircraft agent. Intersection agents can reserve a segment which is to be used by an aircraft that is about to be handed off by the intersection agent. It does so by removing the opposing link, so the incoming edge, from the network. Additionally, the moment an aircraft agent arrives at an intersection agent, this latter agent adds the segment opposing the aircraft's last travelled taxiway. Additionally, each intersection agent is responsible for monitoring the average speed that is being driven on each of its outgoing taxiways, which it stores in the network.



Coordination Mechanism

This appendix provides an explanation of the coordination mechanism that has been implemented in the baseline model by Udluft [73]. This mechanism is based on performing auctions, as is covered in Section C.1. Section C.2 describes how an ATC agent prepares a bid when taking part in such an auction. Section C.3 describes a few design flaws that have been identified and corrected.

C.1. Auction Concept

The goal of an individual ATC intersection agent is to safely guide all aircraft under its control towards their destination, while aiming to do it as fast and efficient as possible. A coordination mechanism has been introduced by means of auctions, which take place during the decision making process of an intersection agent. This coordination technique is characterised as explicit and is executed by the interaction of the agents, which is why it is categorised as In-process planning by Wittenbaum et al. [75].

The outline of the auction concept is that intersection agents prepare a bid for each of their outgoing segments, based on the interest of all aircraft under control. More information regarding the bid determination is found in Section C.2. Based on the outcome of the auction process, an intersection agent knows which segments it can and which ones it cannot use when determining future routes. Algorithm 1 provides a basic overview of the route decision process of intersection agents, in which the auction concept plays a fundamental role.

An auction process is initiated whenever any aircraft under control is within a specified distance from the intersection agent. For the simulation, this distance s_{auction} is set equal to the separation distance s_{sep} plus the distance required to get to a standstill using comfort deceleration dec_{com} . The intersection determines bid prices for all outgoing segments, such that the aircraft's interests are best met. For each aircraft under control and within s_{auction} , an individual auction is initiated. The auction takes place on the aircraft's most preferred segment. The determined bid is forwarded to the ATC agent on the other side of the segment, which compares the received bid with its own bid. In theory, the agent with the highest bid wins. In case the bids are equal, the agent with the highest ID wins. If the auction receiver wins, the auction initiator continues the bidding process by starting an auction on the second most preferred segment. Otherwise, the bid price of the winning bid is forwarded to the auction receiver, which now becomes the auction initiator. The received bidding budget is merged with the agent's personal budget, after which a similar approach is followed in the direction of the the aircraft's destination. The parameter scope of coordination x_{coor} determines how far the bidding process travels. For this research the scope of coordination is set to 2, meaning the bid is allowed to transfer 2 times and the auction reaches the agent at a geodesic distance of 2. The protocol makes sure the bid cannot travel back to agents that already took part in the auction process.

It should be noted that Section C.3 explains that an important bug has been found in the auction process. Due to this bug a bid would always transfer up to three agents, independent of the fact whether earlier bids were won or lost. Additionally, only the auction on the final segment determined the final outcome of the earlier auctions. So in case the final auction is won, all auctions of this bidding process are won.

After each auction process, the ATC agent that initiated the auctions has a list of both won and lost segments.

Algorithm 1 Plan aircraft routes using coordination [73]

```

1: function PLANAIROUROUTE(IntersectionAgentList,s_auction)
2:   for each IntersectionAgent  $\in$  IntersectionAgentList do
3:     Graph  $\leftarrow$  GetCurrentStateTaxiwaySystem()
4:     AC  $\leftarrow$  GetAircraftUnderControl()
5:     ACclose  $\leftarrow$  GetAircraftWithinDistance(AC,s_auction)
6:     if length(ACclose) > 0 then
7:       BidBudget  $\leftarrow$  AllocateBudget(AC)
8:       for each Aircraft  $\in$  ACclose do
9:         BidsWonLost  $\leftarrow$  PerformAuction(BidBudget,Aircraft)
10:        UpdateWonLostOverview(BidsWonLost)
11:      end for
12:    end if
13:    WonLostSegment  $\leftarrow$  LastWonLostOverview()
14:    TempGraph  $\leftarrow$  Graph
15:    TempGraph  $\leftarrow$  RemoveLostSegments(TempGraphWonLostSegment)
16:    TempGraph  $\leftarrow$  AddWonSegments(TempGraphWonLostSegment)
17:    for each Aircraft  $\in$  AC do
18:      Route  $\leftarrow$  PlanAircraftRoute(TempGraph,Aircraft)
19:      SendRouteCommand(Aircraft,Route)
20:    end for
21:  end for
22: end function

```

Taking a copy of the most recent taxiway graph, the ATC agent adds its won segments if they are not in place yet and removes the lost segments. Based on this updated taxiway graph, Dijkstra's algorithm is used to find the shortest paths for the aircraft under its control.

C.2. Determining Bid Price

One of the properties of aircraft n is its value V_n , set equal to the aircraft's total taxi time. The total budget B_t of an ATC agents is the sum of the values of all N aircraft under its control as displayed in Equation (C.1). Reason for total taxi time as an aircraft's value is that aircraft that the model prefers giving priority to aircraft with longer taxi times. This results in aircraft that have been halted for a longer period due to congestion are able to continue their route. Also, this is in line with ATC which prefers handing off arriving aircraft close to the apron area above departing aircraft to limit workload. Alternative options like taxi delay can be used instead.

The total budget has to be allocated over the node's adjacent taxiway segments. One of the aims of an ATC intersection agent, is to guide the aircraft quickly to their destination. To do so, the highest budget is allocated to the segment that is most likely able to get an aircraft on its shortest route towards its destination. The budget allocation on segment level is based on the ideal taxi times in which no other traffic is encountered. While initialising the ATC agents, each agent receives an overview of the taxi times to get from each of its neighbours to every possible destination. By adding the time it takes to get to each of its neighbours, the agent has insights in the unimpeded taxi times t_s to reach a particular destination when using segment s .

For each aircraft n under control, its individual bid B_n equalling its value V_n is distributed over out all outgoing segments, apart from the segment k it is currently on. The remaining set of segments is noted as S . The portion of the budget that is allocated to segment s is proportional to the inverse of the unimpeded taxi time t_s , in relation to all possible segments S . Consequently, a segment that takes half the time to reach the destination compared to another gets allocated twice the budget of the other segment. Equation (C.2) defines how aircraft n , which is taxiing along segment k , allocates its budget on all segments S .

After the budgets of all individual aircraft N are allocated, the ATC agent can determine the total bid that is allocated on each of its outgoing segments. The total bid on segment s is given by Equation (C.3).

$$B_T = \sum_{n=1}^N V_n \quad (\text{C.1})$$

$$B_s^n = \begin{cases} 0, & \text{if } s = k \\ \frac{B_n}{\sum_{i=1}^S \frac{t_s}{t_i}}, & \text{otherwise } s \in S, s \neq k \end{cases} \quad (\text{C.2})$$

$$B_s = \sum_{n=1}^N B_s^n \quad (\text{C.3})$$

C.3. Design Flaws

A few design flaws have been identified. An explanation and correction is given for each of them.

Equal bid

In case of an equal bid on a particular segment, the ATC agent with the highest ID number would win the link. This has been changed to the initiator of the auction, as an auction is only initiated in case an aircraft is willing to use it.

Bid propagation lost bid

Auctions are being propagated throughout the taxiway layout along at least x_{COOR} segments. Although it is being recorded whether a segment is won, the bid is always propagated towards the next segment. This even happened in case the auction on the previous segment has been lost. If the last segment is won automatically all segments are considered to be won, meaning only the final segment is vital to the auction process. This issue has been solved by implementing a condition, which states that the auction process only propagates in case the previous auction is won.

Equal bid

An ATC agent could initiate an auction on a forced outpath of another ATC agent, which it was even able to win. In some cases, this caused all aircraft under control to hold stationary as a route update was required to escape local congestion. This issue is easily solved by making sure that auctions on a forced outpath is always won by the ATC agent responsible for this force.

Unimpeded taxi times

The two main steps that take place during the initialisation phase of a simulation is to create the graph and dummy graph, and the (ATC) agents. One step in the agent initialisation process is to generate a dictionary *link_times* with unimpeded taxi time information towards every possible destination via each of its neighbours. From this dictionary, an ATC agent *A* is able to determine that reaching destination *B* via neighbour *N1* takes twice as long as via neighbour *N2*. This has been included in the baseline model to speed up the auction process. Due to this, each ATC agent has an idea of the outgoing segments that are most promising to have, in order to guide an aircraft quickly towards its destination. This data is used to determine the node's auction bid.

