TU Delft

Delft University of Technology

Document Version
Final published version

Citation (APA)
von der Burg, M. F. (2026). *Towards Automated and Sustainable Airport Surface Movement Operations: Designing Models, Methods, and Tools for Next-generation Concepts of Operations*. [Dissertation (TU Delft), Delft University of Technology]. https://doi.org/10.4233/uuid:d0a3847a-e76f-4737-a1b6-bd0b713fe3ce

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

Copyright
In case the licence states “Dutch Copyright Act (Article 25fa)”, this publication was made available Green Open Access via the TU Delft Institutional Repository pursuant to Dutch Copyright Act (Article 25fa, the Taverne amendment). This provision does not affect copyright ownership.
Unless copyright is transferred by contract or statute, it remains with the copyright holder.

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

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

This work is downloaded from Delft University of Technology.

TOWARDS AUTOMATED AND SUSTAINABLE **AIRPORT SURFACE MOVEMENT OPERATIONS**

Designing Models, Methods, and Tools for Next-generation Concepts of Operations



Malte Frederic von der Burg

Towards Automated and Sustainable Airport Surface Movement Operations

Designing Models, Methods, and Tools for Next-generation Concepts of Operations

Dissertation

for the purpose of obtaining the degree of doctor
at Delft University of Technology
by the authority of the Rector Magnificus,
Prof.dr.ir. H. Bijl,
chair of the Board for Doctorates
to be defended publicly on
Wednesday, 3 June 2026 at 15:00

by

Malte Frederic VON DER BURG

This dissertation has been approved by the (co)promotors.

Composition of the doctoral committee:

Rector Magnificus,	chairperson
Prof.dr.ir. M. Mulder,	Delft University of Technology, *promotor*
Dr.ir. C. Borst,	Delft University of Technology, *promotor*
Dr. O.A. Sharpans'kykh,	Delft University of Technology, *copromotor*
Independent members:	
Prof.dr.ir. J.C.F. de Winter,	Delft University of Technology
Dr. B. Shyrokau,	Delft University of Technology
Prof. Dr.-Ing. M. Schultz,	Universität der Bundeswehr München, Germany
Dr. J. Chen,	Queen Mary University of London, United Kingdom
Reserve member:	
Prof.dr.ir. R.C. Alderliesten,	Delft University of Technology







This research was part of the two projects 'AEON – Advanced Engine Off Navigation' and 'ASTAIR – Auto Steer Taxi at AIRport' which have received funding from SESAR 3 JU under grant agreement No. 892869 (AEON) and No. 101114684 (ASTAIR).

Keywords: Airport Surface Movement Operations, Multi-Agent System, Multi-Agent Motion Planning, Hierarchical-Distributed Control, Concept of Operations, Engine-Off Taxiing, Sustainable Aviation, Automation, Human–Automation Interactions

Printed by: Ipskamp Printing

Cover by: S. Frühwirt

Copyright © 2026 by M.F. von der Burg

ISBN 978-94-6384-962-3

An electronic copy of this dissertation is available at: https://repository.tudelft.nl/

Summary

Airside operations at major airports are under increasing pressure as traffic demand grows, environmental constraints tighten, and expectations for safety remain uncompromised. On the airport surface, increasing congestion, uncertain taxiing behaviour, and rising operational complexity place additional demands on Air Traffic Control Officers (ATCOs), intensifying the risk of inefficiencies, elevated workload, and human error. In parallel, the transition towards more sustainable ground operations, such as the emergence of engine-off taxiing techniques, require a fundamental rethinking of the existing airport operations.

Against this backdrop, regulators, researchers, and practitioners increasingly view AI-enabled decision-support and automation as key enablers for future improvements. Yet, the literature lacks a modelling framework that simultaneously captures the interrelated operational processes, heterogeneous actors, and fidelity requirements necessary to evaluate next-generation concepts for airport surface movement operations (ASM Ops) in a realistic and systematic manner. Many existing models remain limited to isolated subsystems, single concepts, or strong simplifying assumptions, which restrict their ability to compare alternative future approaches at a detailed level. To address this gap, the objective of this thesis is to design models, methods, and tools to investigate next-generation concepts of operations (ConOps) for automated and sustainable ASM Ops.

The thesis is structured around eight research questions (RQs) and proceeds in three phases. The **development phase** (Chapters 2 to 5) first derives modelling requirements from the operational structure of ASM Ops and from the goal to represent surface movements at high fidelity while remaining computationally tractable (RQ 1). ASM Ops encompasses a hierarchy of tasks, ranging from strategic planning (long-term and high-level decisions) to scheduling (tactical allocation of resources), routing (tactical planning of movements), guidance (operational planning of movements), and finally movement execution (controlling aircraft and ground vehicles). This hierarchical decomposition, combined with the need for coordinated oversight to maintain the strict safety standards in aviation, motivates a hierarchical–distributed modelling approach.

To provide the required modularity, adaptability, and expressiveness in both fast-time and real-time simulations, this thesis adopts the multi-agent systems (MAS) paradigm. A generalised MAS architecture is proposed, comprising environmental objects and four agent categories aligned with the hierarchy of ASM Ops tasks (RQ 2). The architecture supports different allocations of responsibilities and different coordination mechanisms, enabling the instantiation of tailored model instances for specific ConOps while retaining a reusable simulation environment and component structure.

To support trajectory-based automation studies at airport scale (RQ 3), the thesis then outlines the Multi-Agent Motion Planning on Airport Surfaces (AS-MAMP) algorithm as the decision-logic for centralised path planning in a fully-automated operational setting, i.e. on EASA's AI Level 3. AS-MAMP is a two-level solver that builds on Priority-Based Search

(PBS) and its variants for high-level conflict resolution. Because existing low-level planners were insufficient to compute realistic 4D ground trajectories under operational constraints, the thesis introduces the novel Safe Interval Motion Planning (SIMP) algorithm. SIMP plans continuous-time trajectories in accordance with the operational processes in ASM Ops (e.g. pushback and engine-start, tug coupling/decoupling, and holding), motions based on finite acceleration, and conflict avoidance in continuous space and time.

The resulting planning approach is evaluated through structured benchmarking on a synthetic airport layout, where the high-level coordination component is compared against PBS variants and SIMP is benchmarked against SIPP and kinodynamic A*. The evaluation is complemented by experiments on the real-world layout of Amsterdam Airport Schiphol to assess scalability and operational relevance under realistic airport complexity. Taken together, these results establish the MAS architecture and AS-MAMP planning capability as an enabling foundation for subsequent operational analyses, while also identifying limitations and implementation challenges that arise when moving towards real-world deployment.

The **validation phase** (Chapters 6 to 8) applies the developed MAS framework in operational studies and subsequently strengthens its realism. Two far-term analyses investigate fully-automated ASM Ops under multi-engine taxiing (RQ 4) and under engine-off taxiing (RQ 5), examining system-level implications under dense traffic assumptions. Building on feedback from ATCOs and other operational experts gathered throughout the thesis, historical aircraft ground tracks are then analysed to improve the calibration of key movement parameters (RQ 6) so that the model is able to more accurately reproduce the historical operations. The MAS model is further extended to represent aircraft towing movements and to assess their impact on regular traffic (RQ 7). Based on these modifications, multiple operational model variations are compared to clarify the sensitivity of results to modelling assumptions and to discuss implications for next-generation surface operations.

Fully-automated operations pose significant implementation challenges in real-world settings, so human involvement remains required for the foreseeable future. Therefore, as a final **exploration phase**, Chapter 9 provides an outlook towards EASA's AI Level 2 concepts by examining how human–automation interaction mechanisms can be embedded into the MAS model to enable operator involvement (RQ 8). These demonstrations focus on technical feasibility – illustrating how interfaces and information exchange could be embedded into the MAS model – rather than validated evidence from human-in-the-loop experiments. Chapter 10 concludes by synthesising the findings across all research questions, reflecting on limitations and implications for future ASM Ops, and outlining directions for continued development and socio-technical validation.

Samenvatting

Airside-operaties op grote luchthavens staan onder toenemende druk door de groeiende verkeersvraag, strengere milieueisen en onverminderd hoge veiligheidseisen. Op het luchthavenoppervlak leiden toenemende congestie, onzeker taxi-gedrag en stijgende operationele complexiteit tot aanvullende eisen aan luchtverkeersleiders (ATCO's), waardoor het risico op inefficiënties, verhoogde werkdruk en menselijke fouten toeneemt. Tegelijkertijd vereist de transitie naar duurzamere grondoperaties, zoals de opkomst van taxi-technieken met uitgeschakelde motoren, een fundamentele heroverweging van de bestaande luchthavenoperaties.

Tegen deze achtergrond beschouwen regelgevende instanties, onderzoekers en praktijkexperts AI-ondersteunde besluitvorming en automatisering in toenemende mate als belangrijke enablers voor toekomstige verbeteringen. Desondanks ontbreekt in de literatuur een modelleringsraamwerk dat gelijktijdig de onderling verweven operationele processen, heterogene actoren en vereisten aan modelgetrouwheid kan vastleggen die nodig zijn om toekomstige operationele concepten voor airport surface movement operations (ASM Ops) op realistische en systematische wijze te evalueren. Veel bestaande modellen blijven beperkt tot geïsoleerde deelsystemen, afzonderlijke concepten of sterk vereenvoudigde aannames, waardoor hun vermogen om alternatieve toekomstige benaderingen op gedetailleerd niveau te vergelijken beperkt is. Om deze leemte te adresseren, heeft dit proefschrift als doel modellen, methoden en tools te ontwerpen om toekomstige concepts of operations (ConOps) voor geautomatiseerde en duurzame ASM Ops te onderzoeken.

Het proefschrift is opgebouwd rond acht onderzoeksvragen (RQs) en bestaat uit drie fasen. De **ontwikkelingsfase** (Hoofdstukken 2 tot en met 5) leidt eerst modelleringsvereisten af uit de operationele structuur van ASM Ops en uit de noodzaak om grondbewegingen detailgetrouw, maar computationeel hanteerbaar te representeren (RQ 1). ASM Ops omvat een hiërarchie van taken: van strategische planning (langetermijnbeslissingen), scheduling (tactische middelenallocatie), routing (tactische bewegingsplanning) en guidance (operationele bewegingsplanning), tot de uitvoering van bewegingen zelf (het besturen van vliegtuigen en grondvoertuigen). Deze hiërarchische decompositie, samen met de noodzaak van gecoördineerd toezicht om de strikte veiligheidsstandaarden in de luchtvaart te waarborgen, motiveert een hiërarchisch gedistribueerde modelleringsaanpak.

Om de vereiste modulariteit, aanpasbaarheid en expressiviteit in zowel fast-time als real-time simulaties te bieden, hanteert dit proefschrift het multi-agent systems (MAS) paradigma. Er wordt een gegeneraliseerde MAS-architectuur voorgesteld, bestaande uit omgevingsobjecten en vier agentcategorieën die aansluiten op de hiërarchie van ASM Ops taken (RQ 2). De architectuur ondersteunt verschillende verdelingen van verantwoordelijkheden en verschillende coördinatiemechanismen, waardoor op maat gemaakte modelinstanties voor specifieke ConOps kunnen worden geïnstantieerd, terwijl een herbruikbare simulatieomgeving en componentstructuur behouden blijven.

Ter ondersteuning van trajectgebaseerde automatiseringsstudies op luchthavenschaal (RQ 3) beschrijft het proefschrift vervolgens het Multi-Agent Motion Planning on Airport Surfaces (AS-MAMP) algoritme als beslislogica voor gecentraliseerde padplanning in een volledig geautomatiseerde operationele setting, dus op EASA's AI Level 3. AS-MAMP is een tweelaags oplossingsalgoritme dat voortbouwt op Priority-Based Search (PBS) en varianten daarvan voor de high-level conflictoplossing. Omdat bestaande low-level planners ontoereikend waren om realistische 4D-grondtrajecten onder operationele randvoorwaarden te berekenen, introduceert het proefschrift het nieuwe Safe Interval Motion Planning (SIMP) algoritme. SIMP plant continue-tijdstrajecten in overeenstemming met de operationele processen binnen ASM Ops (zoals pushback en engine-start, het aankoppelen en loskoppelen van sleepvoertuigen, en holding), bewegingen op basis van eindige versnelling, en conflictvermijding in continue ruimte en tijd.

De resulterende planningsaanpak wordt geëvalueerd aan de hand van gestructureerde benchmarktests op een synthetische luchthavenlayout, waarbij de high-level coördinatiecomponent wordt vergeleken met PBS-varianten, en SIMP wordt gebenchmarkt tegen SIPP en kinodynamic A*. De evaluatie wordt aangevuld met experimenten op de werkelijke layout van Amsterdam Airport Schiphol om schaalbaarheid en operationele relevantie onder realistische luchthavencomplexiteit te beoordelen. Gezamenlijk positioneren deze resultaten de MAS-architectuur en AS-MAMP planningscapaciteit als een faciliterende basis voor daaropvolgende operationele analyses, terwijl zij ook beperkingen en implementatie-uitdagingen blootleggen die ontstaan bij de stap richting toepassing in de operationele praktijk.

De **validatiefase** (Hoofdstukken 6 tot en met 8) past het ontwikkelde MAS-raamwerk toe in operationele studies en vergroot vervolgens de realiteitswaarde ervan. Twee langetermijnanalyses onderzoeken volledig geautomatiseerde ASM Ops onder multi-engine taxiing (RQ 4) en engine-off taxiing (RQ 5), waarbij systeemimplicaties bij hoge verkeersdichtheid worden bestudeerd. Op basis van feedback van ATCO's en andere operationele experts worden vervolgens historische grondtrajecten van vliegtuigen geanalyseerd om belangrijke bewegingsparameters te kalibreren (RQ 6), zodat het model historische operaties nauwkeuriger kan reproduceren. Daarnaast wordt het MAS-model uitgebreid om sleepbewegingen van vliegtuigen te representeren en hun impact op regulier verkeer te beoordelen (RQ 7). Op basis hiervan worden meerdere operationele modelvarianten vergeleken om de gevoeligheid van resultaten voor modelleringsaannames te verduidelijken en implicaties voor toekomstige grondbewegingen op luchthavens te bespreken.

Volledig geautomatiseerde operaties brengen aanzienlijke implementatie-uitdagingen met zich mee in werkelijke omgevingen, waardoor menselijke betrokkenheid in de voorzienbare toekomst noodzakelijk blijft. Daarom biedt Hoofdstuk 9, als afsluitende **exploratiefase**, een vooruitblik op EASA's AI Level 2 concepten door te onderzoeken hoe mechanismen voor mens-automatiseringsinteractie in het MAS-model kunnen worden ingebed om betrokkenheid van de operator mogelijk te maken (RQ 8). Deze demonstraties richten zich op technische haalbaarheid – door te illustreren hoe interfaces en informatie-uitwisseling in het MAS-model kunnen worden geïntegreerd – en niet op gevalideerd bewijs uit human-in-the-loop-experimenten. Hoofdstuk 10 sluit af met een synthese van de bevindingen over alle onderzoeksvragen heen, reflecteert op beperkingen en implicaties voor toekomstige ASM Ops, en schetst richtingen voor verdere ontwikkeling en socio-technische validatie.

Acknowledgements

Digital version intentionally left blank.

Contents

Summary iii

Samenvatting v

Acknowledgements vii

1 Introduction 1
- 1.1 Related European Research Projects . . . 7
- 1.2 Research Focus and Research Questions . . . 8
- 1.3 Research Scope and Assumptions . . . 12
- 1.4 Thesis Overview . . . 15

2 Problem Description and Related Work 19
- 2.1 Operational Background . . . 21
- 2.2 Modelling Requirements . . . 22
- 2.3 Related Work . . . 24
  - 2.3.1 Modelling of Airport Surface Movement Operations . . . 24
  - 2.3.2 Studying New Concepts of Operations . . . 25
  - 2.3.3 Multi-Agent Path Planning Algorithms for Conflict-free Routing . . . 25
- 2.4 Addressing the Research Gaps . . . 28

3 MAS Models to Study Next-Generation ASM Ops 31
- 3.1 Generalised Hierarchical-Distributed Control Architecture . . . 33
- 3.2 MAS Model for Fully-Automated Operations with Centralised Planning . . . 36
  - 3.2.1 Specification of Environment . . . 37
  - 3.2.2 Specification of Airport Ops Agent . . . 38
  - 3.2.3 Specification of Routing Agent . . . 39
  - 3.2.4 Specification of Guidance Agents . . . 40
  - 3.2.5 Specification of Moving Agents . . . 41

4 The AS-MAMP Algorithm 43
- 4.1 High-level Search in AS-MAMP . . . 45
- 4.2 Low-level Search in AS-MAMP . . . 47
- 4.3 Detailed Description of the AS-MAMP Algorithm . . . 50
  - 4.3.1 Adaptations to PBS as High-level Search of AS-MAMP . . . 50
  - 4.3.2 Motion Principles . . . 52
  - 4.3.3 Definition of Motion Profiles, Shapes, and Safety Distances . . . 54
  - 4.3.4 Graph Reservations . . . 56
  - 4.3.5 Conflict Detection . . . 59

4.3.6 SIMP: Objective Function and Heuristic Search . . . 60
4.3.7 SIMP: State Space . . . 61
4.3.8 SIMP: State Generation . . . 62
4.3.9 Replanning . . . 65

5 Evaluating the AS-MAMP Algorithm within the MAS Model 67
5.1 Model Evaluation . . . 69
5.1.1 Definition of Test Cases . . . 69
5.1.2 Benchmarking the High-level Search . . . 71
5.1.3 Benchmarking the Low-level Search . . . 73
5.1.4 Influence of Key Operational Parameters . . . 75
5.1.5 Influence of Planning Window . . . 76
5.1.6 Computational Efficiency and Scalability of AS-MAMP Algorithm . 77
5.1.7 Testing AS-MAMP on Real-world Airport Layout . . . 79
5.1.8 Stress-test: Potential Runway Throughput . . . 81
5.2 Discussion . . . 84
5.2.1 Reflection on the AS-MAMP Algorithm . . . 84
5.2.2 Reflection on the MAS Model . . . 85
5.2.3 Reflections on Generalised Hierarchical-Distributed Modelling Architecture . . . 86
5.3 Conclusions . . . 87

6 Autonomous Multi-Engine Taxiing Operations 89
6.1 Multi-Agent System Model . . . 92
6.1.1 Model specification . . . 92
6.1.2 Activity sequence of Aircraft Agents . . . 95
6.1.3 Routing algorithm . . . 96
6.1.4 Model calibration . . . 97
6.1.5 Verification and validation . . . 100
6.2 Experimental Setup . . . 100
6.2.1 Study assumptions . . . 101
6.2.2 Traffic situation . . . 101
6.2.3 Overview of analysis and key performance indicators . . . 103
6.3 Experimental Results . . . 105
6.3.1 Influencing factors on taxi time comparison . . . 106
6.3.2 Comparison of taxi time distributions per runway . . . 106
6.3.3 Predictability of taxi times . . . 108
6.3.4 Computational efficiency of routing algorithm . . . 108
6.3.5 Delay hotspots in the taxiway network . . . 108
6.3.6 Holding of inbound and outbound flights . . . 110
6.3.7 CTOT-slots of outbound flights . . . 110
6.3.8 Runway sequence and capacity of outbound flights . . . 113
6.4 Future Work . . . 113
6.5 Conclusion . . . 114