It turned out there had been an implementation issue related to the way the unimpeded taxi times are determined. The weights in *link_times* are an estimation of the expected taxi times, in case no other traffic is present. In the baseline model, Dijkstra's algorithm has been used to estimate the taxi time to get from node *A* towards destination *B* via each of its neighbours. An example is shown in Figure C.1. The time to reach node *B* via outgoing segment *A – N1* is based on the taxi time of *A – N1* plus the shortest path between *N1* and *B*. However, for some neighbours the fastest route towards the destination would be via node *A*, either because it is fastest (*N2*) or the only option (*N3*). Consequently, an incorrect timing estimation is given, as such routes are not possible from an operational point of view. An adjustment is made, which is to set use a dummy node as origin (e.g. *A_N2*) to make sure Dijkstra's algorithm does not find a way back. As a consequence, some destinations could no longer be reached when travelling via a particular neighbour, causing the auctions bidding process to be more effective and realistic.

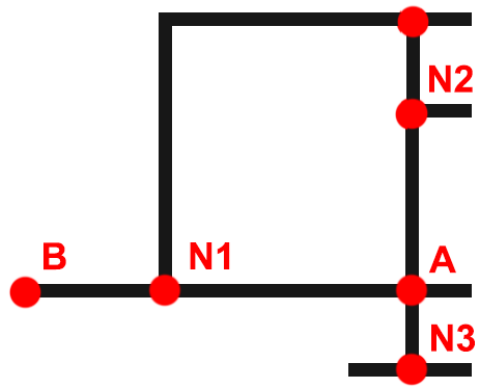


Figure C.1: Establishing of the link_times dictionary



Setting up directed Graph Dijkstra's Algorithm

This appendix gives an overview of the steps that are taken to create the dummy graph, which is used as input for Dijkstra's algorithm. The first step is to initialise the baseline taxiway graph, as is covered in Section D.1, after which Section D.2 describes the process to generate the dummy graph. This dummy graph is a weighted graph, of which the weights are calculated in accordance with the method found in Section D.3. It should be noted that all information in this appendix explains the dummy graph implementation in the baseline model as developed and used by Udluft [73]. A few adjustments have been made to correct some design flaws, as can be read in Section C.3.

D.1. Initialisation of Taxiway Graph

This section presents the algorithm used to initialise the taxiway graph, which is based on two sets of information. One set is *Nodes*, filled with information of all nodes, their location, and type of agent. The second set, *Links*, is an overview of all directed links between the different nodes. The algorithm makes use of the *NetworkX* module and can be found in Algorithm 2. *Nbs* is used as abbreviation of neighbours. Variable *Vmax* represents the maximum taxi speed v_{\max} .

An empty directed graph is created, after which the algorithm runs over the different nodes as present in *Nodes*. Subsequently, the corresponding segments are found for which the node is the source. Taxi time is found by dividing the segment's length by the maximum speed v_{\max} . A weighted link is added to the graph, after which additional information are stored in the link. This information consists of the segment's heading and an empty list called *linked_edges*, in which the related dummy segments are to be stored. This latter aspect is explained in Section D.2. Also the distance and initial average taxi speed are stored. The output is the taxiway graph named *Graph*.

D.2. Initialisation of Dummy Graph

The original taxiway graph is created in Section D.1. Although the shortest path can be determined from this taxiway graph using Dijkstra's algorithm, it is found that routes are selected which require many turns. As an aircraft has to slow down to take these corners, the routes are far from optimal. It is decided to create a different type of graph, such that more realistic routes can be found by Dijkstra's algorithm.

This so-called dummy graph is based on the idea that taking a turn has to be penalised. When calculating the weight of a segment, the next segment is already included to determine whether a turn is required. An example is shown in Figure D.1, which represents the taxiway graph as obtained earlier. For segment $A-D$ multiple dummy links are created. Apart from a copy of $A-D$, three more segments are added, namely $A-D_C$, $A-D_E$, and $A-D_G$; one for each of node D 's outgoing segment. An overview is provided in Table D.1. Nodes C , E , and G are called the target neighbours (*target_nb*) of target node D . As aircraft cannot travel backwards, $A-D_A$ is excluded. To allow for further route continuation, it is required to follow a similar approach for the neighbours of *target_nb*. The three new dummy links are D_E-E_B , D_E-E_F , and D_E-E_H . By following steps for all source nodes, all dummy edges are created. The pseudocode can be

Algorithm 2 Function to create taxiway graph

```

1: function CREATETAXIWAYGRAPH(Vmax,Nodes,Links)
2:   Graph  $\leftarrow$  GetDirectedGraph()
3:   for each Node  $\in$  Nodes do
4:     Graph.AddNode(GetID(Node))
5:   end for
6:   for each Source  $\in$  Nodes do
7:     S_ID  $\leftarrow$  GetID(Source)
8:     Source_loc  $\leftarrow$  GetLoc(Source)
9:     nbs  $\leftarrow$  GetOutgoingLinks(S_ID,Links)
10:    for each Target  $\in$  nbs do
11:      T_ID  $\leftarrow$  GetID(Target)
12:      Target_loc  $\leftarrow$  GetLoc(Target)
13:      Graph.AddLink(S_ID,T_ID)
14:      Distance  $\leftarrow$  Pythagoras(Source_loc,Target_loc)
15:      Weight  $\leftarrow$  Distance/Vmax
16:      Heading  $\leftarrow$  GetHeading(Source_loc,Target_loc)
17:      Linked_edges  $\leftarrow$  list()
18:      Speed  $\leftarrow$  list(Vmax)
19:      Graph[S_ID][T_ID]  $\leftarrow$  StoreInfo(Heading,Linked_edges,Speed,Distance,Weight)
20:    end for
21:  end for
22:  return Graph
23: end function

```

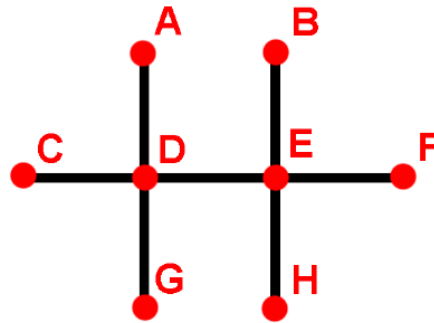


Figure D.1: Concept of the dummy graph

found in Algorithm 3. The original taxiway graph *Graph* and node set *Nodes* are used as input.

For each created edge, different characteristics are derived. Edge $D_E - E_B$ is taken as an example. The source and target are D_E and E_B respectively. The original edge is seen as the first element of the source and target, so $D - E$. A turn is required in case the headings of segments $D - E$ and $E - B$ differ.

If either the target or target's neighbour is an endpoint (so apron or runway), the algorithm finishes creating that segment, after which it continues with the next segment in the *for* loop. It is decided that aircraft leaving the network at an endpoint have to be slowed down to turn speed to either line up on the runway or enter the slow-moving apron area. It is therefore that a turn is required for these segments. An example is given in Table D.1 for the original edge $E - H$.

To create the actual dummy graph, Algorithm 4 is used. All information from the found dummy edges *Dummy_edges* is used as input. Additionally, the original taxiway graph *Graph* and simulation parameters *Constants* are required. While adding the dummy edges to the dummy graph, a connection is made to the original edge from the taxiway graph. Additionally, each segment in the original taxiway graph stores a list with all its linked dummy edges (called *linked_edges*), which is being filled while running the algorithm. The function *GetWeight* is covered in the following section.

Table D.1: Create dummy graph from original graph

Original edge	Dummy edge	Turn
A-D	A-D	-
A-D	A-D_C	True
A-D	A-D_E	True
A-D	A-D_G	False
D-E	D_E-D_B	True
D-E	D_E-D_F	False
D-E	D_E-D_H	True
E-H	E_H-H	True

D.3. Calculation of a Segment's Weight

The taxiway graph is converted to a dummy graph to be able to penalise taking turns. This turning penalty is taken into account when determining the weight of the dummy edges, as is shown in Algorithm 5. Multiple inputs are required like the distance of the dummy edge $Distance$, average speed on this segment $AvgSpeed$, and maximum turn speed $Vturn$. Also the comfort acceleration Acc and deceleration Dec are used, as well as the boolean $Turn$ stating whether a turn is required. In case the average speed on the segment equals zero, the weight is set to infinity. If a turn is required and the average speed is above the required turn speed, the penalty comes in place. This penalty is implemented by determining the required acceleration and deceleration distances, and related times. On the remaining distance aircraft can taxi at the average speed. In all other cases, the weight is the segment's distance divided by the average speed.

D.4. Design Flaws

A few design flaws have been identified. An explanation and correction is given for each of them.

Node numbering endpoints

The setup of the dummy graph is covered in Section D.2. While simulating, it has been observed that in rare cases aircraft were taxiing along one of the main taxiways and decided to use an ATC endpoint as quickest way to make a turn on the taxiway layout.

An example is given in Figure D.2. Due to the situation at stake, aircraft $AC1$ taxiing along segment $A - B$ has to change direction and continue in southern direction. Although one option would be to taxi via B, C, D, E , Dijkstra's algorithm returns route B, G, B, A . This unwanted solution is a negative effect of the way the dummy graph is created. A description of the establishment of the dummy graph is outlined in Section D.2. It explains that the original taxiway segment are duplicated and adapted to the segment one would like to travel on next. So when an aircraft is willing to taxi from node X , via Y towards destination Z , the first dummy segment it uses is $X_Y - Y_Z$. However, the last segment would be $Y_Z - Z$, as destination Z has no more neighbours to include. Additionally, it is explained above that the the routing protocol of ATC is applied from the aircraft's last node instead of its next node. As this is not possible at its origin, there are still dummy segments whose origin is set to an endpoint's ID, like $Z - Y_X$ or $G - B_A$. Based on these understandings, it is clear why the situation as sketched in Figure D.2 leads to dummy path $B_G - G, G - B_A, B_A - A$, which causes a 180° turn at node G .

It is decided that the ID of each endpoint destination i becomes " $i - 1$ ", while the origins remain having ID i . In this way, Dijkstra's algorithm is no longer able to connect these points, meaning aircraft cannot use an endpoint as turning location.

Weight calculation

While replaying the simulated operation, strange routing behaviour was observed. An aircraft might be required to taxi in the opposite direction compared to its current heading, e.g. when it leaves a runway exit which is headed in the opposite direction of its apron. Generally an aircraft would take the first option to perform this turn. In rare occasions, an aircraft remains going straight before making the turn, even when there is no traffic in the area and earlier turnarounds are available. The preference for a longer taxi time and distance could not be explained. Additionally, actual taxi times differed substantially from the taxi time estimations

Algorithm 3 Function to create dummy edges

```

1: function CREATEDUMMYEDGES(Graph,Nodes)
2:   Dummy_edges ← list
3:   for each Source ∈ Graph.nodes do
4:     Source_str ← str(Source)
5:     for each Target ∈ Graph.successors(Source) do
6:       Target_str ← str(Target)
7:       This_edge ← dict()
8:       This_edge["source"] ← Source_str
9:       This_edge["target"] ← Target_str
10:      This_edge["original_edge"] ← [Source,Target]
11:      if Agent_type(Nodes,Target) = ("Runway" or "Apron") then
12:        This_edge["turn"] ← True
13:        Dummy_edges.Add(This_edge)
14:      else
15:        This_edge["turn"] ← False
16:        Dummy_edges.Add(This_edge)
17:        for each Target_nb ∈ Graph.successors(Target) do
18:          if Source ≠ Target_nb then
19:            Target_nb_str ← str(Target_nb)
20:            This_edge ← dict()
21:            This_edge["source"] ← Source_str
22:            This_edge["target"] ← Target_str + "_" + Target_nb_str
23:            This_edge["original_edge"] ← [Source,Target]
24:            This_edge["turn"] ← Diff_heading(Graph,Source,Target,Target_nb)
25:            Dummy_edges.Add(This_edge)
26:            if Agent_type(Nodes,Target_nb) = ("Runway" or "Apron") then
27:              This_edge ← dict()
28:              This_edge["source"] ← Target_str + "_" + Target_nb_str
29:              This_edge["target"] ← Target_nb_str
30:              This_edge["original_edge"] ← [Target,Target_nb]
31:              This_edge["turn"] ← True
32:              Dummy_edges.Add(This_edge)
33:            else
34:              for each Target_nb_nb ∈ Graph.successors(Target_nb) do
35:                if Target ≠ Target_nb_nb then
36:                  Target_nb_nb_str ← str(Target_nb_nb)
37:                  This_edge ← dict()
38:                  This_edge["source"] ← Target_str + "_" + Target_nb_str
39:                  This_edge["target"] ← Target_nb_str + "_" + Target_nb_nb_str
40:                  This_edge["original_edge"] ← [Target,Target_nb]
41:                  This_edge["turn"] ← Diff_heading(Graph,Target,Target_nb,Target_nb_nb)
42:                  Dummy_edges.Add(This_edge)
43:                end if
44:              end for
45:            end if
46:          end if
47:        end for
48:      end if
49:    end for
50:  end for
51:  Dummy_edges ← UniqueList(Dummy_edges)
52:  return Dummy_edges
53: end function

```

Algorithm 4 Function to create dummy graph

```

1: function CREATEDUMMYGRAPH(Graph,Dummy_edges,Constants)
2:   Dummy_graph ← GetDirectedGraph()
3:   Vturn ← Constants["turn_speed"]
4:   Acc ← Constants["acceleration_comfort"]
5:   Dec ← Constants["deceleration_comfort"]
6:   for each Edge ∈ Dummy_edges do
7:     S_ID ← Edge["source"]
8:     T_ID ← Edge["target"]
9:     Turn ← Edge["turn"]
10:    Original_edge ← Edge["original_edge"]
11:    Original_s ← Original_edge[0]
12:    Original_t ← Original_edge[1]
13:    if not Dummy_graph.HasNode(S_ID) then
14:      Dummy_graph.AddNode(S_ID)
15:    end if
16:    if not Dummy_graph.HasNode(T_ID) then
17:      Dummy_graph.AddNode(T_ID)
18:    end if
19:    Graph[Original_s][Original_t]["linked_edges"].Add(list(S_ID,T_ID))
20:    Dummy_graph.AddLink(S_ID,T_ID)
21:    Distance ← Graph[Original_s][Original_t]["Distance"]
22:    Speed ← Graph[Original_s][Original_t]["Speed"]
23:    Weight ← GetWeight(Distance,Speed[0],Vturn,Acc,Dec,Turn)
24:    Dummy_graph[S_ID][T_ID] ← StoreInfo(Turn,Original_edge,Speed,Distance,Weight)
25:  end for
26:  return Graph,Dummy_graph
27: end function

```

Algorithm 5 Function to create dummy weights

```

1: function GETWEIGHT(Distance,AvgSpeed,Vturn,Acc,Dec,Turn)
2:   if AvgSpeed > 0 then
3:     if Turn = True & AvgSpeed > Vturn then
4:       S_acc ← AccelerationDistance(Vturn,AvgSpeed,Acc)
5:       S_dec ← DecelerationDistance(Vturn,AvgSpeed,Dec)
6:       S_cruise ← Distance - S_acc - S_dec
7:       Weight_acc ← (AvgSpeed - Vturn)/Acc
8:       Weight_dec ← (Vturn - AvgSpeed)/Dec
9:       Weight_cruise ← S_cruise/AvgSpeed
10:      Weight ← Weight_acc + Weight_dec + Weight_cruise
11:    else
12:      Weight ← Distance/AvgSpeed
13:    end if
14:  else
15:    Weight ← Inf
16:  end if
17:  return Weight
18: end function

```

and this was found to be mainly the case for aircraft close to their destination. This caused the starting time of future runway modes of operation to be inaccurate. As the estimations of taxi time are based on the graph's current weights, questions were raised whether the weights have been calculated correctly.

Analysis revealed that the initially implemented weight calculations worked fine for the 600 metres long seg-

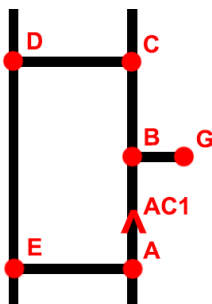


Figure D.2: Route update requiring aircraft to turn around at the apron

ments of the baseline layout (Figure B.2). However, disproportionate weights have been granted to short segments involving turns. As explained in Section D.2, different equations are used to calculate a segment's weight, depending on whether a turn is involved. In case there is, Algorithm 5 estimates the deceleration distance s_{dec} that is required to slow down to v_{turn} to be able to make the turn. Also, an acceleration distance s_{acc} is calculated to get back to v_{max} . Subtracting both distances from the segment's length gives the distance over which an aircraft can travel at maximum speed, s_{cruise} . For the parameter settings specified in Section 5.6, this results in a deceleration and acceleration distance of 411 and 137 metres respectively. Consequently, all segments shorter than 548 metres result in a negative s_{cruise} . The effect is displayed in Figure D.3, which shows the rather high values for short segments that involve a turn. Most segments at AAS are around 100 metres, which results in an estimated taxi time of 23.1 seconds using this approach.