7 Engine-Off Taxiing of Outbound Flights 115
7.1 Multi-Agent System Model . . . 118

7.2 Experimental Setup . . . 121
7.3 Simulation: Results . . . 123
7.3.1 Comparison of Taxi Time Distributions per Runway . . . 124
7.3.2 Predictability of Taxi Time . . . 125
7.3.3 Delay Hotspots in the Taxiway Network . . . 125
7.3.4 Emission Hotspots in the Taxiway Network . . . 125
7.3.5 Comparison of Engine-on Time per Runway . . . 128
7.3.6 Computational Efficiency of Routing Algorithm . . . 129
7.4 Future Work . . . 129
7.5 Conclusion . . . 129

8 Improving the Realism of MAS Models for ASM Ops 131
8.1 Calibration of Model Parameters using Historical Track Data . . . 135
8.1.1 Methodology for Evaluating Historical Track Data . . . 135
8.1.2 Evaluation of Historical Track Data . . . 138
8.1.3 Comparing Calibrated Model to Historical Operations . . . 145
8.1.4 Discussion on Model Calibration and Future Work . . . 155
8.2 Modelling of Aircraft Towing Operations . . . 156
8.2.1 Reconstructing Historical Aircraft Towing Movements . . . 156
8.2.2 Extensions to the Multi-Agent System Model . . . 157
8.2.3 Evaluation of Aircraft Towing Operations . . . 158
8.2.4 Discussion on Towing Movements and Future Work . . . 161
8.3 Evaluating Operational Model Variations . . . 163
8.3.1 Overview of Simulated Model Variations . . . 163
8.3.2 Impact on Taxi Times and Distances . . . 164
8.3.3 Impact on Infrastructure Occupancies . . . 165
8.3.4 Discussion on Model Variations and Future Work . . . 167
8.4 Conclusion . . . 167

9 Towards Human–Autonomy Teaming 169
9.1 Agent-based Model of Human–Automation Interactions . . . 173
9.1.1 Role of the Human Operators . . . 173
9.1.2 Role of the MAS . . . 174
9.1.3 Interaction Agent and its Interaction Tools . . . 174
9.2 Modelling and Implementation of Human–MAS Interactions . . . 178
9.2.1 Case ‘Conformance to Taxiway Procedures of ATC’ . . . 178
9.2.2 Case ‘Departing Aircraft is Delayed’ . . . 181
9.2.3 Case ‘Alternative Pushback Path’ . . . 182
9.2.4 Case ‘Unavailability of Stand’ . . . 185
9.2.5 Case ‘Emergency demands Aircraft to Return to Stand’ . . . 186
9.3 Discussion and Future Work . . . 188
9.4 Conclusion . . . 188

10 Conclusions 191
10.1 Review of Findings . . . 193
10.1.1 Research Questions in Development Phase . . . 194
10.1.2 Research Questions in Validation Phase . . . 197

10.1.3 Research Questions in Exploration Phase 202
10.2 Reflection of Research Objective 203
10.3 Limitations and Research Directions 205

Abbreviations **209**

List of Figures **211**

List of Tables **215**

References **217**

Curriculum Vitæ **227**

List of Publications **229**

1

Introduction

In this chapter, the thesis topic of airport surface movement operations is introduced, and the focus on developing models, methods, and tools to study next-generation concepts of operations is motivated. Subsequently, the research objective and questions are presented, as well as the scope and the main underlying assumptions. The chapter concludes with an overview of the remaining chapters and suggested reading paths through the thesis.

1

The first part of this chapter was published in:

M. von der Burg, J. Kamphof, J. Soomers and A. Sharpanskykh, 'Towards next-generation airport surface movement operations using hierarchical-distributed multi-agent planning', *Transportation Research Part C: Emerging Technologies*, vol. 186, p. 105 583, 2026, ISSN: 0968-090X. DOI: 10.1016/j.trc.2026.105583.

The air transport sector faces three connected challenges. First, the demand is predicted to increase to around 10 billion annual passengers by 2050 [1], more than doubling the amount of 2019. Second, the sector strives to achieve net-zero emissions by the same time [1] to meet the Paris Agreement [2], whose goals were once more reaffirmed at the 2025 UN Climate Change Conference (COP 30) [3]. As third challenge, the level of safety must be sustained or further increased. Zooming in on airports as the key nodes in the air transport network, with large hub airports connecting both passengers and cargo between regions, operational disruptions often cause network-wide disturbances. As such, recent incidents during taxiing and at runways [4], [5] underscore the importance of safety of the operations.

The safe and efficient planning and control of all ground movements at an airport is referred to as Airport Surface Movement Operations (ASM Ops), which take place on the airside of airports as illustrated in Figure 1.1. Especially at large airports, ASM Ops are challenging, involving complex interactions among diverse stakeholders and operational systems [6]. Air Traffic Control Officers (ATCOs) provide clearances and guidance for aircraft and vehicles across the taxiway and runway network. Their decision-making is informed by other stakeholders such as airport operators, airlines, and ground handlers. While pilots and vehicle drivers are responsible for the safe operation of their vehicles and determine their taxi speed accordingly, they rely on the overall coordination by and communication through ATCOs.

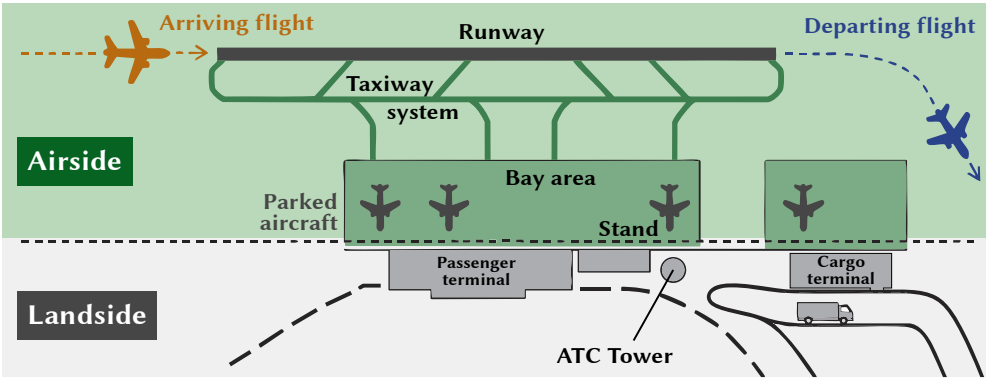


Figure 1.1: Overview of the airport environment; based on [7]

For ASM Ops, the three challenges of meeting the predicted demand, reducing emissions, and maintaining safety are highly interrelated. To meet the predicted demand, either larger or more aircraft are needed, and airports must facilitate this growth with additional capacity. However, it is expected that infrastructural expansions of airports are insufficient to achieve higher airport capacity [8]. Thus, the congestion on the airport surface will increase. As a result, ATCO workload is expected to rise [9] which, in turn, increases the risk of human error and may induce further delays and congestion. Furthermore, the taxiing times of aircraft, i.e. the duration of moving over the airport surface, become harder to predict. In addition to having a direct influence on the respective flights, this may trigger knock-on effects on the entire network, resulting in even higher inefficiencies [10].

1

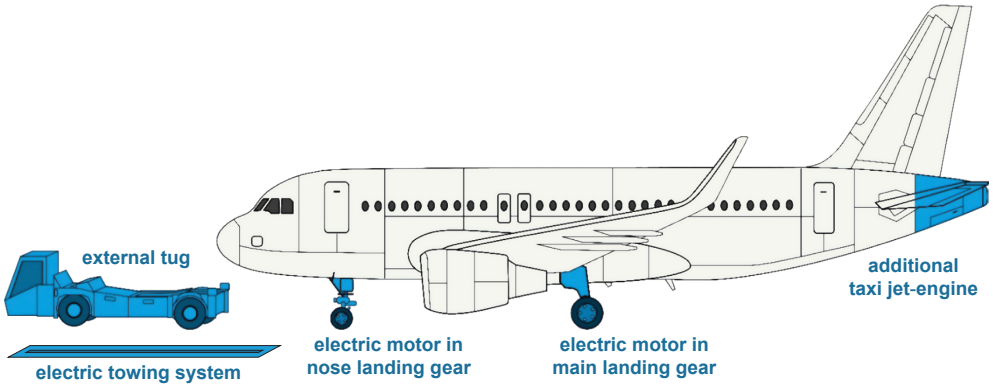


Figure 1.2: Overview of different engine-off taxiing concepts; based on [11]

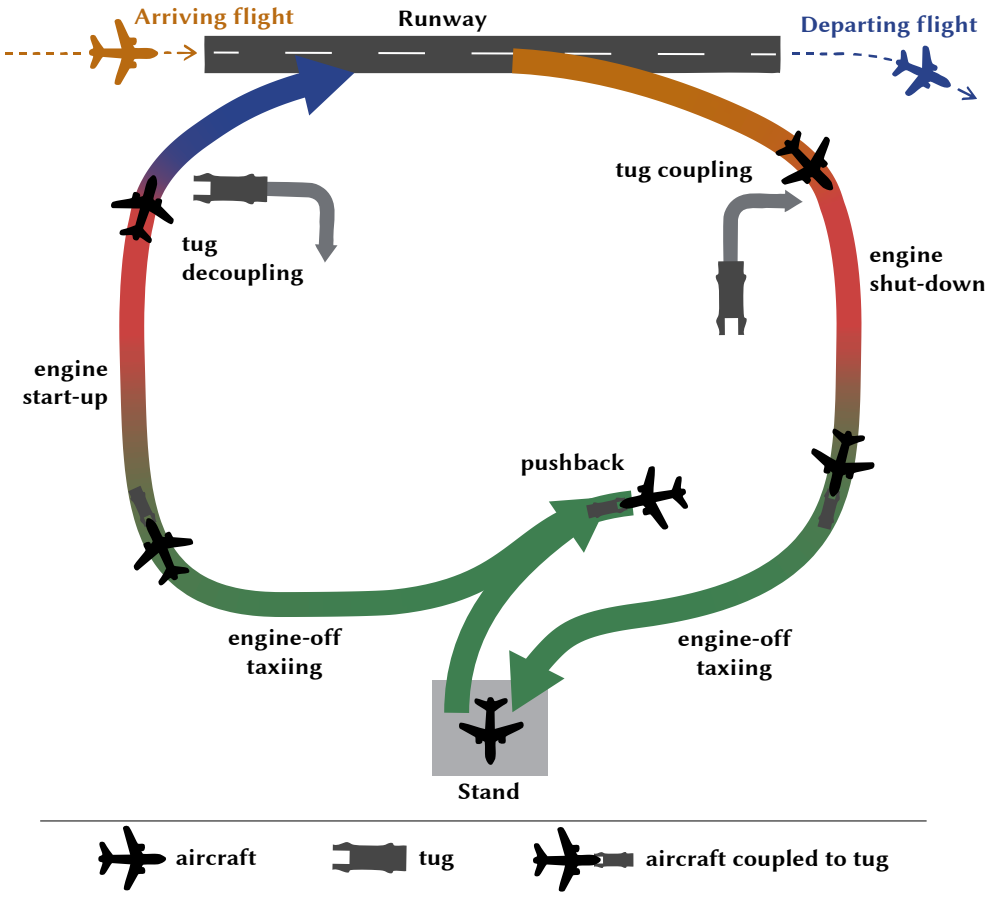


Figure 1.3: Schematic overview of how the TaxiBot concept serves as engine-off technique during inbound and outbound taxiing; based on [12]

Regarding emissions, the taxi times can reach a significant share of the total flight time especially for short-haul flights, with combined taxi-in and taxi-out times reaching up to 30 min for European flights [13]. Furthermore, an analysis using a fuel-burn model showed that between 2 % (long-haul flight, 9 630 km) and 17 % (short-haul flight, 350 km) of total fuel is burnt during taxiing [14]. The corresponding emissions can be effectively reduced when the aircraft do not have to use their engines for taxiing. In recent years, various new engine-off techniques have been developed, potentially offering significant decreases in carbon and noxious emissions [15], [16]. As shown in Figure 1.2, both internal systems such as electric motors mounted to the landing gears and external systems such as electrically powered tugs have been conceived. Among these concepts, the TaxiBot system [12] has so far reached the highest operational maturity. As illustrated in Figure 1.3, these tugs can tow aircraft to runways and back, capable of speeds up to 23 kts. Nonetheless, the use of external tugs adds additional vehicles to the traffic, again amplifying the issues of airport capacity, ATCO workload, and increased probability for human errors. Likewise, alternative propulsion technologies in development such as hydrogen or battery-powered aircraft require new concepts of operations (ConOps) that need to be thoroughly evaluated [17].

Besides introducing new technology for taxiing, another possible direction to address the current challenges is to increase the level of automation. In its AI Roadmap [18], EASA classifies AI applications into three levels, each further divided into two sub-categories. As shown in Figure 1.4, these range from human assistance (Level 1), to human–autonomy teaming (HAT, Level 2), and finally to advanced automation (Level 3).

Level 1 AI: assistance to human	Level 2 AI: human–AI teaming	Level 3 AI: advanced automation
• **Level 1a:** human augmentation • **Level 1b:** human cognitive assistance in decision-making & action selection	• **Level 2a:** human and AI-based system cooperation • **Level 2b:** human and AI-based system collaboration	• **Level 3a:** AI-based system performs human-overridable decisions & actions • **Level 3b:** AI-based system performs non-overridable decisions & actions (e.g. to support safety upon loss of human oversight)

Figure 1.4: EASA's classification of AI applications; based on [18]

In this thesis, the use of the term *AI system* follows the *intelligent agent* perspective of Russell and Norvig [20], which is also adopted by the Organisation for Economic Co-operation and Development (OECD) [19]. As shown in Figure 1.5, an AI system comprises sensors, operational logic, and actuators that perceive and influence an environment. In general, an AI system can be realised through models grounded in expert knowledge and/or historical data, i.e. using model-driven and/or data-driven approaches. Its reasoning may rely on symbolic techniques (rule- or logic-based), statistical techniques, or a combination thereof to support inference and decision-making. Notably, many established AI techniques such as symbolic search, planning, and logic-based reasoning are deterministic, yet are firmly within the AI field [20]. Conversely, a wide range of automation in operational systems is implemented using fixed, deterministic control logic (e.g. procedural sequencing, interlocks, and feedback control) without

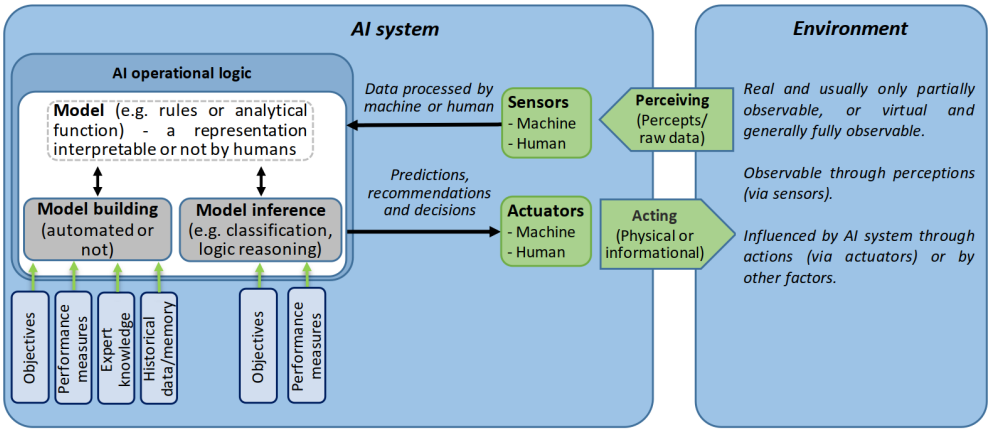


Figure 1.5: Schematic overview of an AI system and its elements, as defined by the AI Group of Experts at the OECD (AIGO); adapted from [19], Figure 1.4, p. 23

employing AI-based problem-solving or learning. Accordingly, throughout the thesis, *automation* is used as an operational umbrella term, while *AI system* and *AI-based* are used when referring to agent-based decision components and their underlying techniques.

Circling back to EASA's AI levels, an AI system applied in Levels 2–3 must be capable of selecting actions autonomously, rather than merely providing advisory outputs as assumed in Level 1. This shift also introduces well-known 'ironies' of higher automation: as systems take over routine control, humans may be placed in supervisory roles in which their situation awareness and manual or cognitive skills can degrade, yet they are still expected to intervene effectively when automation fails [21]. For AI-enabled automation, these concerns are compounded since opaque, adaptive AI create difficulties in calibrating trust and understanding system behaviour, particularly in off-nominal situations [22]. Importantly, Level 2 centres on *teaming*: humans work *interdependently* with an AI system towards a shared objective and exhibit teamwork behaviours such as coordination, communication, and adaptive allocation of functions and authority [23], [24]. HAT can be viewed as a specific class of human–automation *interaction*, which broadly covers how humans and automated systems exchange information and authority during operation [25], [26]. By comparison, HAT places stronger requirements on the quality of these interactions, including shared situation awareness, mutual predictability, and transparency of decision-making.

In ASM Ops, AI systems could provide extensive decision-support for ATCOs, pilots, and vehicle drivers, and could even take over certain repetitive tasks. In previous research, both NASA and DLR developed future ConOps for airports in the US and Europe based on surface trajectory-based operations (STBO), providing four-dimensional trajectories to coordinate taxi routes [27]–[29]. Other research assessed the achievable positioning tolerances of pilots and auto-pilot systems when following the 4D trajectories, and revealed that trajectory errors below 22 m are feasible to achieve, with mean errors even below 7 m for (partially) automated aircraft taxiing [30]. However, high levels of automation are deemed to be reachable only in the far-term [27]. Indeed, the envisioned human–autonomy teaming does not yet exist in real-world applications since today's technology is not

sophisticated enough [31]. To gain the trust of ATCOs, the automation must be proven to be capable of handling tasks autonomously in a safe and efficient way.

On the other hand, instead of the mostly centralised control exerted in present-day operations, future ConOps may rely on alternative control strategies. For instance, previous work studied concepts for air traffic management with decentralised control [32], [33]. Nonetheless, these may substantially change the work of ATCOs, and must thus be evaluated in detail with all involved stakeholders. In turn, any evaluation of future ConOps necessitates a control system model that represents the underlying operations with sufficient fidelity. For instance, the interrelated processes of ASM Ops as well as the new engine-off taxiing techniques must be incorporated to demonstrate the efficacy to safely coordinate also congested traffic situations. To compute realistic 4D trajectories, not only the various operational processes, but also the heterogeneous fleet of differently sized aircraft and ground vehicles with different kinematics must be represented in detail.

By contrast, most existing models of ASM Ops are geared towards studying either an isolated part of the system or one specific operational concept, mainly based on current operations. Furthermore, the previous studies often relied on multiple simplifying assumptions to handle the inherent complexity, e.g. pre-defining taxiing routes and focusing on a single bay area [29]. In turn, such models are inept to study the potentially wide range of ConOps to address new technological solutions, future operational approaches, as well as environmental requirements, let alone to compare the efficacy of different approaches with each other on a detailed level.

1.1 Related European Research Projects

Two recent European research projects, funded by SESAR 3 JU, explored concepts of operations and tools for automated and sustainable airport surface movement operations: the 'AEON – Advanced Engine Off Navigation' project [34] running from November 2020 till December 2022, as well as the 'ASTAIR – Auto Steer Taxi at AIRport' project [35], which was launched in September 2023 and was scheduled till February 2026.

The main objective of AEON was to define a coherent concept of operations for a mix of different engine-off taxiing techniques. As such, both onboard and external systems as well as single-engine taxiing as last resort were investigated. As part of the AEON ConOps [36], multiple algorithms were developed to schedule the fleet of towing tugs (as external system) [37] and to carry out path planning for all concurrent ground movements on the airside of airports [38]. Furthermore, human–machine interfaces (HMIs) for human–automation interactions were prototyped [39]. Throughout the project, various stakeholders such as airport personnel, ground controllers, pilots, and operational experts were consulted. Moreover, real-time simulations of operational scenarios at Paris Charles de Gaulle Airport were conducted to validate AEON's concept of operations as well as the developed tools and interfaces [40]. The validation activities were summarised in three reports, in terms of human performance and liability [41], safety [42], and cost–benefit [43].

Building on the outcomes of AEON, the ASTAIR project set out to design decision-support tools that enhance the safety, efficiency, and sustainability of airport airside operations through advanced automation, while reducing the workload of human operators [44]. For this purpose, AEON's support tools were developed further to take over repetitive control tasks and aid the decision-making of human operators. Furthermore,

prototypes of advanced HMIs for effective and flexible supervision and interaction with the support tools were designed. These tools were validated in human-in-the-loop experiments conducted at the end of the project [45].

For both projects, high-fidelity models were required to create an immersive experience for human operators to conduct meaningful validation studies of such novel concepts of operations. The models needed to be scalable to the varying taxiway networks of different airports, especially large hub airports. Moreover, they had to cope with uncertainty such as operational delays, changes to the runways in use, as well as technical and automation failures. To fulfil the goals of the projects, the conflict-free yet efficient and environmentally optimised routing of both aircraft and tugs was paramount. The underlying algorithms were required to be adaptable to handle, for example, dynamic changes, constraints and priorities, and interactions with human operators. The research presented in this thesis pertains to the two projects by addressing the above mentioned points, as outlined below.

1.2 Research Focus and Research Questions

To address the identified research gap, the objective of this thesis is to:

Research Objective

Design models, methods, and tools to investigate next-generation concepts of operations for automated and sustainable airport surface movement operations.

The contributions of this thesis intend to establish a framework for the systematic evaluation of concepts addressing the interrelated challenges of safety, capacity, and emissions in ASM Ops. Analysis examples include studying higher levels of automation up to fully-automated operations, comparing differing coordination schemes, and quantifying the impact of sustainability measures such as engine-off taxiing.

In the long term, operational systems corresponding to EASA Levels 2–3 should be co-designed through joint efforts across academia, industry, operational experts, and the human operators who will ultimately interact with, supervise, and manage such systems. At the same time, current staffing shortages in ATC make ground controllers already a scarce resource, increasing the need for efficient development and validation pathways. In this context, the models, methods, and tools developed in this thesis are positioned as a prerequisite for exploring the large design space of novel concepts of operations. Fast-time simulations provide the means to identify promising concepts, collect quantitative evidence, and inform discussions with operational experts. Subsequently, real-time simulations and human-in-the-loop experiments enable the assessment of operational feasibility and evaluation of human factors.

To address the research objective, the thesis proceeds in a stepwise manner guided by the research questions introduced below. The first step is to derive the requirements for designing detailed models of ASM Ops based on the real-world operations, leading to the first research question (RQ):

RQ 1 Modelling Requirements

Which modelling requirements must be met when investigating next-generation concepts of operations using a high-fidelity simulation of real-world airport surface movement operations?

Chapter 2 addresses this research question, along with providing an overview of the operational background of ASM Ops. To this end, the description focuses on the operational planning phase, since it is the most demanding one given its inherently short time horizons for decision-making. With the list of modelling requirements defined in Section 2.2, previous work is reviewed, forming the scientific foundation for the methodological choices as well as the remaining research questions, which are introduced in the following.

The design process for a next-generation control system presumably requires multiple iterations before being implemented in practice. To allow design freedom, the model shall accommodate studying concepts for next-generation ASM Ops across, for example, varying degrees of technological maturity, automation, and control centralisation. The computational model underlying the fast-time and real-time simulations must thus have a modular structure, allowing to be adapted and expanded throughout the design process when necessary.

This motivates choosing the modelling paradigm of multi-agent systems [20] to address the main research objective of this thesis of designing models to study future ConOps. Agent-based models offer inherent modularity, flexibility in modelling agent properties, interactions and control structures [46]. They are thus well suited to represent real-world systems, as various studies have shown. For example, multi-agent systems were used to model the coordination and planning of trajectories [47] as well as various other applications in robotics [48], [49], supply chains [50], passenger transport [51] and logistics [52] in cities, smart grids [53], social networks online [54] and in education [55], cloud computing [56], and network security [57]. An elaborate overview of applications of multi-agent systems, as well as the typically application-specific challenges of this modelling paradigm, are provided in [58], [59].

In the context of ASM Ops, multi-agent system modelling thus allows to organise the complex and intertwined operational tasks into multiple smaller tasks and assign these to various agents with different roles. This leads to the second research question:

RQ 2 Multi-Agent System (MAS) Model

Using the multi-agent system paradigm, how can the modelling requirements be addressed to design realistic simulation models for studying next-generation concepts of operations?

To facilitate the design of custom models for studying diverse concepts of next-generation ASM Ops, Chapter 3 introduces a generalised multi-agent architecture with a hierarchical-distributed structure. Based on the generalised architecture, specific model instances can be defined to study new concepts of operations characterised across various dimensions such as procedural aspects (e.g. selected taxiing techniques, effect of and compliance with existing/new procedures), degree of centralisation (from centralised control to distributed control with enacted system-wide constraints), and level

of automation per task (allocated to a human vs. an automated system), requiring different coordination schemas. The generalised architecture offers design freedom to develop more elaborate and tailored models, while enabling to iteratively refine the operational concepts in fast-time simulations and human-in-the-loop experiments with stakeholders. It further decouples the model structure from the embedded algorithms, enabling flexible selection of AI techniques in accordance with the investigated ConOps.

The proposed generalised architecture is illustrated by focusing on one model instance tailored to centralised planning for fully-automated control: despite posing significant challenges for its implementation in real-world settings, it is expected to offer the highest operational efficiency, and can thus be used as baseline for future comparisons with other model instances. In line with current ground control at airports, the movements of aircraft and ground vehicles over the airport surface are planned centrally in the MAS model. Following from the modelling requirements of ASM Ops, the planned agent trajectories must be conflict-free, operationally efficient, and calculated within a low computational runtime. As representation of the decision logic for coordinating the movements, the class of multi-agent motion planning algorithms offers in general promising techniques: such algorithms combine searching for a conflict-free path and for a suitable speed profile along that path. However, these have not yet been applied in detailed models of ASM Ops which address the modelling requirements from RQ 1. Thus, the third research question is as follows:

RQ 3 Multi-Agent Motion Planning

Which algorithmic developments are necessary for multi-agent motion planning algorithms to generate safe, efficient, and computationally scalable trajectories in realistic airport environments?

Multiple algorithmic enhancements were required, collectively forming the novel Multi-Agent Motion Planning on Airport Surfaces (AS-MAMP) algorithm, which is presented in Chapter 4. As its backbone, 4D trajectories are computed that satisfy both the operational processes as well as constraints necessary to coordinate with movements higher in priority, formalised as novel single-agent algorithm called Safe Interval Motion Planning (SIMP). It is inspired by the Safe Interval Path Planning (SIPP) algorithm [60] and its variants. SIMP further extends the capabilities of SIPP-based low-level planners to deal with all important characteristics of ASM Ops such as the agent kinodynamics and sizes, safety zones around agents, and the different operations like pushback, engine-start, tug coupling, and decoupling. In the AS-MAMP algorithm, multi-agent coordination is based on the Priority-Based Search (PBS) algorithm [61] and its greedy variant GPBS [62].

To benchmark the AS-MAMP algorithm, the performance of multiple PBS-variants as well as sequential planning based on the first-come-first-served (FCFS) principle often used in literature is assessed in Chapter 5. Moreover, to showcase the algorithmic properties of SIMP, it is benchmarked against SIPP and A*, requiring additional assumptions to calculate trajectories in the AS-MAMP algorithm. Furthermore, a stress-test of the AS-MAMP algorithm is used to establish the potentially achievable runway throughput for both regular, Multi-Engine Taxiing (MET), as well as engine-off taxiing using tugs such as the TaxiBot, referred to as Tug-Enabled Taxiing (TET).

Notwithstanding, to fulfil the modelling requirement of representing ASM Ops realistically, the MAS model must cope with the complexities of real-world airport layouts and busy flight schedules. However, the aspects influencing the real-world operations vary largely between airports [63] with different operational roles and control functions of human operators. Furthermore, human factors play an important role in the safety and efficiency of the operations.

To limit the scope of this thesis, the verification and validation of the MAS model is carried out based on one large hub airport, namely Amsterdam Airport Schiphol. This is one of the largest airports in Europe and comprises a complex taxiway network, connecting many bay areas to its six runways. These are necessary to sustain operations also in the adverse weather conditions with strong winds from changing directions that occur frequently in the Netherlands. For further scoping, the MAS model for ASM Ops is primarily studied using fast-time simulations of fully-automated operations.

First, multi-engine taxiing is studied as it represents the taxiing technique currently used at large airports. By simulating historical flight schedules, the performance metrics such as taxi time and taxi distance can be compared to those of the real-world operations. This leads to the next research question:

RQ 4 Multi-Engine Taxiing (MET)

Using the MAS model, how do fully-automated multi-engine taxiing operations compare to the present-day, real-world operations in dense flight traffic at Amsterdam Airport Schiphol?

Chapter 6 presents results of fast-time simulations, based on historical flight schedules of two of the busiest days at Schiphol to date. Besides the dense traffic, the two operational days were chosen as the two most common Runway Modes of Operation (RMO) at Schiphol, referred to as 'RMO North' and 'RMO South', were each active on one of the two days.

Subsequently, the conducted MET simulations can be used as baseline to investigate the effects of shifting to engine-off taxiing. Since the concept of TaxiBots [12] is furthest developed and thus the most promising one at this point, the scope of this thesis is limited to exploring tug-enabled taxiing. This leads to the next research question:

RQ 5 Tug-Enabled Taxiing (TET) of Outbound Flights

Using the MAS model, which operational consequences may result from requiring all outbound flights to use tug-enabled taxiing, under the assumption of fully-automated operations in dense flight traffic at Amsterdam Airport Schiphol?

Chapter 7 presents the conducted fast-time simulations using the historical flight schedules from Chapter 6. The tug fleet is assumed to be large enough to cater to every outbound flight. As such, task assignment for the tugs is neglected.

During the course of the research presented in this thesis, elaborate discussions were carried out with ATCOs and operational experts from Amsterdam Airport Schiphol. While they highlighted the overall realism of the simulations, they also provided valuable points to further improve the model fidelity. Their feedback gives rise to the two following research questions:

RQ 6 Calibration of MAS Model

How can historical aircraft ground tracks be used to calibrate the MAS model so that it closely reproduces real-world taxi traffic, and what do they reveal about suitable taxi speed limits for highly-automated operations?

RQ 7 Aircraft Towing Movements

Using the calibrated MAS model, which impact do aircraft towing movements have on the taxi performance of regular traffic?

To answer RQ 6, Chapter 8 presents an elaborate evaluation of historical track data projected onto the taxiway graph to calibrate the model parameters. Accordingly, the alignment between historical and simulated operations is assessed across different taxiway areas. For RQ 7, the MAS model is extended with aircraft towing movements, which is also part of Chapter 8. Based on multiple simulations with differing counts of towing movements, their impact on the regular taxi traffic is subsequently evaluated.

The studies presented in Chapters 4 and 6 to 8 did not involve any interactions with human operators, as they assumed fully-automated operations consistent with EASA Level 3. However, to explore operational concepts grounded in human–autonomy teaming (EASA Level 2), the MAS model must also accommodate input from human operators, validated and refined through human-in-the-loop experiments. As this is a very broad topic, it is only touched upon at the end of this thesis. The insights gained from the previous RQs allow to explore how the developed MAS model, its underlying methods, as well as the simulation platform as decision-support and design-support tool can be extended to embed interactions with human operators. This leads to the final research question:

RQ 8 Human–Automation Interactions

How can human–automation interactions be embedded into the MAS model, methods, and tools for next-generation airport surface movement operations?

Chapter 9 presents the explorations that were carried out on the basis of use cases from interviews and workshops conducted with operational experts as part of the ASTAIR project [63]. The MAS model is integrated into a schematic agent-based framework of human–automation interactions for airport surface movement operations. Following a fictitious ATCO through her shift at Amsterdam Airport Schiphol, different exemplary interaction scenarios are presented, focusing on the technical implementation within the MAS model.

1.3 Research Scope and Assumptions

In this section, the research scope and the core assumptions adopted throughout the thesis are summarised. First, the operational context is framed, followed by an overview of assumptions related to engine-off taxiing, the modelled decision-making, case-study coverage, and aspects regarding human factors.

Operational Context

- The thesis targets next-generation concepts and a functional, role-based representation of airport surface operations.
- Its scope is centred around developing the algorithmic functionality for trajectory planning in the tactical and operational phases of airside surface movement operations.
- The considered future airport infrastructure is assumed to provide reliable communication, as well as prediction and surveillance capabilities to minimise operational uncertainties within the considered time horizons.

This thesis focuses on automated and sustainable airport surface movement operations from the perspective of trajectory planning and coordination on the airside. The modelled multi-actor system is not intended to replicate present-day organisational as well as operational roles but instead focus on the functional roles that the actors shall fulfil. To keep the research objective tractable and attributable, upstream processes such as gate planning, runway allocation, and passenger connection management are treated as boundary conditions. Moreover, where available, the historical flight schedule data and the corresponding Airport Collaborative Decision Making (A-CDM) milestones, as standardised sequence of significant operational events across the inbound, turnaround, and outbound phases of a flight, are used as is. This aligns with the research focus on developing and evaluating planning functionality rather than forecasting the operational context. Consequently, network-level effects that drive A-CDM milestone uncertainty are considered out of scope. Instead, short-horizon predictions are assumed reliable, consistent with extensive work on data-driven prediction approaches [64], [65] and operational efforts such as Schiphol's deep turnaround concept [66]. The infrastructure required for digital information exchange (e.g. through AeroMACS [67]) and surface surveillance (e.g. following the A-SMGCS specifications [68]) is assumed to be operationally available. Besides, any issues and risks in the domain of cyber security that would impair communication and disrupt the operations are disregarded.

Assumptions for Modelling of Engine-off Taxiing Operations

- Engine-off taxiing is modelled using external tugs based on the TaxiBot [12] as reference system, referred to as Tug-Enabled Taxiing (TET).
- In the studied scenarios, TET is applied to all outbound flights.
- The tug fleet is assumed large enough to tend to every tow task without causing delays.
- Task assignment of tugs and fleet management policies are not modelled.

From the range of engine-off taxiing techniques, the TaxiBot concept [12] has reached the highest operational maturity, with a few tugs deployed at selected airports. The underlying operational concept is referred to as tug-enabled taxiing (TET) in this thesis. As industry stakeholders generally regard inbound TET as economically unattractive and operationally impractical, TET is only considered for outbound flights in the considered scenarios. While the TaxiBot concept is currently only certified for a few narrow-body

aircraft, TET operations are assumed for all outbound flights, to study the system-wide operational effects and their impact on algorithmic performance. The tug fleet is assumed to be sufficiently large to serve each towing task without inducing delays, as aircraft would plausibly be assigned conventional taxiing in case of tug unavailability in real-world operations. Consistent with the thesis-wide scoping towards surface trajectory planning, tug task assignment and fleet management policies (e.g. dispatching, prioritisation, charging/maintenance logistics) are treated as out of scope.

Assumptions for Modelled Decision-making

- The planning problem is treated as deterministic to support the verification and validation of the developed algorithms and their underlying functionalities.
- Except otherwise noted, delays are assumed to occur *prior* to the start of taxiing and outside the planning horizon.
- Aircraft and tugs are assumed to execute their assigned trajectories within tight tolerances.

The considered problem of generating conflict-free trajectories for all concurrent surface movements is treated as deterministic, to keep the problem tractable and support the necessary verification and validation efforts. Based on the drawn assumption of advanced prediction and surveillance capabilities, any operational delays are assumed by default to be realised prior to taxi start. This limits the necessary replanning frequency and isolates the necessary algorithmic efforts to coordinate the concurrent movements. In analogy, execution is assumed to be highly conformant: prior work on future concepts of flight-deck automation motivates that pilots (and tug operators) could follow the provided ground trajectories with high precision [30].

Covered Case Studies

- Two types of airport environments are used: a synthetic bottleneck-based layout, and layout variants of Amsterdam Airport Schiphol.
- Aircraft are grouped into six categories in accordance with the ICAO aerodrome reference codes [69].
- Pushback and engine-start manoeuvres are represented; standard pushback procedures are used where available.
- The studied scenarios are evaluated using fast-time simulations, assuming nominal conditions except otherwise noted.

In this thesis, two types of airport environments are used in the case studies. A synthetic bottleneck-based layout is used to develop and stress-test algorithmic functionality under controlled conditions. To assess the realism and scalability in a complex, high-capacity environment, variants of the layout of Amsterdam Airport Schiphol are used, given its demanding, organically evolved taxiway network and multi-runway operating context. These layout choices support stronger verification and validation of the developed routing and coordination logic, while maintaining the research focus on functional and algorithmic contributions rather than airport-specific procedures.

Aircraft are grouped into six categories based on the ICAO aerodrome reference codes [69]. Pushback and engine-start manoeuvres are explicitly modelled, and where available, standard pushback paths (including push-pull manoeuvres) are applied as procedural constraints based on aircraft category. Fast-time simulations are used to obtain the results reported in this thesis, with nominal operations assumed except otherwise noted. Where available, historical flight schedules and corresponding runway configurations are used as simulation inputs to enable a direct comparison between simulated and historical performance indicators at different aggregation levels.

Human Factors

- Given the primary focus on fully automated concepts of operations, human decision-making is *not* modelled, and impacts on human factors are *not* assessed.
- Although human–autonomy teaming is acknowledged as a motivation and long-term goal, it constitutes a research area in its own right and therefore lies beyond the scope of this thesis.
- As an outlook, a limited set of interactions is explored to demonstrate technical feasibility, but not to provide validated human-in-the-loop evidence.

The thesis primarily evaluates automation concepts under the simplifying assumption of fully-automated operations, and therefore does not model human operator behaviour or quantify impacts on controllers, pilots, or ground personnel. Therefore, any assessment of human factors lies beyond the scope of this thesis. Although human–autonomy teaming is acknowledged as a key motivation and long-term goal, rigorous assessment requires dedicated socio-technical evaluation frameworks and iterative human-in-the-loop experiments, which also fall outside the thesis scope.

As a first step towards future research on teaming in ASM Ops, a limited set of interactions is explored towards the end of the thesis. These cases aim to illustrate how technical interfaces and information exchange between human operators and the automated system could be implemented. Accordingly, the automated side is modelled with rule-based adaptations rather than sophisticated, adaptive decision-making. As such, the explored interactions are intended to demonstrate feasibility rather than to provide validated evidence from human-in-the-loop studies.

> The assumptions underlying the models, methods, and case studies presented in this thesis are further discussed in the respective chapters.

1.4 Thesis Overview

While this thesis is structured to be read front to back, most chapters can be read independently as they are written as scientific papers. As guidance, Figure 1.6 provides an overview of the chapters, as well as suggested reading paths through the thesis.