The main issue of this approach is that short segments are relatively heavily penalised. This topic has been interesting food for thought as no clear or simple solution could be found. In principle one would say that all turns should be penalised equally, as they each require an aircraft to slow down to v_{turn} and acceleration to v_{max} afterwards. Complexities rise in case of segments shorter than the earlier mentioned 548 metres, using Figure D.5 as an example. In order to make the turn from segment A_B to B_C , it is required to slow down on segment A_B . Assuming this segment is only 300 metres long, the aircraft is not able to slow down in time from v_{max} to v_{turn} . This means this aircraft should start its initial deceleration on a different segment. In case it comes from segment 1_A this is possible. However, coming from 2_A means the aircraft is already at v_{turn} . So if full deceleration is assumed for calculating corner $2_A \rightarrow A_B$ and full acceleration for $A_B \rightarrow B_C$, segment A_B is penalised twice.

Consequently, for a good taxi time estimation more links should be included apart from the two segments that are used currently. In fact, all segments within a radius of 524 metres should be considered. However, analysing all access routes to get to the segment at stake requires a large amount of information and average speed per taxiway segment. This will drastically increase computational time to update the weights of all segments every 10 seconds, as well as to find the fastest route using Dijkstra's algorithm. Additionally, average speeds are based on historic traffic causing uncertainties for future speeds anyway.

Instead, different objectives are to be considered while improving the current weight calculation: making a turn has to be penalised, the route graph should be simple and updated quickly, and a good estimation of taxi times is preferred. Multiple weighting strategies have been tested and compared based on these conditions. The selected protocol applies a turning penalty to each segment, which is shown in Equation (D.1), in case the dummy edge requires a turn and the applicable speed v_i is larger than the allowed turn speed. The summation is taken of the weight of the segment under investigation, w_i , and the next segment, w_{next} . These two weights are calculated using Equations (D.4) and (D.5) respectively.

For the former, a distinction is made whether the deceleration distance, to go from v_i to v_{turn} , is larger than or equal to the length of the segment. If it is, the maximum allowable speed is found to enter the segment while making sure v_{turn} can be reached. The weight corresponds to the related deceleration time. Otherwise, the deceleration time and remaining cruising time are taken.

Regarding the latter, a correction factor $x_{\text{correction}}$ as found in Equation (D.3). This factor is based on the logic of Figure D.6. In case of a short segment, aircraft can reach speed v before having to slow again to meet the turn speed. The fraction of acceleration and deceleration distance is directly related to the ratio of acceleration and deceleration. So if the acceleration acc_{com} is a third of the deceleration dec_{com} , this translates to the acceleration distance being $3/4$ of the segment's length. This same principle has been applied to come to a correction factor. To make sure the turn penalty is not as high for short segments than in the original sce-

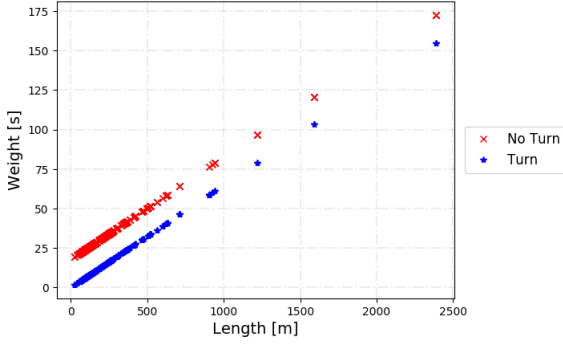


Figure D.3: Original weights per dummy segment

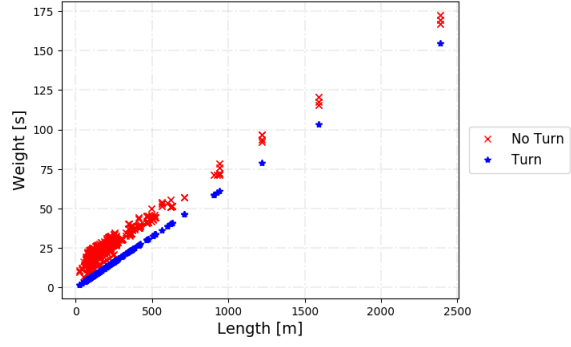


Figure D.4: Updated weights per dummy segment

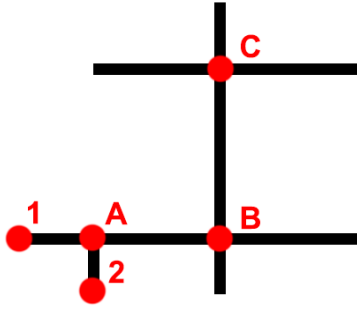


Figure D.5: Complexities to determine the weight of a dummy segment

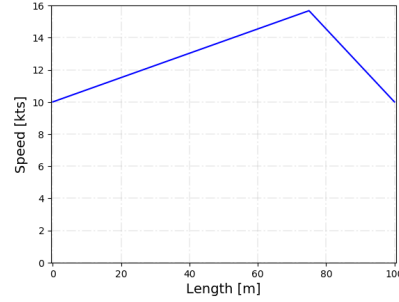


Figure D.6: Aircraft speed profile along a 100 metres long segment that requires two turns

nario, it is decided to only include part of the length of the next segment, namely $x_{\text{correction}} \cdot s_{\text{next}}$. The weight of the next segment, w_{next} , is based on the difference between the acceleration time and the unimpeded time along this defined distance.

The effect of this adjustment is shown in Figure D.4. For the initial weights it is assumed that $v_i = v_{\text{max}}$. From this figure it can be concluded that this more realistic approach, results in a wider range of weights. Also, the found weights are lower on average when compared with the original weights as presented in Figure D.3.

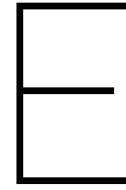
$$w_{\text{total}} = \begin{cases} w_i + w_{\text{next}}, & \text{if Turning and } v_i > v_{\text{turn}} \\ w_{\text{original}} & \text{otherwise} \end{cases} \quad (\text{D.1})$$

$$w_{\text{original}} = \frac{s_i}{v_i} \quad (\text{D.2})$$

$$x_{\text{correction}} = \frac{|dec_{\text{com}}|}{|dec_{\text{com}}| + acc_{\text{com}}} \quad (\text{D.3})$$

$$w_i = \begin{cases} \frac{\sqrt{v_{\text{turn}}^2 + 2 \cdot |dec_{\text{com}}| \cdot s_i} - v_{\text{turn}}}{|dec_{\text{com}}|}, & \text{if } s_{\text{dec}}(v_i, v_{\text{turn}}) \geq s_i \\ \frac{v_{\text{turn}} - v_i}{|dec_{\text{com}}|} + \frac{s_i - s_{\text{dec}}(v_i, v_{\text{turn}})}{v_i} & \text{otherwise} \end{cases} \quad (\text{D.4})$$

$$w_{\text{next}} = \begin{cases} \frac{\sqrt{v_{\text{turn}}^2 + 2 \cdot acc_{\text{com}} \cdot x_{\text{correction}} \cdot s_{\text{next}}} - v_{\text{turn}}}{acc_{\text{com}}} - \frac{x_{\text{correction}} \cdot s_{\text{next}}}{v_i}, & \text{if } s_{\text{acc}}(v_{\text{turn}}, v_i) \geq x_{\text{correction}} \cdot s_{\text{next}} \\ \frac{v_i - v_{\text{turn}}}{acc_{\text{com}}} - \frac{s_{\text{acc}}(v_{\text{turn}}, v_i)}{v_i} & \text{otherwise} \end{cases} \quad (\text{D.5})$$



Sensitivity Analysis Results of Operational Parameters

This appendix provides an overview of the results that have been gathered in the sensitivity analysis of the operational parameters. Four operational parameters have been selected, including the departure throughput $D_throughput$ and the maximum angle between two segments for which an aircraft does not have to slow down, No_turn . The weight of a segment is the ratio of its distance over the average taxi speed. To make sure the weight does not go up to infinity, a minimum fraction of the speed, Min_speed_frac , is used to determine a segment's weight. This latter variable is the minimum fraction of the speed that has to be taken into account and is selected as third operational parameter. The fourth factor, x_{coord} , is the scope of coordination between the ATC agents. This parameter is presented as *Coordination* in the upcoming tables.