Chapter 2 outlines the relevant operational background of ASM Ops, and defines a list of modelling requirements deemed indispensable to model ASM Ops realistically (RQ 1). Subsequently, previous research related to modelling such operations and multi-agent path

1

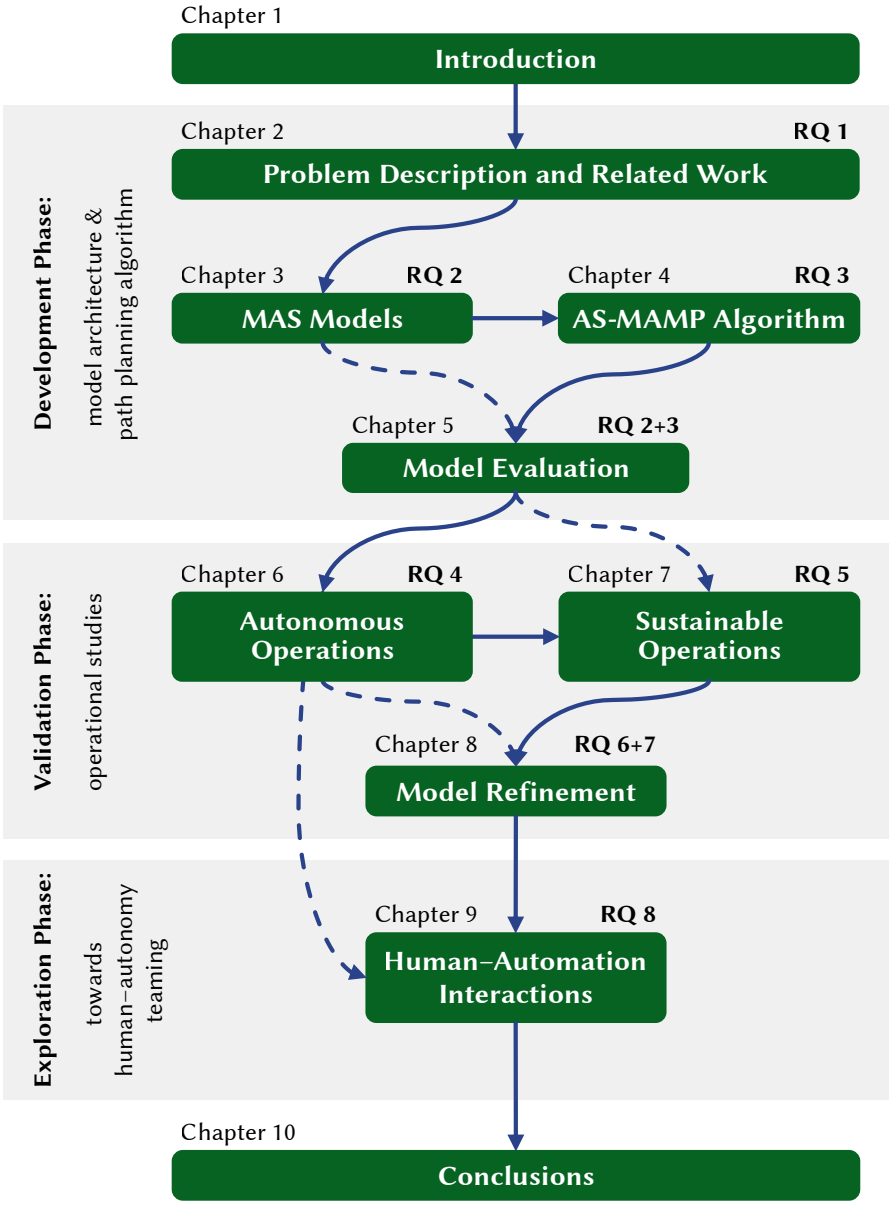


Figure 1.6: Thesis outline with main reading path (solid lines), and possible shortcuts (dashed lines)

planning algorithms is summarised. This chapter concludes by describing the modelling approach to address the requirements using the paradigm of multi-agent systems with built-in multi-agent motion planning.

In Chapter 3, the generalised architecture for instantiating multi-agent system models with hierarchical-distributed agent relations is presented (RQ 2). To illustrate the framework, a MAS model instance for fully-automated operations with centralised planning is specified. Chapter 4 then elaborates on the AS-MAMP algorithm that represents the agent decision logic for planning conflict-free trajectories by constructing a priority order between all concurrent movements of aircraft and ground vehicles (RQ 3). Subsequently, Chapter 5 evaluates the MAS model by assessing the performance of the AS-MAMP algorithm as its most important planning component. The effects of important algorithmic as well as operational parameters on the AS-MAMP algorithm are discussed by running fast-time simulations on both a synthetic as well as a real-world airport layout. The chapter ends with a discussion on the chosen modelling approach, its limitations and implementation challenges, and directions for future work.

Shifting to operational studies, Chapter 6 presents the investigation of fully-automated ASM Ops in which aircraft deploy multi-engine taxiing (RQ 4). The agent kinodynamics are calibrated using historical trajectory data corresponding to the flight schedule used in the fast-time simulations. The influence of the planning window and replanning period are evaluated by analysing the trade-off between the operational efficiency, predictability, and computational runtime. Furthermore, various operational consequences such as delay hotspots, taxi time efficiency, inbound and outbound holding, compliance with departure slots issued by Eurocontrol (i.e. CTOT-slots), and runway capacity are compared to their historical counterparts.

Chapter 7 investigates the potential operational impact posed by tug-enabled taxiing (RQ 5). As such, the core drivers for the expectable influence on the key performance indicators such as the taxi times are analysed. The chapter is complemented by evaluating the impact on delay hotspots, emission hotspots, among others, and providing a rough estimate of the achievable fuel and emission savings through tug-enabled taxiing.

In Chapter 8, tracks of historical aircraft ground trajectories are evaluated to address the points of improvement gathered in discussions with ATCOs and operational experts throughout the course of this thesis. Based on the conducted investigation, the calibration of the model parameters is improved (RQ 6). Furthermore, the MAS model is extended to assess the impact of aircraft towing movements on regular traffic (RQ 7). Subsequently, multiple operational model variations are compared and discussed.

In Chapter 9, the MAS model is embedded into an agent-based schema for human–automation interactions, allowing human operators to share information and authority with the MAS model (RQ 8). Different exploratory interactions are presented stemming from workshops and interviews that were carried out with operational experts as part of the ASTAIR project [63]. The context of each interaction is introduced, followed by a description of how the interaction was modelled and implemented in the MAS model, as well as placing it in broader perspective within ASM Ops.

The answers to the research questions defined above are summarised in Chapter 10. Moreover, the main conclusions and contributions of this thesis are outlined. The thesis concludes by providing directions for future research based on the identified limitations and assumptions of the presented work.

2

Problem Description and Related Work

This chapter first introduces the problem of airport surface movement operations (ASM Ops), and defines a list of requirements that are deemed indispensable to model the operations realistically (Section 2.1). Based on these, previous research related to modelling ASM Ops and multi-agent path planning algorithms is discussed in Section 2.3. This chapter concludes by providing a summary of the generalised multi-agent system model architecture as approach to address the identified research gaps (Section 2.4).

2

This chapter is based on the following publication:

M. von der Burg, J. Kamphof, J. Soomers and A. Sharpanskykh, 'Towards next-generation airport surface movement operations using hierarchical-distributed multi-agent planning', *Transportation Research Part C: Emerging Technologies*, vol. 186, p. 105 583, 2026, ISSN: 0968-090X. DOI: 10.1016/j.trc.2026.105583.

This chapter first describes the operational background of current and potential future airport surface movement operations. Then, modelling requirements are formulated for creating high-fidelity models to study next-generation operational concepts. Moreover, the modelling requirements will guide the assessment of previous work carried out in Section 2.3 as well as for proposing a generalised system modelling architecture in Section 2.4.

2.1 Operational Background

In its current form, ASM Ops are controlled and coordinated by Air Traffic Control Officers (ATCOs) who control the movements on the airport surface from the centrally located control tower. To ensure safe and efficient operations, the ATCOs provide clearances and guidance for all aircraft and ground vehicles moving along the taxiway network. ATCOs rely on other stakeholders to aid their decision-making, such as airport operators, ground handlers, and airlines that manage e.g. the flight schedules, stand allocation, and turn-around processes. As pilots and drivers are responsible for the safety of their individual vehicle, they determine the taxiing speed themselves. However, as they are unaware of the entire traffic situation and pertaining information, they rely on the overall coordination by and communication through the ATCOs, and are obliged to follow the ATCOs' instructions. As consequence, ASM Ops are an inherently cooperative system.

Currently, aircraft mainly use their own engines to move over the airport surface, referred to as multi-engine taxiing (MET) in this thesis. After landing, inbound flights taxi to the stand, during which their engines cool down. Then, the turn-around processes commence, some of which are to unload the baggage, refuel, clean, and let new passengers board the aircraft. Once ready for pushback with a pushback tractor (PBT) coupled to the aircraft, the existing taxi procedures can be split into three parts: 1) the pushback and (possibly) push-pull manoeuvre, 2) PBT decoupling and engine-start, and 3) taxiing to the runway. Before step 3) can be performed, the aircraft engines must be running. The engine-start procedure usually begins once the pushback commences. Only in the case of a push-pull manoeuvre, engine-start must be performed entirely after finishing the pull manoeuvre as it is carried out to avoid engine blast.

In next-generation ASM Ops, new means of taxiing as well as novel aircraft propulsion systems are envisioned. As such, different engine-off taxiing techniques have matured beyond the research stage, with the TaxiBot system [12] being closest to operational deployment. We thus pick the latter as example of a future ConOps, referred to as tug-enabled taxiing (TET) in this work. Based on the TaxiBot specification and interviews with operational experts, outbound TET operations can be divided into four stages: 1) pushback and direction switch, 2) tug-enabled taxiing to a decoupling location close to the runway, 3) decoupling and all-clear signal between tug and aircraft, and 4) taxiing onto the runway. Depending on airport regulations, engine-start can potentially be performed during stage 2). Nonetheless, holding at the decoupling location may be necessary when the duration of pushback, taxiing, and decoupling is shorter than the time needed to start the engines. For arriving aircraft, TET comprises three relevant stages: 1) exiting the runway and taxiing to a coupling location close to the runway, 2) coupling to the waiting tug, and 3) tug-enabled taxiing to the ramp.

For both flight directions, the tugs have to drive to the coupling location, and once decoupled travel to the next assignment or the tug-base. Airports may provide extra service-roads, or require them to use the taxiway network managed by ground control. Eventually, the tugs are expected to be equipped with electric engines [16]. In turn, this requires recharging over the day of operations, either at dedicated charging stations or at the stand, affecting the task assignment and route planning of tugs after decoupling.

2.2 Modelling Requirements

To study new concepts for ASM Ops in detail, the real-world operations must be modelled, simulated, and thoroughly evaluated, requiring a high-fidelity model. In turn, it must address the requirements derived from the operational background. Using Figure 2.1, the modelling requirements are discussed below.

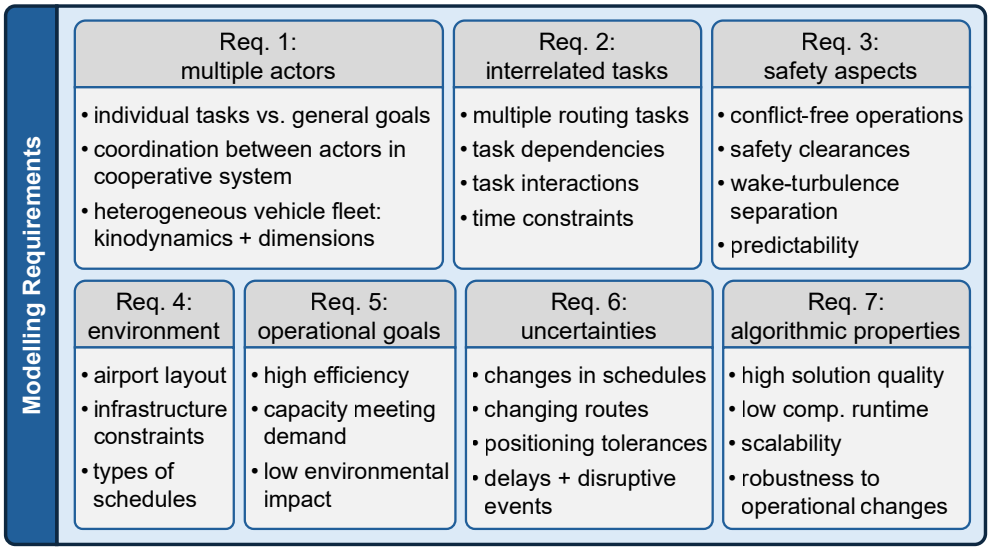


Figure 2.1: Overview of modelling requirements to study new concepts of operations for airport surface movements in detail

Requirement 1: system model shall represent multiple interacting actors

Various stakeholders partake in ASM Ops, sharing the general objectives of safe, efficient, and predictable operations besides their individual tasks, which may be conflicting between actors. Aircraft and additional vehicles such as tugs share limited resources, i.e. runways, taxiways, and bay areas. Differences in dimensions across aircraft types and between aircraft and tugs lead to a heterogeneous vehicle fleet. Moreover, their kinodynamics or non-holonomic properties must be accounted for, i.e. finite acceleration and deceleration, minimum speed to avoid stop-and-go, and speed restrictions on both straight as well as curved taxiway segments. Due to the strict safety standards, all actors are part of a *cooperative system*, making coordination between actors and control of their actions essential.

Requirement 2: system model shall accommodate the interrelated tasks

In general, ASM Ops comprise different operational tasks such as regular taxiing, pushback manoeuvres, engine-start, holding, and tug operations. In parts, these tasks depend on and form interactions between each other, e.g. between aircraft in queues, landing aircraft vacating the runway, or in the bay areas. Moreover, some tasks have associated time constraints that must either be surpassed or reached in time (i.e. deadlines). Examples include an arriving flight whose assigned stand is still occupied, a departing flight that must take off within a departure slot due to airspace restrictions (i.e. CTOT-slots assigned by Eurocontrol), or aircraft in need of a tug or towing vehicle.

Requirement 3: system model shall comply with safety-related aspects

The operations must meet the strict safety standards at all times. As such, all vehicles must keep sufficient safety clearances to each other while moving over the airport surface. Furthermore, any engine-blast risk must be minimised and sufficient wake-turbulence separation during takeoff and landing must be kept. Moreover, the system model should exhibit predictable and consistent behaviour under both nominal and non-nominal operational conditions, which is essential for the identification and mitigation of potential hazards.

Requirement 4: system model shall represent airport environments in detail

To represent diverse ASM Ops across different airport sizes and use cases, the system model must be capable of accommodating variations in airport topology and complexity. Especially for large airports, such as Amsterdam Airport Schiphol (AAS), the taxiway system is rather complex and contains not only straight and curved segments, but also compounded intersections, runway crossings and go-arounds, long single-lane taxiways, and designated holding areas. Other infrastructural constraints include the inability of aircraft to turn around on a taxiway segment, temporarily unavailable taxiways, limited stand space, and pushback procedures. Additionally, the model should incorporate operational aspects such as arrival and departure schedules, and the runway mode of operation (RMO).

Requirement 5: system model shall fulfil operational goals

The system model shall optimise the operations to meet key operational objectives such as high operational efficiency through the reduction of taxi times and delays, and maintaining throughput levels that align with current and projected traffic demand. Furthermore, the model should account for additional performance metrics such as the environmental impact, e.g. fuel consumption and emissions. This allows to assess the trade-offs between different operational goals as well as compare alternative operational strategies under varying scenarios.

Requirement 6: system model shall accommodate operational uncertainties

In actual operations, changes to planned actions or anticipated effects arise that must be handled accordingly, also during the routing of vehicles [70]. Examples include changes to flight schedules, task assignments of tugs, or changing runway configurations. Such changes often have short lead times (e.g. RMO changes may be known less than 30 min in advance). During taxiing, vehicle routes may change due to either operational reasons or errors in following the instructions provided by ATC. Furthermore, positioning tolerances

must be accounted for, due to sensor noise or technological limits in precisely executing the assigned vehicle trajectories. Delays in procedures and actions, communication, and various other forms [6], up to (confined) disruptive events such as sudden vehicle breakdowns, may occur throughout the operations.

Requirement 7: decision-support algorithms shall be effective and efficient

The decision-support algorithms deployed in the system model shall deliver high-quality solutions, generating conflict-free and efficient trajectories that are operationally feasible and that realistically reflect operational constraints. To enable fast-time simulation and iterative optimisation, the algorithms shall achieve low computational runtimes and scale with airport layout complexity and traffic density. Moreover, they shall remain robust to dynamic operational changes mentioned in Requirement 6, thereby ensuring reliable performance across a range of scenarios.

2.3 Related Work

In line with its current form of control, most previous work modelled ASM Ops as a centralised system, and studied related sub-problems. Other research has focused on developing new concepts of operations of ASM Ops as a centralised, but also as a distributed system. We will describe these in the next subsection, and thereafter outline the suitability of multi-agent path planning algorithms for planning conflict-free trajectories in ASM Ops.

2.3.1 Modelling of Airport Surface Movement Operations

In previous research, ASM Ops were divided into specific optimisation problems: the Ground Routing Problem (GRP) seeks conflict-free taxi schedules, typically framed as MILP or GA formulations [71], while the Runway Scheduling Problem (RSP) maximised runway throughput via MIP, DP, branch-and-bound, or heuristics [72], [73]. Guépet, Briant, Gayon *et al.* [74] reviewed studies that fully or partially couple GRP and RSP, and argued that their better integration significantly reduces the taxi times. Additional sub-problems tackled runway-induced taxi congestion through gate-holding [75]–[77], or pushback control and derated takeoffs to lower emissions [78].

Other research focused on approaches for predicting taxi times and anticipating delays under uncertainty [79], [80]. To this end, different approaches were used, e.g. using a stochastic model [70], chance-constraint programming [81], or a fuzzy rule-based approach [82]. Other studies used data-driven approaches to predict the landing times of aircraft [83], [84], to visually show the runway exit prediction [85], or to provide recommended gate-hold times of outbound aircraft (using historical trajectories to fix their paths) [86].

Collectively, these studies have shown tangible gains in operational efficiency and predictability by optimising the taxi times as well as narrowing the forecast errors. However, they still treat ground, runway, and turnaround processes as weakly-coupled subsystems. As a consequence, system-wide objectives influenced by dependencies across process boundaries are not optimised. This motivates integrated, concept-driven frameworks for assessing new concepts of operations (ConOps), as discussed in the next subsection.

2.3.2 Studying New Concepts of Operations

As approach towards fully-automated tower operations, Morris, Pasareanu, Luckow *et al.* [6] proposed a centralised automation system using discretised time and a grid-based layout with obstacles to represent taxiway networks. Their prototype of a surface management system focuses on data analysis for behaviour modelling, as well as continuous route planning, scheduling, and monitoring.

NASA's ConOps for far-term Surface Trajectory-Based Operations (STBO) envisions that pilots actively participate in precisely meeting time-based goals [27]. In two phases, their approach uses time-based 4D taxi clearances first provided as coarse targets at specific locations, and eventually as high-precision trajectories. However, for the second phase, the authors highlighted that more research into advanced flight deck equipment is needed. Focusing on the first phase, various fast-time and human-in-the-loop trials using their developed decision-support tools showed promising results for more efficient and predictable operations [87]–[90].

Using a simulation platform of similar scope, DLR tested prototypes of the Airport Collaborative Decision Making (A-CDM) [91] and the Advanced Surface Movement Guidance and Control System (A-SMGCS) [92] approaches for European airports. Likewise, their human-in-the-loop simulations showed similar operational benefits when pilots adhere to the conflict-free 4D trajectory clearances [93]. In a joint effort, NASA and DLR also developed a shared concept of operations and evaluated it using fast-time simulations [29]. However, the taxi routes were pre-defined and were planned on a first-come-first-served basis. Both aspects likely lead to sub-optimal solutions and may not be applicable to complex traffic situations.

In general, centralised systems may lack knowledge of local conditions affecting operations, and communication might be impeded by capacity and bandwidth issues [33]. Alternatively, ASM Ops can also be modelled with decentralised control, as studied by Udluft [33] for a basic airport layout. In such a system, the communication and control mechanisms are shifted to multiple local entities. While these approaches enhance robustness, resilience, and the quality of local solutions, they do not guarantee a global optimum and may lead to inefficient use of global resources.

Collectively, simulation platforms that focus on evaluating novel concepts for ASM Ops commonly generate realistic 4D trajectories but rather use simple optimisation techniques to coordinate the movements, often following the FCFS principle. As already outlined in the previous section, paths can also be planned without such stringent order with potentially better solutions. To this end, the domain of multi-agent path planning (MAPP) provides a multitude of algorithms to solve both simplified and complex problems, as discussed next.

2.3.3 Multi-Agent Path Planning Algorithms for Conflict-free Routing

Interestingly, many MAPP-related papers motivate their work with ASM Ops, e.g. [94]–[96], while such algorithms are often neglected in operations-focused research, e.g. [79], [97]–[99]. To close this gap, we first provide an overview of important MAPP concepts and algorithms, from solving classical problems towards using them for real-world applications. As summarised below, two-level solvers stand out in their wide range of applicability, with Priority-Based Search (PBS) being especially suitable for ASM Ops. In a separate subsection, we then outline the Safe Interval Path Planning (SIPP) algorithms as state-of-the-art

single-agent planner used in two-level solvers. Finally, we provide a summary of the suitability of different path planning algorithms for ASM Ops.

2 Overview of Algorithms to Solve MAPP Problems

Much research has concentrated on the classical multi-agent path finding (MAPF) problem in which several assumptions are utilised to simplify the search: the environment is defined as static, both time and space are discretised, agents are depicted as points that only occupy a single location per time step, and move between these within unit time [100]. Moreover, it is typically assumed that the generated plans are perfectly executed [101]. Comprehensive overviews of the MAPF problem and its algorithmic landscape are provided in [102] (MAPF definitions, variants, benchmarks), [103] (MAPF variants focused on search-based optimal solvers), and [104] (classical but also beyond-classical MAPF solvers).

For solving classical MAPF problems, two-level solvers have proven to be most suited: they combine the strengths of different algorithms and yield better solution quality and lower computational times than other approaches such as reduction-based, rule-based, or other one-level search-based solvers. Prominent examples of optimal two-level MAPF solvers are M* [105], the Increasing Cost Tree Search (ICTS) [106], and Conflict-Based Search (CBS) [107].

The latter has found wide use to generate optimal solutions to classical MAPF instances. CBS is based on the ideas of building a binary conflict-tree during the search, which is expanded using a best-first search to sequentially resolve all conflicts between agents. To improve its computational runtime, several sub-optimal variants were proposed [108]–[110]. While these algorithmic improvements showed promising results in trading off solution quality for runtime, any CBS-variant may still be too slow for applications requiring close to real-time path planning computations, especially when relaxing the assumptions of classical MAPF, as done by e.g. Li, Surynek, Felner *et al.* [111] and Walker, Sturtevant and Felner [112].

As alternative two-level solver, Ma, Harabor, Stuckey *et al.* [61] introduced the Priority-Based Search (PBS) algorithm. PBS generates a priority-tree to explore the state space by applying different priority orderings for the agents, requiring a lower prioritised agent to avoid all paths of those higher in priority. The priority-tree is expanded using a depth-first search, resulting in fast execution times. While PBS is unbounded suboptimal and incomplete, the authors claim that it provides near-optimal solutions, also on an airport-like map (DAO-map brc202d). Recently, Chan, Stern, Felner *et al.* [62] introduced Greedy PBS (GPBS) and multiple algorithmic extensions for PBS to lower the computational runtime especially for MAPF instances with both many agents and obstacles. Overall, the PBS-family is regarded as one of the leading MAPF algorithms, and was thus used in several recent studies [113], [114]. Its algorithmic properties and the concept of reasoning with priorities make it particularly well suited for generating conflict-free trajectories in ASM Ops.

As general approach to decrease execution times of MAPF algorithms, Silver [115] introduced the idea of windowing the search, which is also referred to as bounded-horizon planning in other articles [116]. Using a windowed approach, conflict resolution is done up to a pre-defined time horizon w, while conflicts beyond that limit are ignored. This decreases the search space and thus the required runtime of the path planning algorithm.

In previous research, some of the assumptions of classical MAPF mentioned above were relaxed. As examples, studies explored the online MAPF variant requiring frequent replanning [117], the MAPFR problem for continuous, non-uniform edge weights [118], MAPF instances with delay probabilities [101], or robustness in planning [119]. Li, Surynek, Felner *et al.* [111] explored MAPF instances with Large Agents (LA-MAPF), in which agents occupy not one but a set of points in a graph at any time step. New conflict types are necessary to coordinate their movements, increasing the number of possible collisions. Since MAPF plans can usually not be directly executed by real-world robots [120], Hoenig, Kumar, Cohen *et al.* [121] proposed MAPF-POST which uses a temporal network to post-process MAPF solutions so that non-holonomic robots can execute the plans with guaranteed safety distances between them.

Nonetheless, the strong assumptions to solve such MAPF instances do often not apply to real-world problems [94]. Dependent on the application, continuous time and space dimensions as well as non-holonomic, heterogeneous agent properties may need to be planned for. The combination of these is often referred to as multi-agent or multi-robot motion planning (MAMP or MRMP). The latter usually addresses open spaces requiring roadmaps to be generated through e.g. sampling-based techniques [122]. However, the taxiway layouts of airports do not resemble open spaces, but rather consist of sparsely located intersections connected by roads in a continuous, two-dimensional Euclidean environment. Sampling-based techniques are thus not applicable to coordinate the agents in ASM Ops. Instead, the taxiway network can be transformed into a graph using the physical coordinates of intersections as nodes and the length of the interconnecting taxiways as edge weights, leading to a representation in continuous space. Alternatively, one study transformed the taxiway network using a grid-based layout with obstacles [6]. For both layout transformations, the above mentioned two-level search-based solvers can be applied when coupled with a low-level algorithm that can deal with the continuous dimensions and non-holonomic agent properties.

Safe Interval Path Planning (SIPP) Algorithms

As key advancement for single-agent path planning, Phillips and Likhachev [60] introduced the Safe Interval Path Planning (SIPP) algorithm to deal with dynamic obstacles. SIPP is based on the idea of aggregating time steps into time intervals: the duration that a moving obstacle occupies a graph location is defined as unsafe interval (USI), while the time between two USIs is denoted as safe interval (SI). This representation of the time dimension reduces the amount of potential states at every graph location and therefore the search space. Besides decreasing the runtime significantly, this also enables planning with continuous time and space, arbitrary motion durations, and different agent speeds.

For this reason, SIPP was subsequently used and further extended as single-agent planner [123]–[125], and as low-level solver for solving multi-agent instances [120], [126], [127]. To generalise SIPP to continuous time with large agents moving on 4-neighbour grids, Ma, Hönig, Kumar *et al.* [128] used SIPP with a reservation table (SIPPwRT), in which the shape of agents is expressed by a safety radius. Together with the velocities of two agents, but without accounting for acceleration and deceleration rates, they then compute time offsets to set safe intervals for lower prioritised agents. Li, Chen, Harabor *et al.* [129] generalised SIPP to handle soft obstacles, called SIPP with Soft Constraints (SIPPS). Soft obstacles mark the pending conflicts with the plans of other agents that are not yet set as

hard constraints. These are tracked in SIPPS to minimise the number of unresolved conflicts when creating a new plan. The authors used SIPPS in their proposed algorithm MAPF-LNS2, which starts with an invalid MAPF solution still containing unresolved conflicts. The remaining count of collisions is then gradually minimised by repeatedly replanning for a subset of agents. MAPF-LNS2 thus lowers the remaining collisions within a defined runtime limit, or improves the initial solution to be conflict-free.

Nonetheless, Ali and Yakovlev [130] argued that all previous work on SIPP does not account for the kinodynamic constraints of agents prohibiting them to stop instantaneously. They show that these agent properties make vanilla SIPP and its variants incomplete. To address this issue, they proposed SIPP with (wait) interval projection (SIPP-IP) and proved that it is both complete and optimal. In SIPP-IP, each state S has an associated non-overlapping, projected safe interval representing the time interval that can be safely reached from the safe interval at a previous wait-location in the path to S. However, their idea only holds for states that originated from a wait-location, limiting its applicability to real-world problems.

Summarising the Applicability of Existing Path Planning Algorithms for ASM Ops

Table 2.1 summarises the properties of the algorithms reviewed above, in line with the requirements defined in Section 2.2. While MILP and CBS solvers provide optimal solutions, they are intractable to plan for many agents in a complex operational environment within a reasonable time. Machine-learning/reinforcement-learning (ML/RL) approaches are deemed not yet capable to handle the intrinsic complexities of realistic representations of ASM Ops. MAPF-LNS2 aims to iteratively improve invalid solutions with remaining conflicts, is typically applied to problem instances involving thousands of agents, and was so far only tested on uniform grid environments. All these properties of MAPF-LNS2 make it less suitable for the operations considered in this thesis. In contrast, prioritised planning algorithms such as PBS and FCFS are well aligned with conflict resolution strategies in ASM Ops, in which priorities are commonly assigned. While FCFS likely provides inefficient solutions, PBS-based algorithms are regarded to deliver near-optimal solutions, and to scale well also in complex environments. We thus deem PBS-like algorithms to be most suitable to carry out multi-agent coordination in ASM Ops models.

In addition to prioritised planning, the efficient generation of conflict-free, 4D agent trajectories is essential for obtaining high-quality solutions within reasonable runtimes. To this end, the SIPP-family offers a strong basis for single-agent planning. However, SIPP must be tailored to the operational context of ASM Ops and extended to support motions with finite acceleration and non-zero initial speeds.

2.4 Addressing the Research Gaps

The reviewed literature showed that most existing models of ASM Ops are geared towards studying either an isolated part of the system or one specific operational concept, mainly based on current operations. Such models are thus inept to study the potentially wide range of ConOps that results from new technological solutions as well as future operational and environmental requirements, let alone to compare the efficacy of different approaches with each other on a detailed level.

Table 2.1: Algorithmic properties of route optimisation techniques as backbone for a high-fidelity system model of ASM Ops

Criterion	MILP	FCFS	ML/RL	CBS	PBS	MAPF-LNS2
planning scheme	single-stage	sequential	varies	best-first search	depth-first search	iterative refinement
low-level solver	MILP solver	A*, SIPP	learned policy	A*, SIPP	A*, SIPP	SIPPS
conflict handling	predefined constraints	sequential	learned avoidance	constraint-tree	priority-tree	via sub-groups with priorities
search space	very large	small	small after (extensive) learning	very large	medium	medium
solution quality	optimal	suboptimal (mostly low)	depends on learning	optimal	near-optimal	anytime improvement
safety guarantees	yes	yes	?	yes	yes	conflicts may remain
scalability	low	high	high (post-training)	low	medium	high
real-time use	no	yes	yes	no	potentially	potentially
typical environment	graph-based	varying	open spaces	varying	varying	uniform grid

To address the research gap, a generalised multi-agent system architecture is proposed for studying next-generation concepts of airport surface movement operations. The architecture has a flexible, hierarchical-distributed structure, taking the defined requirements from Section 2.2 as well as the outlined advantages and limitations of both centralised and distributed systems into account. Although highly distributed systems are conceivable, the stringent safety standards (Requirement 3) necessitate a form of operational oversight, thereby motivating the adoption of an underlying hierarchy in the generalised architecture to coordinate the operations.

We use the paradigm of multi-agent systems due to their demonstrated suitability to represent real-world systems, as shown across various applications [59], and their inherent modularity, flexibility, and expressiveness. Both heterogeneous agent properties and behaviour as well as randomness can be integrated into an agent-based model [46], facilitating the design of detailed models to address the identified modelling requirements. In the context of ASM Ops, multi-agent system modelling thus allows to organise the complex and intertwined operational tasks into multiple smaller tasks and assign these to various agents with different roles.

Based on the generalised architecture, specific model instances can be defined to study new concepts of operations characterised across various dimensions such as procedural aspects (e.g. selected taxiing techniques, effect of and compliance with existing/new procedures), degree of centralisation (from centralised control to distributed control with enacted system-wide constraints), and level of automation per task (allocated to a human vs. an automated system), requiring different coordination schemas. As a consequence, the generalised architecture offers design freedom to develop more elaborate and tailored models, while allowing to reuse the overall simulation environment as well as model components. Moreover, it enables comparing system-level performance (see Requirement 5) across varying ConOps, and to iteratively refine the operational concepts in fast-time simulations and human-in-the-loop experiments with stakeholders.

> This chapter summarised the operational background of ASM Ops and established the modelling requirements underlying the models, methods, and tools developed in this thesis (addressing RQ 1). In the next chapter, the proposed generalised architecture for multi-agent system models to study future ConOps of ASM Ops (RQ 2) is outlined, constituted by four agent categories as well as environmental objects. Moreover, one model instance designed for fully-automated ASM Ops with centralised path planning is introduced, which forms the basis for the MAS models used in the case studies reported in Chapters 5 to 9.

3

Multi-Agent System Models to Study Next-Generation Airport Surface Movement Operations

As outlined in Chapter 2, coordinating the movements of aircraft along the surface of busy airports is a complex task, requiring coordination among various stakeholders to ensure safe and efficient operations. In response to the interrelated challenges of ever-increasing demand, emission reduction, and sustaining safety levels, technological advances are emerging which, in turn, necessitate novel concepts of operations (ConOps). To this end, diverse operational concepts with varying technological aspects, level of automation, and degree of control centralisation could potentially offer a solution. However, before such innovative ConOps are matured to be deployable at real-world airports, extensive evaluation based on computational modelling is necessary.

To enable such detailed modelling and analysis of the large variety of concepts for next-generation airport surface movement operations, this chapter proposes a generalised architecture based on the hierarchical-distributed multi-agent system modelling paradigm (Section 3.1). We illustrate the generalised architecture by providing a specific model instance for a ConOps based on centralised planning and distributed, fully-automated control, outlined in Section 3.2.

3

This chapter is based on the following publication:

M. von der Burg, J. Kamphof, J. Soomers and A. Sharpanskykh, 'Towards next-generation airport surface movement operations using hierarchical-distributed multi-agent planning', *Transportation Research Part C: Emerging Technologies*, vol. 186, p. 105 583, 2026, ISSN: 0968-090X. DOI: 10.1016/j.trc.2026.105583.

ASM Ops tasks can be divided in several categories: planning (strategic and long-term decisions), scheduling (tactical decisions), routing (tactical path planning), providing guidance (operational path planning), and steering an aircraft or ground vehicle (operational plan execution). The progression from strategic planning to operational control reflects a hierarchical structure among the tasks. At the same time, the strict safety standards of ASM Ops (Requirement 3) necessitate coordinated oversight across all operational levels. Together, these aspects motivate the choice of a hierarchical-distributed system architecture to manage the complexity and to ensure safety as well as efficiency.

3.1 Generalised Hierarchical-Distributed Control Architecture

To enable studying diverse concepts of next-generation ASM Ops, we thus propose a generalised multi-agent system architecture with a hierarchical-distributed structure. It comprises multiple levels of control through distinctive agent categories with potentially limited system knowledge, scope of control, and access to information. We define the agent categories in analogy to the hierarchy of ASM Ops tasks mentioned above. Illustrated by Figure 3.1, we outline the different agent categories and environmental objects constituting the generalised architecture in the following.

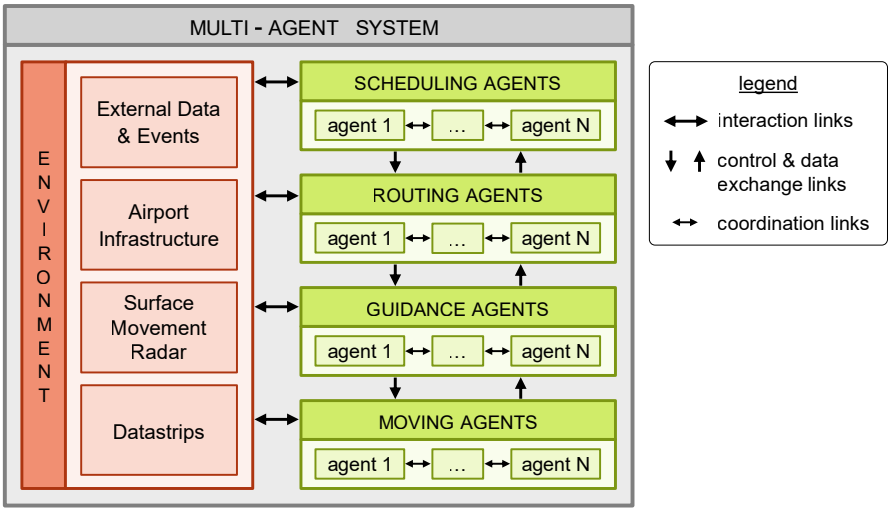


Figure 3.1: Generalised hierarchical-distributed multi-agent system modelling architecture to study diverse concepts for next-generation airport surface movement operations

In the generalised architecture, agents are associated with ASM Ops tasks to exert control and exchange data in line with the hierarchical placement of the agent categories. Within each category, multiple agents may operate and engage in centralised and/or distributed control mechanisms, informed by communication within their level as well as across hierarchical layers. Furthermore, all agents can potentially interact with environmental objects as part of their perception and decision-making processes.

The agent category for strategic planning tasks such as defining the flight plans is deliberately disregarded, since such tasks are carried out long before the actual ASM Ops take place. They are placed in the environmental object of **External Data & Events** along with other information from stakeholders outside the scope of ASM Ops. These include dynamic system inputs such as weather forecasts and weather events, as well as CTOT-slots assigned by the network manager, among others. The environmental object called **Airport Infrastructure** constitutes the taxiway network as well as other infrastructure such as remote parking locations, recharging stations, or ground vehicle depots to enact the specific ConOps. It is represented by a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$, where edges $\epsilon \in \mathcal{E}$ represent taxiway centrelines and nodes $\nu \in \mathcal{V}$ correspond to merging points between taxiways as well as special points such as stands or tug coupling/decoupling locations. The edges are constructed using Bézier curves with non-uniform lengths l_ϵ, enabling a realistic and flexible representation of airport layouts. To reflect operational constraints, additional edge attributes such as speed limits $v_{\text{lim},\epsilon}$ based on the radii of curvature can be assigned to the edges. Moreover, the environment provides the **Surface Movement Radar** as means of surface surveillance to provide information on e.g. cooperative ground movements (i.e. those that allow to be detected). In analogy to present-day flight strips, the **Datastrips** store any information pertaining to ground movements such as schedule or route data. Dependent on the modelled ConOps, different agents can access, supplement, or store data in the environmental objects.