Each parameter is tested for different values and simulated for both the 3rd and 7th of May, after which a statistical analysis is performed. All simulated scenarios are then compared with respect to the baseline scenario for both the individual days as well as the two days combined. Additionally, the combined set is analysed with respect to the actual operation, whose performance is extracted from ADS-B data. Table E.1 until Table E.4 present the results of this analysis for the performance indicators taxi time, taxi distance, average taxi speed and average density respectively. N represents the sample size, while P and r are the statistical significance and effect size respectively. This latter parameter is used to determine the practical relevance. P -values ≤ 0.05 and effect sizes $r \geq 0.3$ are shown in green. The symbol " - " indicates that no statistical results could be determined, as no change was observed.

Additionally, the results of all scenarios are used as input for a correlation analysis. The results are shown in Table E.5 until E.8 for taxi time, taxi distance, average taxi speed, and average density respectively.

Table E.1: Results sensitivity analysis of operational parameters on taxi time

	w.r.t. baseline								w.r.t. actual		
	03-05-2016			07-05-2016			Both		Both		
	N	P	r	N	P	r	P	r	N	P	r
Baseline	914	-	-	853	-	-	-	-	1766	0.000	0.28
D_throughput_-20%	914	0.000	0.25	853	0.000	0.33	0.000	0.28	1766	0.000	0.17
D_throughput_-10%	914	0.000	0.19	853	0.000	0.30	0.000	0.24	1766	0.000	0.23
D_throughput_+10%	914	0.000	0.23	853	0.000	0.30	0.000	0.26	1766	0.000	0.31
No_turn_-10deg	914	0.003	0.07	853	0.000	0.23	0.000	0.14	1766	0.000	0.25
No_turn_+10deg	914	0.000	0.18	853	0.000	0.26	0.000	0.21	1766	0.000	0.33
Min_speed_frac_-50%	914	0.886	0.00	853	0.317	0.02	0.806	0.00	1766	0.000	0.28
Min_speed_frac_+50%	914	0.178	0.03	853	-	-	0.178	0.02	1766	0.000	0.28
Coordination_0	914	0.857	0.00	853	0.417	0.02	0.689	0.01	1766	0.000	0.28
Coordination_1	914	0.647	0.01	853	0.294	0.03	0.722	0.01	1766	0.029	0.28
Coordination_3	914	0.000	0.05	853	0.893	0.00	0.035	0.04	1766	0.000	0.28

Table E.2: Results sensitivity analysis of operational parameters on taxi distance

	w.r.t. baseline								w.r.t. actual		
	03-05-2016			07-05-2016			Both		Both		
	<i>N</i>	<i>P</i>	<i>r</i>	<i>N</i>	<i>P</i>	<i>r</i>	<i>P</i>	<i>r</i>	<i>N</i>	<i>P</i>	<i>r</i>
Baseline	914	-	-	853	-	-	-	-	1766	0.000	0.07
D_throughput_-20%	914	0.310	0.02	853	0.308	0.02	0.377	0.01	1766	0.000	0.08
D_throughput_-10%	914	0.978	0.00	853	1.00	0.00	0.986	0.00	1766	0.000	0.07
D_throughput_+10%	914	0.886	0.00	853	-	-	0.886	0.00	1766	0.000	0.07
No_turn_-10deg	914	0.021	0.05	853	0.000	0.17	0.000	0.10	1766	0.000	0.13
No_turn_+10deg	914	0.002	0.14	853	0.000	0.07	0.000	0.12	1766	0.732	0.01
Min_speed_frac_-50%	914	0.520	0.02	853	0.317	0.02	0.783	0.00	1766	0.000	0.07
Min_speed_frac_+50%	914	0.068	0.04	853	-	-	0.068	0.03	1766	0.000	0.07
Coordination_0	914	0.624	0.01	853	0.009	0.06	0.450	0.01	1766	0.000	0.06
Coordination_1	914	0.114	0.04	853	0.110	0.04	0.044	0.03	1766	0.000	0.06
Coordination_3	914	0.020	0.05	853	0.109	0.04	0.014	0.04	1766	0.000	0.08

Table E.3: Results sensitivity analysis of operational parameters on average taxi speed

	w.r.t. baseline								w.r.t. actual		
	03-05-2016			07-05-2016			Both		Both		
	<i>N</i>	<i>P</i>	<i>r</i>	<i>N</i>	<i>P</i>	<i>r</i>	<i>P</i>	<i>r</i>	<i>N</i>	<i>P</i>	<i>r</i>
Baseline	914	-	-	853	-	-	-	-	1766	0.000	0.29
D_throughput_-20%	914	0.000	0.27	853	0.000	0.33	0.000	0.30	1766	0.000	0.20
D_throughput_-10%	914	0.000	0.21	853	0.000	0.3	0.000	0.25	1766	0.000	0.25
D_throughput_+10%	914	0.000	0.22	853	0.000	0.3	0.000	0.26	1766	0.000	0.32
No_turn_-10deg	914	0.000	0.14	853	0.000	0.28	0.000	0.20	1766	0.000	0.25
No_turn_+10deg	914	0.000	0.24	853	0.000	0.28	0.000	0.25	1766	0.000	0.36
Min_speed_frac_-50%	914	0.906	0.00	853	0.317	0.02	0.911	0.00	1766	0.000	0.29
Min_speed_frac_+50%	914	0.345	0.02	853	-	-	0.345	0.02	1766	0.000	0.29
Coordination_0	914	0.253	0.03	853	0.712	0.01	0.499	0.01	1766	0.000	0.29
Coordination_1	914	0.469	0.02	853	0.159	0.03	0.108	0.03	1766	0.000	0.29
Coordination_3	914	0.084	0.04	853	0.735	0.01	0.084	0.03	1766	0.000	0.29

Table E.4: Results sensitivity analysis of operational parameters on average density

	w.r.t. baseline								w.r.t. actual		
	03-05-2016			07-05-2016			Both		Both		
	<i>N</i>	<i>P</i>	<i>r</i>	<i>N</i>	<i>P</i>	<i>r</i>	<i>P</i>	<i>r</i>	<i>N</i>	<i>P</i>	<i>r</i>
Baseline	914	-	-	853	-	-	-	-	1766	0.000	0.50
D_throughput_-20%	914	0.000	0.34	853	0.000	0.4	0.000	0.46	1766	0.000	0.37
D_throughput_-10%	914	0.000	0.32	853	0.000	0.44	0.000	0.37	1766	0.000	0.45
D_throughput_+10%	914	0.000	0.36	853	0.000	0.43	0.000	0.39	1766	0.000	0.52
No_turn_-10deg	914	0.000	0.12	853	0.000	0.31	0.000	0.21	1766	0.000	0.47
No_turn_+10deg	914	0.000	0.31	853	0.000	0.39	0.000	0.34	1766	0.000	0.54
Min_speed_frac_-50%	914	0.662	0.01	853	0.500	0.02	0.748	0.01	1766	0.000	0.50
Min_speed_frac_+50%	914	0.005	0.07	853	-	-	0.005	0.05	1766	0.000	0.50
Coordination_0	914	0.996	0.00	853	0.238	0.03	0.458	0.01	1766	0.000	0.50
Coordination_1	914	0.024	0.05	853	0.855	0.00	0.062	0.03	1766	0.000	0.50
Coordination_3	914	0.000	0.12	853	0.215	0.03	0.000	0.08	1766	0.000	0.51

Table E.5: Results Spearman's Rank Correlation of operational parameters on taxi time

	03-05-2016			07-05-2016			Both	
	<i>N</i>	<i>Rho</i>	<i>P</i>	<i>N</i>	<i>Rho</i>	<i>P</i>	<i>Rho</i>	<i>P</i>
D_throughput	3656	-0.02	0.140	3412	-0.062	0.000	-0.04	0.001
No_turn	2742	-0.02	0.198	2559	-0.027	0.175	-0.03	0.068
Min_speed_frac	2742	0.00	0.964	2559	0.000	0.997	0.00	0.969
Coordination	3656	0.00	0.955	3412	-0.001	0.955	0.00	0.950

Table E.6: Results Spearman's Rank Correlation of operational parameters on taxi distance

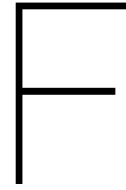
	03-05-2016			07-05-2016			Both	
	<i>N</i>	<i>Rho</i>	<i>P</i>	<i>N</i>	<i>Rho</i>	<i>P</i>	<i>Rho</i>	<i>P</i>
D_throughput	3656	0.00	0.969	3412	0.00	0.990	0.00	0.993
No_turn	2742	0.01	0.757	2559	0.01	0.754	0.01	0.667
Min_speed_frac	2742	0.00	0.971	2559	0.00	0.990	0.00	0.967
Coordination	3656	0.00	0.939	3412	-0.00	0.958	0.00	0.945

Table E.7: Results Spearman's Rank Correlation of operational parameters on average taxi speed

	03-05-2016			07-05-2016			Both	
	<i>N</i>	<i>Rho</i>	<i>P</i>	<i>N</i>	<i>Rho</i>	<i>P</i>	<i>Rho</i>	<i>P</i>
D_throughput	3656	0.06	0.000	3412	0.05	0.007	0.05	0.000
No_turn	2742	0.15	0.000	2559	0.09	0.000	0.12	0.000
Min_speed_frac	2742	0.00	0.962	2559	0.00	0.992	0.00	0.971
Coordination	3656	0.00	0.940	3412	0.00	0.983	0.00	0.959

Table E.8: Results Spearman's Rank Correlation of operational parameters on average density

	03-05-2016			07-05-2016			Both	
	<i>N</i>	<i>Rho</i>	<i>P</i>	<i>N</i>	<i>Rho</i>	<i>P</i>	<i>Rho</i>	<i>P</i>
D_throughput	3656	-0.04	0.009	3412	-0.07	0.000	-0.06	0.000
No_turn	2742	-0.02	0.192	2559	-0.04	0.074	-0.03	0.034
Min_speed_frac	2742	0.00	0.993	2559	0.00	1.000	0.00	0.995
Coordination	3656	-0.01	0.677	3412	0.00	0.987	0.00	0.748



Sensitivity Analysis Results of Aircraft Dynamics

This appendix provides an overview of the results that have been gathered in the sensitivity analysis of the aircraft dynamics. Five parameters have been selected, including the maximum speed v_{\max} an aircraft agent can travel, and the maximum turn speed v_{turn} that is allowed when making a turn. Acc_{com} is the comfort acceleration of an aircraft, while dec_{com} and dec_{\max} are the comfort and maximum deceleration respectively. Additionally, 8 different speed profiles are tested, in which a combination of above mentioned parameters is changed. More details regarding the different speed profiles is given in Table 6.6.

Each parameter is tested for different values and simulated for both the 3rd and 7th of May, after which a statistical analysis is performed. All simulated scenarios are then compared with respect to the baseline scenario for both the individual days as well as the two days combined. Additionally, the combined set is analysed with respect to the actual operation, whose performance is extracted from ADS-B data. Table E1 until Table E4 present the results of this analysis for the performance indicators taxi time, taxi distance, average taxi speed and average density respectively. N represents the sample size, while P and r are the statistical significance and effect size respectively. This latter parameter is used to determine the practical relevance. P -values ≤ 0.05 and effect sizes $r \geq 0.3$ are shown in green. The symbol "-" indicates that no statistical results could be determined, as no change was observed.

Additionally, the results of the scenarios in which an individual parameter is changed are used as input for a correlation analysis. The results are shown in Table E5 until E8 for taxi time, taxi distance, average taxi speed, and average density respectively.

Table E1: Results sensitivity analysis of aircraft dynamics on taxi time

	w.r.t. baseline								w.r.t. ADSB		
	03-05-2016			07-05-2016			Both		Both		
	<i>N</i>	<i>P</i>	<i>r</i>	<i>N</i>	<i>P</i>	<i>r</i>	<i>P</i>	<i>r</i>	<i>N</i>	<i>P</i>	<i>r</i>
Baseline	914	-	-	853	-	-	-	-	1766	0.000	0.28
Profile1	913	0.000	0.55	853	0.000	0.57	0.000	0.56	1765	0.000	0.16
Profile2	914	0.000	0.52	853	0.000	0.57	0.000	0.54	1766	0.000	0.52
Profile3	914	0.000	0.34	853	0.000	0.50	0.000	0.41	1766	0.000	0.16
Profile4	914	0.000	0.36	853	0.000	0.53	0.000	0.44	1766	0.000	0.37
Profile5	914	0.000	0.45	853	0.000	0.50	0.000	0.47	1766	0.015	0.04
Profile6	748	0.000	0.28	161	0.000	0.50	0.000	0.31	909	0.000	0.34
Profile7	913	0.000	0.50	853	0.000	0.54	0.000	0.52	1765	0.000	0.04
Profile8	914	0.000	0.43	853	0.000	0.53	0.000	0.47	1766	0.016	0.45
<i>v</i> _{max_-20%}	914	0.000	0.45	853	0.000	0.47	0.000	0.46	1766	0.000	0.06
<i>v</i> _{max_+20%}	914	0.000	0.33	853	0.000	0.36	0.000	0.34	1766	0.000	0.37
<i>v</i> _{turn_-20%}	904	0.000	0.38	853	0.000	0.52	0.000	0.44	1756	0.000	0.14
<i>v</i> _{turn_+20%}	914	0.000	0.38	853	0.000	0.52	0.000	0.45	1766	0.000	0.38
<i>acc</i> _{com_-20%}	914	0.000	0.35	853	0.000	0.50	0.000	0.42	1766	0.000	0.17
<i>acc</i> _{com_+20%}	914	0.000	0.35	853	0.000	0.54	0.000	0.44	1766	0.000	0.34
<i>dec</i> _{com_-20%}	914	0.000	0.30	853	0.000	0.43	0.000	0.36	1766	0.000	0.25
<i>dec</i> _{com_+20%}	488	0.000	0.32	853	0.000	0.42	0.000	0.38	1340	0.000	0.32
<i>dec</i> _{max_-20%}	914	0.005	0.07	853	0.754	0.01	0.008	0.04	1766	0.000	0.26
<i>dec</i> _{max_+20%}	914	0.228	0.03	853	0.034	0.05	0.039	0.03	1766	0.000	0.28

Table E2: Results sensitivity analysis of aircraft dynamics on taxi distance

	w.r.t. baseline								w.r.t. ADSB		
	03-05-2016			07-05-2016			Both		Both		
	<i>N</i>	<i>P</i>	<i>r</i>	<i>N</i>	<i>P</i>	<i>r</i>	<i>P</i>	<i>r</i>	<i>N</i>	<i>P</i>	<i>r</i>
Baseline	914	-	-	853	-	-	-	-	1766	0.000	0.07
Profile1	913	0.122	0.04	853	0.001	0.08	0.775	0.00	1765	0.000	0.08
Profile2	914	0.688	0.01	853	0.014	0.06	0.119	0.03	1766	0.000	0.07
Profile3	914	0.223	0.03	853	0.000	0.13	0.000	0.07	1766	0.090	0.03
Profile4	914	0.058	0.04	853	0.000	0.13	0.000	0.07	1766	0.000	0.11
Profile5	914	0.016	0.06	853	0.000	0.20	0.000	0.11	1766	0.000	0.14
Profile6	748	0.004	0.07	161	0.001	0.19	0.000	0.08	909	0.174	0.03
Profile7	913	0.004	0.07	853	0.000	0.20	0.000	0.13	1765	0.000	0.14
Profile8	914	0.882	0.00	853	0.000	0.15	0.000	0.06	1766	0.000	0.10
<i>v</i> _{max_-20%}	914	0.009	0.06	853	0.000	0.17	0.000	0.11	1766	0.000	0.13
<i>v</i> _{max_+20%}	914	0.051	0.05	853	0.000	0.14	0.000	0.09	1766	0.701	0.01
<i>v</i> _{turn_-20%}	904	0.097	0.04	853	0.005	0.07	0.006	0.05	1756	0.058	0.03
<i>v</i> _{turn_+20%}	914	0.024	0.05	853	0.629	0.01	0.028	0.04	1766	0.000	0.08
<i>acc</i> _{com_-20%}	914	0.567	0.01	853	0.000	0.11	0.002	0.05	1766	0.010	0.04
<i>acc</i> _{com_+20%}	914	0.076	0.04	853	0.000	0.15	0.000	0.07	1766	0.000	0.11
<i>dec</i> _{com_-20%}	914	0.714	0.01	853	0.486	0.02	0.978	0.00	1766	0.000	0.07
<i>dec</i> _{com_+20%}	488	0.691	0.01	853	0.500	0.02	0.961	0.00	1340	0.000	0.09
<i>dec</i> _{max_-20%}	914	0.409	0.02	853	0.446	0.02	0.278	0.02	1766	0.000	0.06
<i>dec</i> _{max_+20%}	914	0.005	0.07	853	0.301	0.03	0.004	0.05	1766	0.000	0.07