The **Scheduling Agents** $a_i \in \mathcal{A}^{\text{SKL}}$ are responsible to manage all airport schedules such as flight schedules, runway configurations, or task allocations of ground vehicles. Furthermore, they issue constraints pertaining to the Airport Infrastructure, e.g. due to the runways in use or temporarily unavailable taxiway segments. Both schedules and layout constraints are shared with the **Routing Agents** $a_i \in \mathcal{A}^{\text{RTN}}$. Their task is to conduct path planning for individual ground movements, and may include coordinating plans of multiple movements, dependent on the underlying ConOps. Based on the available route information, the **Guidance Agents** $a_i \in \mathcal{A}^{\text{GDS}}$ can further refine the routing plans, and exert their control by providing e.g. clearances or route instructions to the **Moving Agents** $a_i \in \mathcal{A}^{\text{MOV}}$ who are responsible to execute the operational plans by safely steering the aircraft or ground vehicles.

Besides regular taxiing, other operational processes such as inbound/outbound holding, pushback/push-pull manoeuvres, or coupling/decoupling of tugs (see Section 2.1) must be executed, accounting for dependencies between tasks and time constraints (Requirement 2). To this end, we represent the tasks as basic activities and their interdependencies as activity sequences per Moving Agent, defined as:

Definition 1 Activities and Activity Sequence of Moving Agents

Each activity is represented as a tuple $\alpha = (\tau, \theta)$, with the activity type $\tau \in \mathcal{T} = \{\texttt{GoTo}, \texttt{Follow}, \texttt{Wait}\}$ and set of parameters θ:

- for `GoTo` activities, θ is defined by a set of potential goal locations $G \subset V$;
- for `Follow` activities, θ is defined by a predefined path $P = \langle \epsilon_1, \epsilon_2, \ldots, \epsilon_m \rangle$ to be followed;
- for `Wait` activities, θ is defined by a wait duration $\Delta t \in \mathbb{R}_{>0}$.

Each Moving Agent $a_i \in \mathcal{A}^{\text{MOV}}$ is assigned an *activity sequence*:

$$\mathcal{S}_i = \langle\, \alpha_1\,,\, \alpha_2\,,\, \ldots\,,\, \alpha_n \,\rangle_i\,, \quad \alpha_{k,i} = (\tau, \theta)_{k,i} \tag{3.1}$$

which must be executed in the given order. Each agent's first activity $\alpha_{1,i}$ must begin at a designated start location $\nu_{\text{s},i} \in \mathcal{V}$, specified by the agent's initial position within the airport layout.

As example, the activity sequence of an outbound aircraft requiring a push-pull manoeuvre can be expressed as sequence of activity types

$$\langle\, \texttt{Follow}\,,\, \texttt{Wait}\,,\, \texttt{Follow}\,,\, \texttt{Wait}\,,\, \texttt{GoTo} \,\rangle$$

with the corresponding parameters to follow a standard pushback-and-pull path with an intermittent stop to switch movement direction towards the location at which the pushback tractor is decoupled, and taxiing towards one of the runway entry points. Moreover, the definition of activities allows to express activity sequences for any technological concept: both multi-engine taxiing as well as tug-enabled taxiing including inbound/outbound holding as well as adherence to CTOT-slots can be modelled [131], [132]. Likewise, movements of aircraft with novel, more sustainable propulsion systems such as hydrogen-powered or battery-powered aircraft can be represented as a combination of `GoTo`, `Follow`, and `Wait` activities.

Based on the generalised architecture, a large variety of models can be instantiated that are tailored to a specific ConOps, while retaining the overall system structure. To this end, the operational tasks can be re-allocated to different agents, and their decision-making abilities can be customised with different coordination mechanisms. For example, the degree of control centralisation could differ between model instances: in a centralised setting, all movements could be coordinated up front by one or multiple Routing Agents, based on schedules issued by the Scheduling Agents. This would minimise the guidance and control tasks carried out by Guidance Agents and Moving Agents during plan execution. In contrast, in a model with distributed path planning, the Moving Agents could reason themselves about each required action in their activity sequence. They could then negotiate with other agents to resolve conflicting situations based on their individual circumstances and operational needs. In such a setting, the Routing Agents and Guidance Agents could act as intermediaries, set traffic rules, or issue special priorities, in accordance with the schedules provided by the Scheduling Agents. Likewise, the generalised architecture allows to define ConOps with differing levels of automation: each ASM Ops task can either be allocated to a human, an automated system, or split into a collaborative effort between humans and automation. In line with these examples, we elaborate one model instance representing a ConOps for fully-automated ASM Ops with centralised path planning in the following subsection.

To summarise, the proposed generalised modelling architecture fulfils Requirements 1 and 2 through its hierarchical-distributed architecture as well as the allocation of tasks to the corresponding agent categories, and Requirement 4 with the environment definitions. The remaining requirements comprising safety aspects, operational goals, dealing with uncertainties, and the algorithmic properties of the utilised coordination mechanisms must be ensured when instantiating a MAS model based on a specific ConOps.

3.2 Multi-Agent System Model for Fully-Automated Operations with Centralised Planning

For the remainder of this thesis, we consider an operational concept for fully-automated operations with centralised planning which was modelled based on the generalised architecture presented in the previous subsection. Although this ConOps poses significant implementation challenges in real-world settings, it is expected to offer the highest operational efficiency. In turn, it can serve as a suitable baseline for future comparisons with alternative operational concepts. In the following, we first present an overview of the multi-agent model structure and its important properties, with Figure 3.2 illustrating the MAS model. The subsequent sections then outline the complete specifications of the model components.

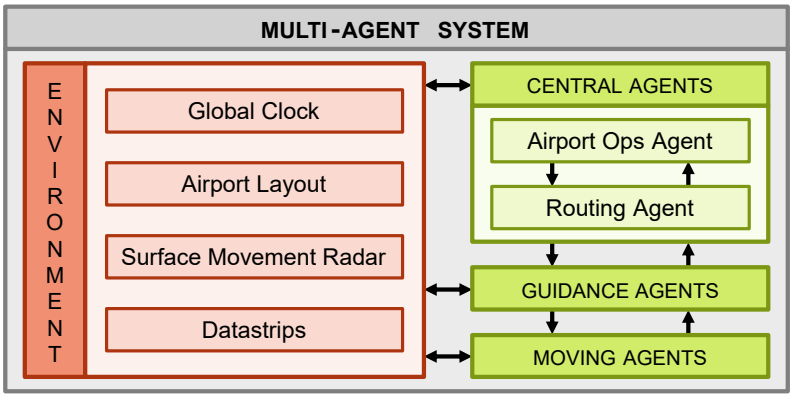


Figure 3.2: Instantiation of multi-agent system model for fully-automated airport surface movement operations (ASM Ops) with centralised planning and distributed plan execution

As sole Scheduling Agent, the centralised **Airport Ops Agent** a_{APO} schedules the flights and tasks of ground vehicles, and issues layout constraints on the **Airport Infrastructure**. It shares both schedules and constraints with the single, centralised **Routing Agent** a_{RTN}. Its task is to conduct centralised path planning to provide conflict-free 4D trajectories for all ground movements, formalised as routing plans:

Definition 2 Routing Plans

Based on an agent's activity sequence $\mathcal{S}_i$, path $\mathcal{P}_i = \langle \nu_1 , \nu_2 , \cdots , \nu_m \rangle_i$ for agent $a_i \in \mathcal{A}^{\text{MOV}}$ is defined. The path forms the basis for the agent's routing plan Π_i defined as:

$$\Pi_i = \langle \pi_i(\nu_1) , \pi_i(\nu_2) , \cdots , \pi_i(\nu_m) \rangle , \quad \pi_i(\nu_k) = (t_{\nu_k} , d_{\nu_k} , v_{\nu_k} , \alpha_{\nu_k})_i \tag{3.2}$$

where each tuple specifies per node ν_k the respective time t_{ν_k}, travelled distance d_{ν_k}, speed v_{ν_k}, and activity α_{ν_k}.

The Routing Agent shares the routing plans with the **Guidance Agents** $a_i \in \mathcal{A}^{\text{GDS}}$ that are positioned at every junction in the taxiway network. Each Guidance Agent

controls those agents that are travelling towards its location by sending them instructions based on the received trajectories. To ensure safe operations, the Guidance Agents monitor the ground movements, and locally adjust the planned trajectories, if necessary (Requirement 3). To this end, the Guidance Agents use the **Surface Movement Radar** (SMR) as one of the environmental services to obtain the position and speed of an agent. Once an agent a_i has passed the Guidance Agent's location, it hands over the control to the next Guidance Agent along the agent's path by transferring the **Datastrip** $\mathcal{D}_i$ associated with that agent. $\mathcal{D}_i$ stores all data pertaining to the agent's routing plan, its activity sequence $\mathcal{S}_i$, and the current SMR data of a_i. The **Moving Agents** $a_i \in \mathcal{A}^{\text{MOV}}$ represent the aircraft pilots and tug drivers, and are modelled to be fully cooperative. We thus assume that they execute the received instructions to the best of their capabilities when taxiing over the airport surface. To address Requirement 3, all Moving Agents must keep a minimum safety distance to each other, expressed as a circular safety zone $\mathcal{Z}_i$ around the agents. Since pilots and auto-pilots can only follow planned trajectories with limited precision (see Requirement 6), the safety zones include a positioning tolerance δ_{pos} as allowable deviation from the intended trajectories.

In this MAS instance, while any perturbations can in general be communicated upstream, the Moving Agents as well as the Guidance Agents rely strongly on the centralised routing carried out by the Routing Agent. Therefore, to meet the operational goals outlined in Requirement 5, the Routing Agent shall determine conflict-free 4D trajectories that minimise the total taxi time of all Moving Agents. As connected objectives, the resulting trajectories shall also reduce the total taxi distance and mitigate any stop-and-go movements. Moreover, to keep the centralised planning tractable (Requirement 7) and to account for operational uncertainties (Requirement 6), we base the decision-logic of the Routing Agent on the concept of windowed prioritised planning:

Definition 3 Windowed Prioritised Planning

Centralised path planning is done in planning rounds. Each planning round k starts at time $t_{\text{p},k} \geq 0$. In each round, the Routing Agent computes or updates the routing plans Π_i of those agents $a_i \in \mathcal{A}^{\text{MOV}}$ that are taxiing within the planning window $[t_{\text{p},k}, t_{\text{p},k} + w_{\text{p}}]$ where $w_{\text{p}} > 0$ denotes the fixed span of the planning window. In turn, the Routing Agent plans periodically within the maximum replanning period $h_{\text{p}} > 0$, such that $t_{\text{p},k+1} \leq t_{\text{p},k} + h_{\text{p}}$.
To deconflict concurrent agent trajectories, the Routing Agent assigns priorities between Moving Agents $a_i, a_j \in \mathcal{A}^{\text{MOV}}$, with $a_i \prec a_j$ indicating that agent a_i has higher priority than agent a_j. To ensure that a_j gives way to a_i, the routing plan Π_j of agent a_j must not violate the safety zone $\mathcal{Z}(\Pi_i)$ around the routing plan Π_i of agent a_i within the planning window. Conflicts outside the current planning window can be safely disregarded as these are accounted for in subsequent planning rounds.

3.2.1 Specification of Environment

The Environment comprises a Global Clock to synchronise the time between all agents, the Airport Layout representing the taxiway network, the Surface Movement Radar to track Moving Agents on the airport surface, and the Datastrips associated with every

Moving Agent. The latter is used to pass control between Guidance Agents and to store the necessary operational as well as analysis data.

The taxiway network is represented by a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with nodes $\nu \in \mathcal{V}$ and directional edges $\epsilon \in \mathcal{E}$. An exemplary Airport Layout is shown in Figure 3.3. Tug decoupling locations (orange nodes) are specific to each runway (grey edges) and are used for tugs to couple to or decouple from aircraft that taxi with tug-enabled taxiing. Each bidirectional road segment between two nodes is constructed from two unidirectional edges that connect the nodes. Taxiway edges (black) as well as service road edges (blue) can be obtained from the actual locations of these taxiways at an airport, such as Amsterdam Airport Schiphol. Ramp nodes (green) represent the aircraft parking positions. The airport layout graph is fully accessible to the Routing Agent and remains static throughout the simulation. However, edges can be constrained to prohibit vehicles to be routed over these. Such layout constraints are set by the Airport Ops Agent, as outlined further below.

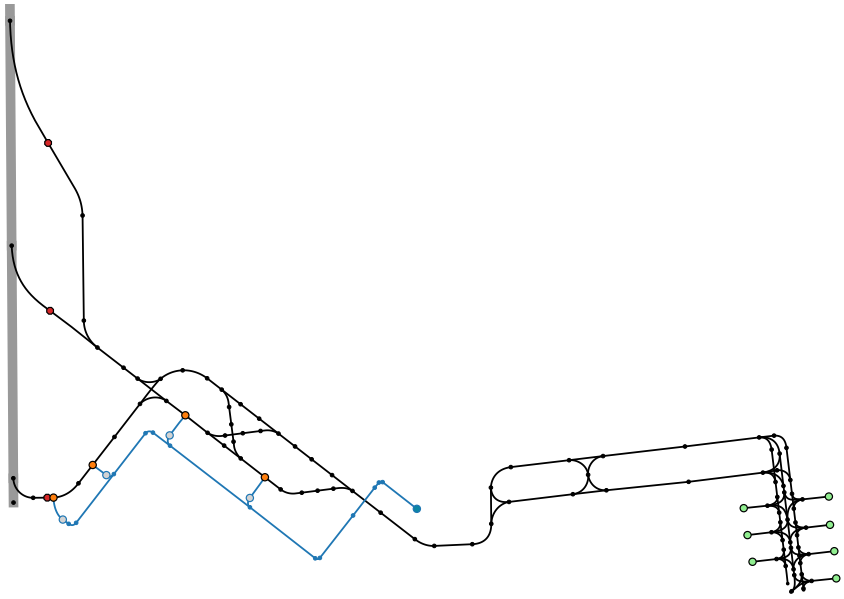


Figure 3.3: Representation of taxiway networks as graph: taxiway centrelines as well as pushback paths are shown as black, runways as grey, and service roads as blue edges; small nodes represent taxiway intersections (black) or service road intersections (blue); large nodes indicate aircraft ramps (green), stopbars (red), decoupling locations (orange), or their associated all-clear points (grey)

3.2.2 Specification of Airport Ops Agent

The Airport Ops Agent manages all information that influences the airport surface movement operations as a whole: it schedules the flights and tasks of ground vehicles, handles necessary changes of these, and determines the active runways, i.e. the runway mode of operation (RMO), for the next 15 min. Furthermore, it notifies the Routing Agent of any temporarily unavailable taxiway segments. The Airport Ops Agent has the following properties:

P1. Receive Request Property: both Routing Agent and Guidance Agents can send information that affect the operations to the Airport Ops Agent, which then acts accordingly. For instance, a Guidance Agent could report a blocked taxiway segment, or the Routing Agent may request a change in the schedules to allow for conflict-free routing.

P2. Update Schedule Entry Property: the Airport Ops Agent schedules all flights and tasks of tugs and updates the entry in the respective schedule accordingly, including any necessary changes that arise during the operations (see P1). The updated schedules as well as the time point of the earliest change are shared with the Routing Agent.

P3. Update RMO Property: at each time point t, the Airport Operations Agent checks whether the active runways have to change. If so, it adjusts the blockage intervals from $t + 15$ min onwards for all edges affected by the RMO change. It then shares the new set of layout constraints with the Routing Agent.

P4. Request Replanning Property: if the time point of earliest change in either schedules or RMO falls within the planning window w_{p}, the Airport Ops Agent sends a request for immediate re-planning of the routes to the Routing Agent.

3.2.3 Specification of Routing Agent

The role of the Routing Agent is to perform the central route planning. To do so, it gathers the updated schedule data, layout constraints, along with local information from the Guidance Agents, and converts this into route parameters per agent. Then, it computes a plan for all vehicles within the planning window w_{p} using the AS-MAMP algorithm specified in Chapter 4, and shares the updated routing plans with the Guidance Agents. The properties of the Routing Agent are as follows:

P1. Receive Request Property: the Routing Agent can receive a re-planning request from both Airport Ops Agent and Guidance Agents. The former triggers re-planning if changes to the schedules or the layout constraints become effective within the planning window, while the Guidance Agents do so when the deviation between planned and executed route of a Moving Agent exceeds time thresholds. The request is stored internally and further handled in P2.

P2. Check Replanning Property: at each time point t the Routing Agent checks whether a re-planning request was made or the internal re-planning timer elapsed. If so, the timer is reset, and the routes are re-planned by executing P3, P6, and P7.

P3. Update Constraints for Route Planning Property: the layout constraints from the Airport Ops Agent are gathered and stored internally. These change foremost when the RMO switches, and allow the Routing Agent to anticipate on upcoming taxiway closures or re-openings corresponding to the altered RMO.

P4. Handle Time Constraints of Moving Agent Property: to address potential time constraints of a Moving Agent (pertaining to Requirement 2), the Routing Agent assesses whether these are achievable and whether remote and/or stand holding is necessary, which influences the activity sequence of the Moving Agent that it defines through P5. Furthermore, it sets operational constraints pertaining to the time constraints, to account for these during path planning done by its P7.

P5. Define Activity Sequence of Moving Agent Property: to include the various parts of the route of a Moving Agent such as push-pull manoeuvres, coupling/decoupling of tugs, or holding in path planning, the Routing Agent converts these into one of the following three activities, in accordance with Definition 1:

- it assigns a `GoTo` activity when a Moving Agent shall taxi from one point to another point at the airport (potentially provided as a set of selected points), e.g. for regular taxiing;
- it assigns a `Follow` activity when a Moving Agent shall sequentially follow a predefined list of path segments, e.g. as part of pushback and push-pull manoeuvres;
- it assigns a `Wait` activity to prescribe a waiting location at which the Moving Agent shall wait for a specified duration, e.g. for coupling to or decoupling from a tug.

Using a combination of these, the Routing Agent defines an activity sequence according to the schedule of a Moving Agent, which has to be executed in the respective order.

P6. Update Route Parameters of Moving Agents Property: per Moving Agent that is scheduled to taxi within the planning window, the Routing Agent collects all necessary parameters to plan its route. This encompasses the kinodynamic representation according to the vehicle type of the Moving Agent, the activity sequence derived from its schedule (shared by the Airport Ops Agent), and in case the Moving Agent is already taxiing also the latest execution data (shared by the controlling Guidance Agent).

P7. Do Route Planning Property: based on the collected information, the Routing Agent updates the routing plans of all agents. Following Definition 3, it generates trajectories that are conflict-free within its planning horizon by assigning priorities between agents. Its underlying decision logic is expressed by the AS-MAMP algorithm (see Chapter 4). The Routing Agent shares the resulting routing plans with the Guidance Agents.

3.2.4 Specification of Guidance Agents

Each Guidance Agent controls the Moving Agents that are travelling towards its location and hands over the control to the next Guidance Agent once a Moving Agent has passed its location. Based on the generated routing plans, it gives instructions to the Moving Agents, and monitors their execution. It acquires the necessary position and speed data from the radar. When the executed route deviates from the plan, a Guidance Agent can adjust the route locally to minimise the deviations. If the deviation to the planned route exceeds time thresholds, it can request re-planning of the routes. The Guidance Agents thus act as link between the centralised planning and the local operations. They have the following properties:

P1. Receive Updated Routing Plan Property: once a new routing plan is issued by the Routing Agent, the Guidance Agent updates the planned routes of the Moving Agent under its control and generates new corresponding instructions by executing P2.