Table E3: Results sensitivity analysis of aircraft dynamics on average taxi speed

	w.r.t. baseline								w.r.t. ADSB		
	03-05-2016			07-05-2016			Both		Both		
	<i>N</i>	<i>P</i>	<i>r</i>	<i>N</i>	<i>P</i>	<i>r</i>	<i>P</i>	<i>r</i>	<i>N</i>	<i>P</i>	<i>r</i>
Baseline	914	-	-	853	-	-	-	-	1766	0.000	0.29
Profile1	913	0.000	0.57	853	<0.05	0.59	0.000	0.58	1765	0.000	0.17
Profile2	914	0.000	0.55	853	0.000	0.58	0.000	0.56	1766	0.000	0.53
Profile3	914	0.000	0.40	853	0.000	0.49	0.000	0.44	1766	0.000	0.17
Profile4	914	0.000	0.37	853	0.000	0.55	0.000	0.45	1766	0.000	0.38
Profile5	914	0.000	0.51	853	0.000	0.52	0.000	0.52	1766	0.035	0.04
Profile6	748	0.000	0.32	161	0.000	0.53	0.000	0.35	909	0.000	0.38
Profile7	913	0.000	0.54	853	0.000	0.56	0.000	0.55	1765	0.002	0.05
Profile8	914	0.000	0.49	853	0.000	0.54	0.000	0.41	1766	0.000	0.47
$v_{\max_ -20\%}$	914	0.000	0.48	853	0.000	0.49	0.000	0.49	1766	0.000	0.08
$v_{\max_ +20\%}$	914	0.000	0.36	853	0.000	0.40	0.000	0.38	1766	0.000	0.38
$v_{\text{turn}_ -20\%}$	904	0.000	0.43	853	0.000	0.54	0.000	0.48	1756	0.000	0.14
$v_{\text{turn}_ +20\%}$	914	0.000	0.41	853	0.000	0.54	0.000	0.47	1766	0.000	0.40
$acc_{\text{com}_ -20\%}$	914	0.000	0.39	853	0.000	0.48	0.000	0.44	1766	0.000	0.18
$acc_{\text{com}_ +20\%}$	914	0.000	0.35	853	0.000	0.55	0.000	0.44	1766	0.000	0.36
$dec_{\text{com}_ -20\%}$	914	0.000	0.34	853	0.000	0.45	0.000	0.39	1766	0.000	0.26
$dec_{\text{com}_ +20\%}$	488	0.000	0.32	853	0.000	0.45	0.000	0.40	1340	0.000	0.33
$dec_{\text{max}_ -20\%}$	914	0.001	0.07	853	0.514	0.02	0.017	0.04	1766	0.000	0.28
$dec_{\text{max}_ +20\%}$	914	0.287	0.02	853	0.023	0.05	0.675	0.01	1766	0.000	0.29

Table E4: Results sensitivity analysis of aircraft dynamics on average density

	w.r.t. baseline								w.r.t. ADSB		
	03-05-2016			07-05-2016			Both		Both		
	<i>N</i>	<i>P</i>	<i>r</i>	<i>N</i>	<i>P</i>	<i>r</i>	<i>P</i>	<i>r</i>	<i>N</i>	<i>P</i>	<i>r</i>
Baseline	914	-	-	853	-	-	-	-	1766	0.000	0.50
Profile1	913	0.000	0.60	853	0.000	0.61	0.000	0.60	1765	0.000	0.21
Profile2	914	0.000	0.59	853	0.000	0.60	0.000	0.60	1766	0.000	0.61
Profile3	914	0.000	0.40	853	0.000	0.57	0.000	0.48	1766	0.000	0.34
Profile4	914	0.000	0.41	853	0.000	0.55	0.000	0.47	1766	0.000	0.56
Profile5	914	0.000	0.53	853	0.000	0.58	0.000	0.55	1766	0.000	0.18
Profile6	748	0.000	0.35	161	0.000	0.53	0.000	0.37	909	0.000	0.55
Profile7	913	0.000	0.58	853	0.000	0.60	0.000	0.59	1765	0.558	0.01
Profile8	914	0.000	0.53	853	0.000	0.58	0.000	0.56	1766	0.000	0.59
$v_{\max_ -20\%}$	914	0.000	0.56	853	0.000	0.57	0.000	0.56	1766	0.000	0.19
$v_{\max_ +20\%}$	914	0.000	0.42	853	0.000	0.48	0.000	0.45	1766	0.000	0.57
$v_{\text{turn}_ -20\%}$	904	0.000	0.48	853	0.000	0.56	0.000	0.51	1756	0.000	0.18
$v_{\text{turn}_ +20\%}$	914	0.000	0.45	853	0.000	0.53	0.000	0.49	1766	0.000	0.57
$acc_{\text{com}_ -20\%}$	914	0.000	0.41	853	0.000	0.56	0.000	0.48	1766	0.000	0.36
$acc_{\text{com}_ +20\%}$	914	0.000	0.39	853	0.000	0.53	0.000	0.45	1766	0.000	0.55
$dec_{\text{com}_ -20\%}$	914	0.000	0.22	853	0.000	0.39	0.000	0.29	1766	0.000	0.46
$dec_{\text{com}_ +20\%}$	488	0.000	0.21	853	0.000	0.34	0.000	0.28	1340	0.000	0.54
$dec_{\text{max}_ -20\%}$	914	0.000	0.09	853	0.581	0.01	0.000	0.06	1766	0.000	0.47
$dec_{\text{max}_ +20\%}$	914	0.110	0.04	853	0.273	0.03	0.354	0.02	1766	0.000	0.51

Table F5: Results Spearman's Rank Correlation of aircraft dynamics on taxi time

	03-05-2016			07-05-2016			Both	
	<i>N</i>	<i>Rho</i>	<i>P</i>	<i>N</i>	<i>Rho</i>	<i>P</i>	<i>Rho</i>	<i>P</i>
v_{\max}	2742	-0.07	0.000	2559	-0.08	0.000	-0.07	0.000
v_{turn}	2732	-0.06	0.002	2559	-0.08	0.000	-0.07	0.000
acc_{com}	2742	-0.05	0.016	2559	-0.07	0.000	-0.05	0.000
dec_{com}	2316	0.02	0.332	2559	-0.01	0.45	0.01	0.588
dec_{\max}	2742	0.00	0.829	2559	0.00	0.85	0.00	0.797

Table F6: Results Spearman's Rank Correlation of aircraft dynamics on taxi distance

	03-05-2016			07-05-2016			Both	
	<i>N</i>	<i>Rho</i>	<i>P</i>	<i>N</i>	<i>Rho</i>	<i>P</i>	<i>Rho</i>	<i>P</i>
v_{\max}	2742	0.00	0.989	2559	0.00	0.827	0.00	0.891
v_{turn}	2732	0.00	0.994	2559	0.00	0.892	0.00	0.919
acc_{com}	2742	0.00	0.877	2559	-0.01	0.793	0.00	0.768
dec_{com}	2316	0.04	0.069	2559	0.00	0.981	0.02	0.174
dec_{\max}	2742	0.00	0.898	2559	0.00	0.972	0.00	0.930

Table F7: Results Spearman's Rank Correlation of aircraft dynamics on average taxi speed

	03-05-2016			07-05-2016			Both	
	<i>N</i>	<i>Rho</i>	<i>P</i>	<i>N</i>	<i>Rho</i>	<i>P</i>	<i>Rho</i>	<i>P</i>
v_{\max}	2742	0.30	0.000	2559	0.26	0.000	0.27	0.000
v_{turn}	2732	0.27	0.000	2559	0.23	0.000	0.24	0.000
acc_{com}	2742	0.16	0.000	2559	0.16	0.000	0.15	0.000
dec_{com}	2316	0.08	0.000	2559	0.05	0.010	0.06	0.000
dec_{\max}	2742	0.01	0.694	2559	0.00	0.824	0.00	0.720

Table F8: Results Spearman's Rank Correlation of aircraft dynamics on average density

	03-05-2016			07-05-2016			Both	
	<i>N</i>	<i>Rho</i>	<i>P</i>	<i>N</i>	<i>Rho</i>	<i>P</i>	<i>Rho</i>	<i>P</i>
v_{\max}	2742	-0.13	0.000	2559	-0.14	0.000	-0.13	0.000
v_{turn}	2732	-0.18	0.000	2559	-0.11	0.000	-0.14	0.000
acc_{com}	2742	-0.06	0.000	2559	-0.09	0.000	-0.07	0.000
dec_{com}	2316	0.07	0.001	2559	-0.01	0.498	0.02	0.283
dec_{\max}	2742	-0.02	0.419	2559	0.00	0.900	-0.01	0.503



Statistical Hypothesis Testing

G.1. Baseline Scenario

This section lists the statistical hypotheses that have been tested to compare the baseline scenario of the simulated operation with respect to the actual operation.