P2. Send Instructions to Moving Agent Property: as one-way means of communication, the Guidance Agents send instructions for executing the next part of the planned

routes to the Moving Agents. This property is thus executed whenever a Moving Agent has to change its speed or heading within the pre-defined execution time window w_{exec}. Furthermore, coupling/decoupling as well as engine-start instructions are sent.

P3. Monitor Execution of Planned Routes Property: at each time point t, each Guidance Agent obtains the positions, headings, and speeds of all Moving Agents under its control from the radar. The agent then estimates the time and speed at which each Moving Agent will pass its location, and shares these with the Routing Agent. The Guidance Agents also use the radar data to monitor that their instructions are executed according to the planned routes. If deviations arise, they can primarily re-instruct the Moving Agents to close the gap. As last resort, they can execute P4 to trigger centralised re-planning of the routes.

P4. Request Replanning Property: with this property, the Guidance Agents request re-planning from the Routing Agent iff the deviation between a planned and executed route of a Moving Agent exceeds pre-defined thresholds.

P5. Communicate with Moving Agent Property: the Guidance Agents communicate with the Moving Agents under their control whenever required. Thus, they can pass on information from or about a Moving Agent to the centralised agents.

P6. Handover Control Property: whenever a Moving Agent has passed the location of a Guidance Agent, it transfers the respective Datastrip and thus the control responsibility to the closest next Guidance Agent on the planned route of the Moving Agent. Once a Moving Agent has reached its goal, the DS is placed in storage, and can be utilised for post-operation analyses.

3.2.5 Specification of Moving Agents

Moving Agents represent either an aircraft or a tug. They are modelled to have a circular size with a shape diameter corresponding to their respective type. All aircraft are categorised according to the six size-types of the ICAO aerodrome reference codes [69], as listed in Table 3.1. Tugs are modelled with a diameter of 12 m based on the reference TaxiBot system [12].

Table 3.1: Categorisation of aircraft sizes into ICAO-types

ICAO-type	shape diameter [m]	exemplary aircraft type
ICAO-A	12	Cirrus Vision SF50
ICAO-B	25	Cessna, Learjet
ICAO-C	40	A320, B737, EMB 170/190
ICAO-D	54	A300F, B767
ICAO-E	72	A350, B747, B787
ICAO-F	80	A380, B747-800F

When consecutive aircraft take off from a runway, the Moving Agents have to keep a time-based separation minima to ensure sufficient wake turbulence separation. The times are dependent on the type-combination of leading and following aircraft, as listed in Table 3.2. The values were mapped to the ICAO-types from Table 3.1 on the basis of the RECAT-EU time-based separation minima [133].

The role of the Moving Agents is to execute the instructions that they received from the Guidance Agents. They thus have the following properties:

P1. Execute Instructions Property: the Moving Agents execute the received instructions to the best of their capabilities to limit the amount of necessary re-instructions. The trajectories provided to the Moving Agents include an allowable positioning tolerance around the speed profile in analogy to the Ownship-display used in [90].

P2. Communicate with Guidance Agent Property: in any case, the Moving Agents can communicate with the controlling Guidance Agent to exchange status updates and relevant operational information. They thus notify the Guidance Agent whenever a process such as tug decoupling or engine-start is estimated to finish. As further example, they could notify the Guidance Agent that their preferred maximum speed is lower than expected, so that the routing plans can be adapted accordingly.

Table 3.2: Minimum separation times in [s] for departing aircraft to mitigate wake turbulence. The aircraft types correspond to those defined in Table 3.1

leading \ following	ICAO-A	ICAO-B	ICAO-C	ICAO-D	ICAO-E	ICAO-F
ICAO-A	80	60	60	60	60	60
ICAO-B	100	60	60	60	60	60
ICAO-C	120	60	60	60	60	60
ICAO-D	120	100	80	60	60	60
ICAO-E	140	120	100	60	60	60
ICAO-F	180	160	140	120	100	60

In this chapter, RQ 2 was addressed by presenting a generalised MAS architecture as well as a reference model instance for studying a concept of operations under fully automated control with centralised path planning. As the central enabling component of this concept, the centralised planning function is assigned to the Routing Agent. To represent the Routing Agent's decision logic and to address Requirement 7 from Chapter 2, the next chapter introduces the novel Multi-Agent Motion Planning on Airport Surfaces (AS-MAMP) algorithm, which computes conflict-free and efficient 4D trajectories for all ground movements (RQ 3).

4

The AS-MAMP Algorithm: Multi-Agent Motion Planning on Airport Surfaces

As essential part of the MAS model instance presented in Section 3.2, this chapter introduces the newly developed Multi-Agent Motion Planning on Airport Surfaces (AS-MAMP) algorithm. The AS-MAMP algorithm represents the decision-logic for coordinating all ground movements, associated with the centralised Routing Agent in that MAS model. It is a two-level solver that builds on prominent prioritised planners such as Priority-Based Search (PBS) and its variant Greedy PBS.

As current low-level solvers were insufficient to plan realistic 4D trajectories for ground movements, we developed the novel Safe Interval Motion Planning (SIMP) algorithm. By defining activity sequences per agent, SIMP plans trajectories across the operational processes during taxiing such as pushback, tug coupling/decoupling, and engine-start. The motions of aircraft and ground vehicles are based on finite acceleration, and avoid dynamic obstacles in continuous space and time, necessitating to define states with *feasible motion intervals*.

After introducing the AS-MAMP algorithm, defining the core concepts, and providing an in-depth description of the algorithmic components in this chapter, the algorithmic performance of AS-MAMP is evaluated in Chapter 5.

4

This chapter is based on the following publication:

M. von der Burg, J. Kamphof, J. Soomers and A. Sharpanskykh, 'Towards next-generation airport surface movement operations using hierarchical-distributed multi-agent planning', *Transportation Research Part C: Emerging Technologies*, vol. 186, p. 105 583, 2026, ISSN: 0968-090X. DOI: 10.1016/j.trc.2026.105583.

The Multi-Agent Motion Planning on Airport Surfaces (AS-MAMP) algorithm is a two-level solver that uses a high-level search for multi-agent coordination by iteratively updating the individual agent trajectories in its low-level search. Figure 4.1 illustrates the core structure of the AS-MAMP algorithm: it combines PBS variants (see Section 2.3.3) as high-level with the newly developed Safe Interval Motion Planning (SIMP) algorithm as low-level. Based on an initial set of agent trajectories, the loop comprising conflict detection, selection, and resolution is initiated. The search continues until a conflict-free solution is found, returning the coordinated agent trajectories as routing plans. We summarize the core algorithmic concepts of AS-MAMP and define key terms in the following. An in-depth description of AS-MAMP is provided in Section 4.3.

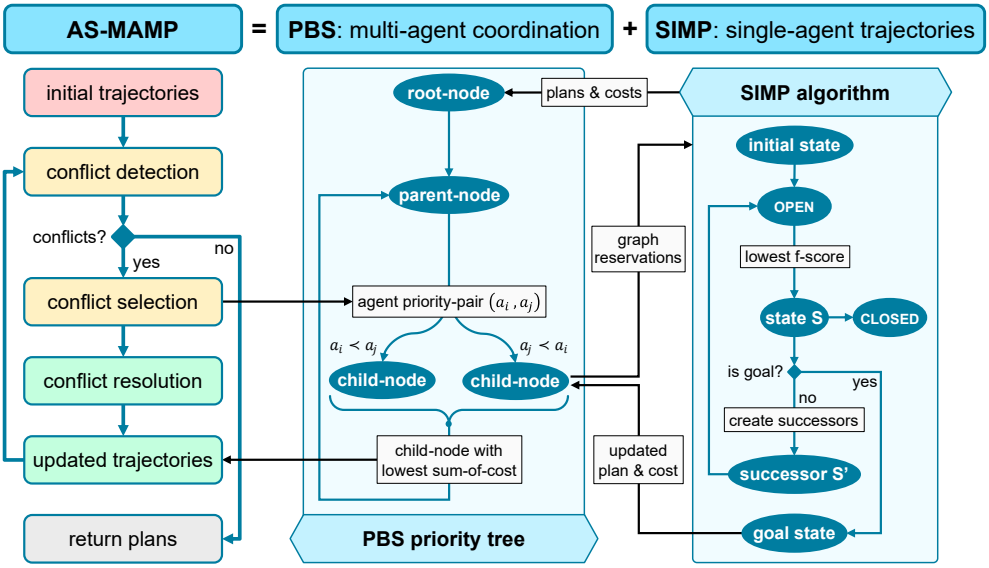


Figure 4.1: Overview of the Multi-Agent Motion Planning on Airport Surfaces (AS-MAMP) algorithm; the representation of the SIMP algorithm is simplified, see the pseudocode in Algorithms 3 and 4 for details

4.1 High-level Search in AS-MAMP

The PBS algorithm de-conflicts agents whose 4D trajectories intersect by constructing a priority ordering: it maintains a priority tree with each parent-node having up to two child-nodes to establish a priority-relation between two conflicting agents. Per child-node, a new priority-pair is added that constrains one of the two agents to avoid the other agent's path. Based on the sum-of-cost of the paths of all agents in each child node, PBS picks the priority order that results in the lowest overall cost, and expands this node next. In contrast to the original PBS algorithm, we define the cost of an agent trajectory as a linear combination of its taxi time and taxi distance:

Definition 4 Cost Function

The routing plan cost $c(\Pi_i)$ of agents $a_i \in \mathcal{A}^{\text{MOV}}$ is defined as sum of taxi time $t_{\text{taxi},i}$ and travelled distance $d_{\text{taxi},i}$, with a pre-defined constant $c_{\text{d},i}$ [s/m] translating d_{taxi} into a cost with unit time:

$$c(\Pi_i) = (t_{\text{taxi},i} + c_{\text{d},i} \cdot d_{\text{taxi},i}) \tag{4.1}$$

As defined in Definition 3, routing plans are deconflicted by accounting for the safety zones around agents. During replanning in AS-MAMP, higher-priority agents are treated as dynamic obstacles, with their respective shapes (see Section 4.3.3) temporarily blocking parts of the graph $\mathcal{G}$, added to the graph reservations set for an agent:

Definition 5 Graph Reservations

The 4D trajectory of an agent $a_i \in \mathcal{A}^{\text{MOV}}$ must satisfy all graph reservations $\mathcal{R}_i$ imposed on that agent. Each graph reservation $\varrho \in \mathcal{R}_i$ is defined as a tuple:

$$\varrho = (\lambda, [t_\text{s}, t_\text{e}], \kappa) \tag{4.2}$$

- where $\lambda \in \mathcal{L}$ is a graph location with
$$\mathcal{L} \subset \mathcal{V} \cup \mathcal{E} \cup \{(\epsilon, \delta_1, \delta_2) \mid \epsilon \in \mathcal{E},\ 0 \leq \delta_1 < \delta_2 \leq 1\}$$
denoting either a node, an edge, or a segment of an edge between relative distances δ_1 and δ_2;
- $[t_\text{s}, t_\text{e}] \subset \mathbb{R}_{\geq 0}$ is the time interval during which the location λ must not be traversed by agent a_i;
- $\kappa \in \mathcal{K}$ indicates the reservation type, e.g. dynamic obstacle, wake-turbulence separation, CTOT-slot, etc.

The graph reservations are also used to detect conflicts between agents:

Definition 6 Conflicting Agent Pair and Priority Relation

Let $a_i, a_j \in \mathcal{A}^{\text{MOV}}$ be two agents with respective sets of graph reservations $\mathcal{R}_i$ and $\mathcal{R}_j$. The pair (a_i, a_j) is called a *conflicting agent pair* if there exist reservations $\varrho_k \in \mathcal{R}_i$ and $\varrho_l \in \mathcal{R}_j$ for which locations and time intervals overlap, i.e.

$$\lambda_k \cap \lambda_l \neq \emptyset \quad \wedge \quad [t_\text{s}, t_\text{e}]_k \cap [t_\text{s}, t_\text{e}]_l \neq \emptyset \tag{4.3}$$

Following from Definition 3, the 4D trajectories of a_i and a_j are deconflicted by creating a new PBS child-node with an additional *priority relation* $\prec \in \mathcal{A}^{\text{MOV}} \times \mathcal{A}^{\text{MOV}}$ such that either $a_i \prec a_j$ or $a_j \prec a_i$. A new routing plan of the lower-priority agent is calculated that satisfies the updated set of graph reservations by invoking the low-level solver, which is further specified in the following subsection. The high-level search continues until a child-node is expanded without any collisions. The changes to the original PBS algorithm are outlined in Section 4.3.1, with the corresponding pseudocode provided in Algorithm 1. In Section 5.1.2, we benchmark the standard depth-first search in PBS against multiple variants to expand the priority-tree, including the first-come-first-served (FCFS) method.

4.2 Low-level Search in AS-MAMP

As low-level solver in AS-MAMP for calculating a single-agent 4D trajectory that respects all graph reservations defined for that agent, the Safe Interval Motion Planning (SIMP) algorithm was developed. Similar to SIPP (see Section 2.3.3), the SIMP algorithm reasons with safe time intervals, iteratively explores the search space with states defined on nodes, and uses motion principles for the mapping between states. However, to deal with the graph reservations in SIMP, safe intervals are not only defined on nodes but at any graph location $\lambda \in \mathcal{L}$ from Definition 5:

Definition 7 Unsafe and Safe Intervals

For an agent $a_i \in \mathcal{A}^{\text{MOV}}$, the time intervals of the graph reservations set for a location $\lambda \in \mathcal{L}$ are merged and ordered in time to form the *unsafe intervals* for a_i at λ:

$$\mathcal{U}_{i,\lambda} = \langle\, [t_\text{s}, t_\text{e}]_1, [t_\text{s}, t_\text{e}]_2, \cdots, [t_\text{s}, t_\text{e}]_n \,\rangle_{i,\lambda} \tag{4.4}$$

Then, the *safe intervals* denote the complement of $\mathcal{U}_{i,\lambda}$ as the sequence of time intervals accessible to a_i at λ:

$$\mathcal{I}_{i,\lambda} = \left([0, \infty] \setminus \bigcup \mathcal{U}_{i,\lambda} \right)^{\text{max}} = \langle\, [0, t_{1,\text{s}}], [t_{\text{e},1}, t_{\text{s},2}], \cdots, [t_{\text{e},n}, \infty] \,\rangle_{i,\lambda} \tag{4.5}$$

Moreover, in contrast to the instantaneously accelerated motions assumed in SIPP, the low-level search shall account for the kinodynamic agent constraints, requiring to define the following motion principles:

Definition 8 Motion Principles

Motions are approximated with piecewise-constant longitudinal acceleration, i.e. we apply the equations for rectilinear motions:

$$\text{final speed} \quad v_\text{f} = v_\text{i} + a\,t = \sqrt{v_\text{i}^2 + 2\,a\,d} \tag{4.6}$$

$$\text{traversal time} \quad t = \frac{2d}{v_\text{i} + v_\text{f}} = \frac{v_\text{f} - v_\text{i}}{a} \tag{4.7}$$

$$\text{travelled distance} \quad d = v_\text{i}\,t + \frac{1}{2}\,a\,t^2 = \frac{v_\text{i} + v_\text{f}}{2}\,t \tag{4.8}$$

with initial speed v_i and acceleration/deceleration a. In SIMP, feasible motions respecting the kinodynamic agent constraints as well as the graph reservations are calculated on-the-fly during the search, with the cost of a motion defined in analogy to Equation (4.1).

Per vehicle type, a maximum speed limit is set for both straight and curved edges. Furthermore, to avoid stop-and-go movements, the agents need to have a minimum speed with the exception of special locations in the taxiway network to fulfil `Wait` activities.

As consequence of the finite acceleration underlying the motion principles, agents are also not able to stop instantaneously. In turn, this violates a core assumption of SIPP and makes it incomplete, as noted by Ali and Yakovlev [130]. The minimum speed requirement

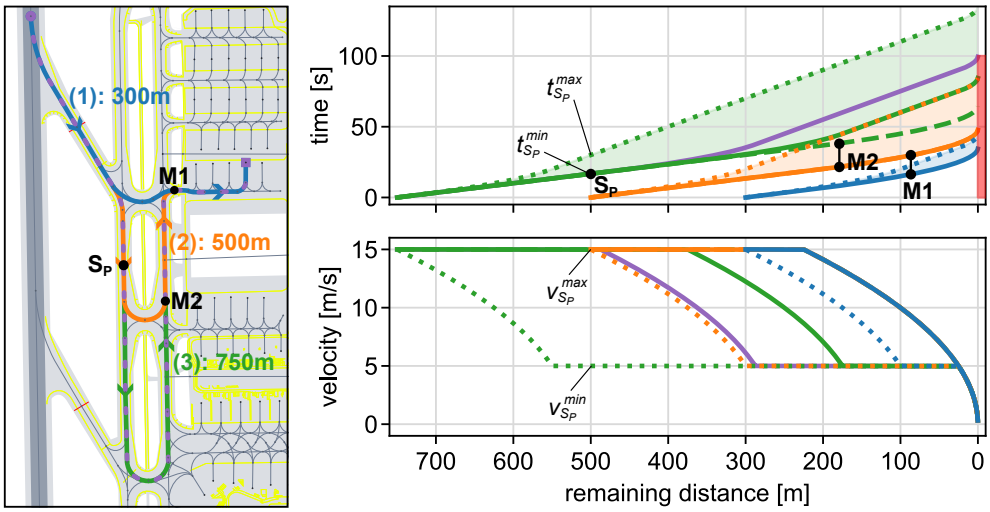


Figure 4.2: Exemplary trajectories through layout (left) and state space (right) to a stand that is occupied till $t = 100$ s resulting in a long goal constraint (red shade at 0 m)

at most edges and nodes further complicates this issue, illustrated in Figure 4.2: the designated stand of a landing aircraft is still occupied for a while, i.e. a long goal-constraint is placed on its goal-node, indicated by the red shade at 0 m in the upper right plot. Since the aircraft is neither allowed to stop along the direct route to its stand, nor travel slower than the minimum speed, it must take a detour. In the low-level search, states along path (1) in blue, then along path (2) in orange, and finally along path (3) in green would be created. However, in vanilla SIPP each node would have a single safe time interval. Thus, the state of the orange path at the merge location M1 would be discarded as its f-score is higher than the already existent state of the blue path at M1. Likewise, the state at M2 of the green path would not be added to its OPEN-list. As a consequence, the low-level search would fail to find a solution. To address this issue, we take the approach in [130] further, and combine an agent configuration with a feasible motion interval (FMI) to distinguish states in SIMP:

Definition 9 State Definition in SIMP

A state S is defined by an agent configuration S_{cfg} and a *feasible motion interval* (FMI). The configuration comprises the current node ν, next node ν' (as direction), speed limit v_{lim} (depending on the previous and next edge), and the current activity α_k from the agent's activity sequence $\mathcal{S}$:

$$S_{\text{cfg}} = (\nu\,,\,\nu'\,,\,v_{\text{lim}}\,,\,\alpha_k) \tag{4.9}$$

The corresponding FMI of state S is defined as:

$$S_{\text{FMI}} = (t_{\text{S,min}}\,,\,t_{\text{S,max}}\,,\,v_{\text{S,max}}\,,\,v_{\text{S,min}}) \tag{4.10}$$

with the quickest feasible motion arriving at $t_{\text{S,min}}$ with speed $v_{\text{S,max}}$, and the slowest feasible motion arriving at $t_{\text{S,max}}$ with $v_{\text{S,min}}$.