Main hypothesis

H_0 The simulated operation matches the actual operation

H_1 The simulated operation does not match the actual operation

This hypothesis is split in four other hypotheses, for each of the key performance indicators.

Hypothesis taxi time:

H_0 The simulated operation matches the actual operation in terms of taxi time

H_1 The simulated operation does not matches the actual operation in terms of taxi time

A Wilcoxon test was conducted to evaluate whether the simulated operation shows similar taxi time compared to the actual operation. The results indicate a significant difference, $z= 47.65$, $P < .05$, $r= 0.31$. The median of the ranks in favour of simulated operation is 4061.5, while the mean of the ranks in favour of the actual operation is 6832. The null hypothesis is thereby rejected. The results show a practical relevance that is medium.

Hypothesis taxi distance:

H_0 The simulated operation matches the actual operation in terms of taxi distance

H_1 The simulated operation does not matches the actual operation in terms of taxi distance

A Wilcoxon test was conducted to evaluate whether the simulated operation shows similar taxi distance compared to the actual operation. The results indicate a significant difference, $z= -29.85$, $P < .05$, $r= 0.19$. The median of the ranks in favour of simulated operation is 3838, while the mean of the ranks in favour of the actual operation is 4442. The null hypothesis is thereby rejected. The results show a practical relevance that is trivial.

Hypothesis average taxi speed:

H_0 The simulated operation matches the actual operation in terms of average taxi speed

H_1 The simulated operation does not matches the actual operation in terms of average taxi speed

A Wilcoxon test was conducted to evaluate whether the simulated operation shows similar average taxi speed compared to the actual operation. The results indicate a significant difference, $z= -47.97$, $P < .05$, $r= 0.31$. The median of the ranks in favour of simulated operation is 6981, while the mean of the ranks in favour of the actual operation is 4167. The null hypothesis is thereby rejected. The results show a practical relevance that is medium.

Hypothesis average density: **H_0** The simulated operation matches the actual operation in terms of average density **H_1** The simulated operation does not matches the actual operation in terms of average density

A Wilcoxon test was conducted to evaluate whether the simulated operation shows similar average density compared to the actual operation. The results indicate a significant difference, $z = -78.41$, $P < .05$, $r = 0.51$. The median of the ranks in favour of simulated operation is 2505.5, while the mean of the ranks in favour of the actual operation is 6561. The null hypothesis is thereby rejected. The results show a practical relevance that is large.

Since each of the four sub hypotheses is rejected, also the main hypothesis is rejected, it is found that the simulated and actual operation are significantly different from each other.

G.2. Sensitivity Analysis

This section is divided in two parts. First of all, the sensitivity analysis of the operational parameters is covered, followed by the aircraft dynamics.

Operational Parameters

Different scenarios in which individual operational parameters are changed, have been tested with respect to the baseline scenario. It should be noted that a Bonferroni correction factor of 5 is applied, as these five hypothesis have only be defined after obtaining the results. Therefore, α is set to $\alpha/5 = 0.01$. The results of the sensitivity analysis are presented in E. Although many hypotheses could be created, it is decided to focus on a smaller subset.

As aircraft spend only limited time in the runway queue, it is expected that the departure throughput $D_throughput$ has no impact on the taxi time.

Sensitivity departure throughput: **H_0** A small decrease in departure throughput does not affect taxi time **H_1** A small decrease in departure throughput does affect taxi time

A Wilcoxon test was conducted to evaluate whether the scenario with a 10% decrease in departure throughput shows similar taxi time compared to the baseline scenario. The results indicate a significant difference, $z = -14.33$, $P < .01$, $r = 0.24$. The median of the ranks in favour of new scenario is 263.5, while the mean of the ranks in favour of the baseline scenario is 428.5. The null hypothesis is thereby rejected. The results show a practical relevance that is trivial.

Aircraft taxi most part of their route along straight segments. Also, most corners at the layout of AAS are perpendicular, which is why it is expected that the parameter no_turn has no impact on the taxi time.

Sensitivity turn angle: **H_0** A small decrease in turn angle does not affect taxi speed **H_1** A small decrease in turn angle does affect taxi speed

A Wilcoxon test was conducted to evaluate whether the scenario with a 10° decrease in turn angle shows similar taxi speed compared to the baseline scenario. The results indicate a significant difference, $z = -12.06$, $P < .01$, $r = 0.20$. The median of the ranks in favour of new scenario is 491.5, while the mean of the ranks in favour of the baseline scenario is 478. The null hypothesis is thereby rejected. The results show a practical relevance that is trivial.

The parameter min_speed_frac is implemented in the model to prevent a segment's weight to increase infinity, causing aircraft to make detours to avoid this segment. It is assumed that the implementation of this parameter is sufficient to encourage aircraft to be guided along this segment. Therefore, no large detours are required, which is why it is expected that the parameter has no effect on taxi distance.

Sensitivity minimum speed fraction: **H_0** An increase in minimum speed fraction does not affect taxi distance **H_1** An increase in minimum speed fraction does affect taxi distance

A Wilcoxon test was conducted to evaluate whether the scenario with a 50% increase in minimum speed fraction shows similar taxi distance compared to the baseline scenario. The results indicate a significant difference, $z = -1.83$, $P = 0.07$, $r = 0.03$. The median of the ranks in favour of new scenario is -, while the mean of the ranks in favour of the baseline scenario is 2.5. The null hypothesis is thereby accepted. The results show a practical relevance that is trivial.

It is mentioned by Udluft that scope of coordination only benefits an operation that is congested [73]. It is therefore expected that increasing the scope of coordination has no effect on the taxi time or taxi distance.

Sensitivity scope of coordination: **H_0** A small increase in scope of coordination does not affect taxi time **H_1** A small increase in scope of coordination does affect taxi time

A Wilcoxon test was conducted to evaluate whether the scenario with a +1 increase in scope of coordination shows similar taxi time compared to the baseline scenario. The results indicate a significant difference, $z = -2.11$, $P = 0.03$, $r = 0.04$. The median of the ranks in favour of new scenario is 59.5, while the mean of the ranks in favour of the baseline scenario is 69.5. The null hypothesis is thereby accepted. The results show a practical relevance that is less than trivial.

Sensitivity scope of coordination: **H_0** A small increase in scope of coordination does not affect taxi distance **H_1** A small increase in scope of coordination does affect taxi distance

A Wilcoxon test was conducted to evaluate whether the scenario with a +1 increase in scope of coordination shows similar taxi distance compared to the baseline scenario. The results indicate a significant difference, $z = -2.45$, $P = 0.01$, $r = 0.04$. The median of the ranks in favour of new scenario is 38.5, while the mean of the ranks in favour of the baseline scenario is 49.5. The null hypothesis is thereby accepted. The results show a practical relevance that is less than trivial.

Aircraft Dynamics

Two examples are given regarding the hypotheses that have been tested for the aircraft dynamics. The scenario in which the aircraft dynamics are decreased by 20% is tested with respect to the baseline scenario, and the actual operation.

As the taxi speed of an aircraft is influenced by stop commands from ATC and nearby congestion, it is expected that changing the maximum speeds of an aircraft has limited to no impact on the taxi time of an aircraft.

Sensitivity aircraft dynamics: **H_0** A decrease in aircraft speed does not affect taxi time **H_1** A decrease in aircraft speed does affect taxi time

A Wilcoxon test was conducted to evaluate whether the scenario with a 20% decrease in maximum speed and maximum turn speed, shows similar taxi time compared to the baseline scenario. The results indicate a significant difference, $z = -32.59$, $P < 0.05$, $r = 0.55$. The median of the ranks in favour of new scenario is 334, while the mean of the ranks in favour of the baseline scenario is 906. The null hypothesis is thereby rejected. The results show a practical relevance that is large.

Additionally, a Wilcoxon test was conducted to evaluate whether the scenario with a 20% decrease in maximum speed and maximum turn speed, shows similar taxi time compared to the actual operation. The results indicate a significant difference, $z = -3.12$, $P < 0.05$, $r = 0.05$. The median of the ranks in favour of new scenario is 836, while the mean of the ranks in favour of the baseline scenario is 915.5. The null hypothesis is thereby rejected. The results show a practical relevance that is less than trivial.