Given the minimally allowed speed $v_{\min}$,

$$v_{\min} \leq v_{\mathrm{S},\min} \leq v_{\mathrm{S},\max} \leq v_{\mathrm{S},\mathrm{lim}} \tag{4.11}$$

When another state with identical configuration exists, and their FMIs overlap, the FMIs are adjusted to be temporally disjunct: for two states S and S^* with the f-scores $f_{\mathrm{S}} < f_{\mathrm{S}^*}$, the FMI of S^* is adjusted so that $t_{\mathrm{S}^*,\min} \geq t_{\mathrm{S},\max}$ and $v_{\mathrm{S}^*,\max} \leq v_{\mathrm{S},\min}$. In case $t_{\mathrm{S}^*,\max} < t_{\mathrm{S},\max}$, the new state is discarded. If $f_{\mathrm{S}} = f_{\mathrm{S}^*}$, the state with the earlier arrival time is kept, and the FMI of the other state is adjusted accordingly.

In the following, we illustrate the use of FMIs with the example in Figure 4.2. In its upper right plot, the arrival time interval of state S_{P} is visualised by the shaded green area between the solid and dotted lines representing the earliest arrival time $t_{\mathrm{S_P},\min}$ and latest arrival time $t_{\mathrm{S_P},\max}$, respectively. Likewise, the lower right plot shows the speed range of S_{P} between $v_{\mathrm{S_P},\max} = 15\,\mathrm{m/s}$ (solid green line) and $v_{\mathrm{S_P},\min} = 5\,\mathrm{m/s}$ (dotted green line).

At the merge location M1, the FMI of the orange state does not overlap with that of the blue state, and is directly added to OPEN. At M2, the FMIs of the orange state $S_{\mathrm{o,M2}}$ and green state $S_{\mathrm{g,M2}}$ are respectively defined as:

$$S_{\mathrm{o,FMI}} = (t_{\mathrm{S},\min}\,,\, t_{\mathrm{S},\max}\,,\, v_{\mathrm{S},\max}\,,\, v_{\mathrm{S},\min})_{\mathrm{o}} = (23\,\mathrm{s}\,,\, 45\,\mathrm{s}\,,\, 13\,\mathrm{m/s}\,,\, 5\,\mathrm{m/s})$$
$$S_{\mathrm{g,FMI}} = (t_{\mathrm{S},\min}\,,\, t_{\mathrm{S},\max}\,,\, v_{\mathrm{S},\max}\,,\, v_{\mathrm{S},\min})_{\mathrm{g}} = (38\,\mathrm{s}\,,\, 95\,\mathrm{s}\,,\, 13\,\mathrm{m/s}\,,\, 5\,\mathrm{m/s})$$

which overlap on the interval between (38 s, 45 s). The g-score of the green state must be higher since its path is longer. Therefore, $f_{\mathrm{S_o,t=38s}} < f_{\mathrm{S_g,t=38s}}$. As a result, the motion of the green state must be adjusted to create a temporally disjunct FMI: its adjusted earliest arrival time $t_{\mathrm{S_g},\min} \cong t_{\mathrm{S_o},\max} = 38\,\mathrm{s}$ with a speed of $v_{\mathrm{S_g},\max} = v_{\mathrm{S_o},\min} = 5\,\mathrm{m/s}$. Subsequent states are created in analogy to keep the FMIs temporally disjunct. In Figure 4.2, this is visualised by the solid green line matching the dotted orange line.

However, when following the resulting green trajectory (solid line), the aircraft still arrives too early at the goal. Thus, the motions along the previous edges must be adjusted so that the goal is reached just after the end of the goal constraint, as illustrated by the purple trajectory along path (3). We use this case to introduce two schemes in SIMP to anticipate restrictions as well as to repair infeasible motions:

Definition 10 Anticipation and Backtracking Schemes in SIMP

Let S denote the currently explored state.
The *anticipation scheme* in SIMP sets auxiliary constraints for the motion over the current edge $\epsilon = (\nu, \nu') \in \mathcal{E}$ to ensure that time and speed restrictions at upcoming edges can still be fulfilled. These constraints are state-dependent, i.e. only valid for state S.
The purpose of the *backtracking scheme* is to repair an infeasible motion over the current edge ϵ, necessitating a lower speed or later arrival time at S. To this end, it recursively backtracks to a preceding state S_{pre} from which a feasible motion over the sub-path $\mathcal{P}_{\mathrm{sub}} = \langle \epsilon_{\mathrm{pre}}\,,\, \cdots\,,\, \epsilon \rangle$ exists, allowing to create a feasible successor S^* of S.

To illustrate the backtracking scheme, let us assume that a state at the beginning of the final edge towards the goal is explored, leading to an infeasible motion since the required time $t_S = 100\,\text{s}$ and speed $v_S = 0\,\text{m/s}$ cannot be satisfied. The backtracking scheme then iteratively backtracks to the parent state S_P at approximately 500 m remaining distance to the goal to create the purple trajectory (solid line). However, the anticipation scheme allows SIMP to create the purple trajectory directly from S_P, which avoids the recursive calculation of motion feasibility that is necessary in the backtracking scheme. To this end, it checks the upcoming edges for restrictions, finding the goal constraints at the end of paths (2) and (3). By checking whether the motion along the paths can satisfy all restrictions, SIMP then creates a successor at the goal, connected by intermediate states to S_P. Using the anticipation scheme reduces the runtime of SIMP by limiting the creation of states that would lead to an infeasible trajectory. Although SIMP primarily uses this scheme, it can only anticipate on those constraints that are valid for a node or an entire edge. We thus need the backtracking scheme to resolve infeasible motions that result from graph reservations defined only for a segment of an edge (see Section 4.3.4 for further details).

We further minimise the amount of generated states in SIMP by defining a heuristic function that uses domain-specific circumstances to estimate the remaining cost to reach the goal location:

Definition 11 Heuristic Function in SIMP

Similar to A* and SIPP, the SIMP algorithm uses an admissible heuristic function $h(S)$ to estimate the remaining cost from a given search state S to one of the goal locations $G \in \mathcal{V}$ associated with the final activity α_f in the agent's activity sequence $\mathcal{S}$. $h(S)$ accounts for the current edge ϵ as travel direction, current speed v_S, speed restriction at the goal $v_{\text{lim,goal}}$, the set of remaining activities $\mathcal{S}_{\text{rem}} \subseteq \mathcal{S}$, and the vehicle category `cat` associated with the agent:

$$h(S) = h\left(\epsilon, v_S, v_{\text{lim,goal}}, \mathcal{S}_{\text{rem}}, \texttt{cat}\right) \tag{4.12}$$

The SIMP algorithm terminates if either a new plan is found, or no unexplored states are left, and the high-level search continues as described above. An in-depth description of the AS-MAMP algorithm and the respective pseudocodes of PBS and SIMP are provided in the following section.

4.3 Detailed Description of the AS-MAMP Algorithm

In the following subsections, the technical details of the AS-MAMP algorithm and the definitions from the previous sections are outlined, discussed, and illustrated.

4.3.1 Adaptations to PBS as High-level Search of AS-MAMP

The pseudocode of the adapted version of PBS as high-level solver of AS-MAMP is provided in Algorithm 1 for reference. Changes to the original algorithm from [61] are marked by red line-numbers.

Algorithm 1: High-Level Search of AS-MAMP as adaptation of PBS from [61] (changes marked by red line-numbers)

```
   Input: MAMP instance, searchMode, rootConstraints, rootPriorities (= ∅ by default)
   ## root-node: SAMP-plan for all agents, adhere to root-constraints
1: root ← SearchNode(type="root")   # create root-node as class-instance
2: root.plan ← ∅
3: root.rootConstraints ← rootConstraints
4: root.priorities ← rootPriorities
5: foreach agent a_i do
6:     success ← root.updatePlan(a_i) # invokes SIMP, updates plan in-place
7:     if not success then
8:         return "no solution"
9: root.cost ← c_N                                          # see Equation (4.13)
   ## child-nodes: detect and resolve conflicts between agents
10: STACK ← root
11: while stack ≠ ∅ do
12:     N ← pick and remove node from STACK according to searchMode
13:     conflict ← N.detectConflict()   # get conflict pair, see Algorithm 2
14:     if no conflict then
15:         break                        # N contains conflict-free solution
16:     N.children ← ∅          # conflict exists: resolve it for both agents
17:     foreach a_i in conflict do
18:         N' ← SearchNode(type="child")
19:         N'.plan ← N.plan
20:         N'.rootConstraints ← N.rootConstraints
21:         N'.priorities ← N.priorities ∪ {a_j ≺ a_i}      # agent a_j has priority over a_i
22:         success ← N'.updatePlan(a_i)      # updates a_i and all a_k for which
              a_i ≺ a_k
23:         if success then
24:             N'.cost ← c_N                                   # see Equation (4.13)
25:             N.children ← N.children ∪ {N'}
26:     STACK ← STACK ∪ {N.children in order of decreasing cost}
27: return N.plan of solution-node N OR "no solution"
   Output: conflict-free solution to MAMP-instance if existent and found
```

PBS optimises the sum-of-cost of the agent trajectories. Thus, the cost of a PBS search-node N is defined as:

$$c_N = \sum_{i=0}^{A} c_{\text{prio},i} \cdot c(\Pi_i) \tag{4.13}$$

with the number of agents A, the routing plan cost $c(\Pi_i)$ from Definition 4, and $c_{\text{prio},i}$ as dimensionless cost to weigh the importance of the route of agent i for the overall optimisation goal. For instance, aircraft can be assigned a higher value for c_{prio} than tugs to increase their prioritisation: an increase in plan cost of the aircraft is penalised more than that of the tug, potentially influencing the resulting priority order.

In the root-node of PBS (Lines 1 to 9), the trajectories of all agents are calculated taking merely the constraints set by the Routing Agent (see its P3 and P4) into account. This represents the single-agent motion planning (SAMP) solution of the AS-MAMP search for each Moving Agent. Note that a partial priority ordering can be set in the root-node of the adapted version of PBS as well.

The next search-node is picked based on the chosen *search-mode*, for which we distinguish between either a depth-first, best-first, or greedy search. The **depth-first search** uses the top node of the STACK as new parent-node. Newly created child-nodes are thus placed on top of the STACK in order of decreasing cost. The **greedy search** follows the approach of Chan, Stern, Felner *et al.* [62] to greedily place the child-node with the lowest number of remaining conflict-pairs on top of the STACK irrespective of its sum-of-cost. However, in contrast to the original Greedy PBS (GPBS) implementation, we do not alter the low-level solver. With these two search-modes, PBS only backtracks when it cannot find a solution in the explored branch. While these quickly lead to a solution, the resulting priority orderings may be sub-optimal and thus lead to higher taxi times of the agents. In contrast, the **best-first search** returns the optimal solution within the priority-tree as it continues to pick the node with the lowest cost in the priority-tree until the solution is conflict-free. However, this search mode is limited by both computational time and memory, making it feasible only for small agent populations. In Chapter 5, we use this search-mode to assess the suboptimality of the depth-first and greedy searches for the analysed case studies. As discussed in Section 5.2.1, implementing other search-modes may lead to faster as well as higher quality solutions, and should thus be explored in future work.

The authors of GPBS also introduced the concept of using induced constraints (IC) to further speed up the high-level search. While their use does not necessarily sacrifice solution quality, it does affect the chosen conflict to be resolved (see Section 4.3.5) and thus the resulting priority order. We adopt this technique as variation of all three search-modes when benchmarking AS-MAMP as presented in Chapter 5.

4.3.2 Motion Principles

Definition 8 is based on the following assumptions for modelling vehicle motions:

- the motions are approximated with piecewise-constant acceleration, i.e. $a(t) = \text{const.}$, forming a *piecewise motion segment* over the time interval t;
- since the radius of curvature r of all taxiway segments is large in comparison to the travelled distance and the change in velocity within that distance, we neglect the normal component of the acceleration;

- despite that, speed limits have to be met to traverse curved taxiway segments that have a radius of curvature below a pre-defined threshold value.

Differentiating $a(t) = \text{const.}$ twice and assuming relative distances per motion segment yields Equations (4.6) to (4.8).

We apply these equations to calculate on-the-fly motions along the graph edges during single-agent path planning. Since the equations are only valid for constant acceleration, we split motions into piecewise segments of constant acceleration if needed, and allow for kinks both at nodes and anywhere on edges. In general, the speed profile is optimised for smallest traversal time and highest final speed. The speed is bounded by an agent-dependent maximum speed v_{max} as well as a minimum speed v_{min} when in motion. During planning, reaching a full stop is only allowed at the end-nodes of edges. The agents can wait and hold arbitrarily long at these, as long as no reservations from other agents exist. While waiting is bound to nodes, holding may also occur at the beginning (but not at the end) of a motion along an edge.

By convention, a motion always connects the state at the beginning of an edge with the state at the beginning of a subsequent edge. A trajectory over multiple edges is thus split into one motion per edge before generating the resulting states on the respective nodes (see Section 4.3.7). Note that the duration of motions can have arbitrary values, allowing for execution in continuous time.

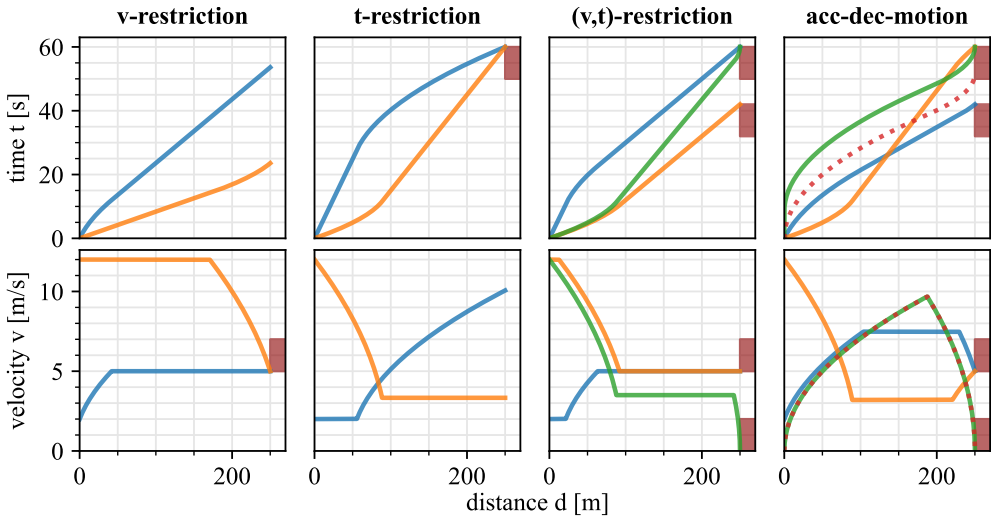


Figure 4.3: Exemplary motions visualised per column both in (d, t)-graph (top) and (d, v)-graph (bottom); the speed and time restrictions for the different examples are marked by red shades

Figure 4.3 illustrates different motions over one or multiple edges with a path length of $d = 250\,\text{m}$. The exemplary motions are plotted in both a (d, t)-graph (top) and a (d, v)-graph (bottom). Speed limits and minimal traversal times as restrictions are depicted by red shades. In the first column, a speed limit of $v_{\text{f,lim}} = 5\,\text{m/s}$ must be reached at the end of the path. Thus, to minimise the traversal time, acceleration from $v_{\text{i}} < v_{\text{f,lim}}$ to $v_{\text{f}} = v_{\text{f,lim}}$ takes place immediately (blue line), while deceleration from $v_{\text{i}} > v_{\text{f,lim}}$ to $v_{\text{f}} = v_{\text{f,lim}}$ occurs towards the end of the path (orange line).

In contrast, when the minimal traversal time must be exceeded due to time restrictions (second column), the motion is optimised for maximum final speed: acceleration occurs towards the end of the segment (blue line), while deceleration happens immediately (orange line). Although both motions take the same time, the blue trajectory yields a substantially shorter traversal time on subsequent edges because of its higher final speed. This illustrates that the fastest overall motion depends not only on past but also on upcoming restrictions. Reaching each node as quickly as possible is thus not always ideal.

The two remaining columns show combinations of speed and time restrictions. In the regular case (third column), acceleration (blue line) and deceleration (orange line) take place in between the two segments with constant speeds v_i and $v_\mathrm{f} = v_\mathrm{f,lim}$, ensuring both constraints are satisfied. In contrast, when $v_\mathrm{f,lim} = 0\,\mathrm{m/s}$ (green line), a constant speed v_const with $v_\mathrm{f} > v_\mathrm{const} \geq v_\mathrm{min}$ is inserted between two deceleration segments to meet the minimal traversal time. Any set of constraints requiring $v_\mathrm{const} < v_\mathrm{min}$ leads to an infeasible motion by definition.

The regular motions with monotonic profiles described so far do not cover special cases such as $v_\mathrm{i} = v_\mathrm{f,lim} = 0\,\mathrm{m/s}$. Furthermore, these profiles are too restrictive when motions span multiple edges, as required for backtracking and anticipation (see Definition 10). We therefore define **acc-dec-motions**, which include an acceleration and a deceleration phase to and from an intermediate speed v_mid. This speed results from the restrictions at the beginning and end of the travelled distance, and is bounded by v_max and v_min. To cover the full distance, an additional segment with constant speed $v_\mathrm{const} = v_\mathrm{mid}$ may be inserted in between. By convention, except for holding at the start of an acc-dec motion, a constant speed is neither allowed at its start nor its end.

The last column in Figure 4.3 shows examples of such motions. In comparison to the third column, the acc-dec-motion allows the blue and orange motions to reach the end of the respective other time restriction depicted by the red shades. The green and dotted red lines depict the special case mentioned above: while the dotted red motion ends in the red shaded area and is therefore infeasible, the green motion includes holding at its start and thus exceeds the minimal traversal time. For these cases, a maximum holding time $t_\mathrm{hold,max}$ can be defined as additional constraint if needed. Potentially, this results in a constant intermediate segment with a smaller v_mid.

4.3.3 Definition of Motion Profiles, Shapes, and Safety Distances

In general, agents have a circular shape with a radius according to their type as listed in Table 3.1. Each shape has a centre, front, and back, as illustrated in Figure 4.4. The centre represents an agent's reference point as defined in [111]. A motion thus specifies the movement of the centre. We refer to the centre's collective motion over multiple edges as a **motion profile**, i.e. the agent's four-dimensional trajectory. The motion profile specifies the time points and speeds at the absolute distances at which the acceleration changes. Therefore, for each segment in the motion profile with constant acceleration, we can use Equations (4.6) to (4.8) to determine any time point t, travelled distance d as absolute or relative value, and speed v on the continuous scale. As the front and the back of a shape move in unity with the centre, their respective motion profile is obtained by shifting the absolute distances of the central motion profile by the radius of the shape. Then, the corresponding motion over an edge in the path of the agent can be determined in analogy to obtaining a single point in the profile as described above.

In real-world operations, taxiing vehicles have to keep a minimum safety distance. For aircraft, this is mainly due to the engine blast that poses a safety risk to any object or person in their wake. We know from expert interviews that other vehicles should keep a safety distance of around twice the length of the aircraft in front of them to mitigate the risk of engine blast. The experts highlight that the actual safety distance maintained by pilots and ground vehicle drivers often falls short of the recommended length. Since pilots and auto-pilots can only follow planned trajectories with limited precision (see Requirement 6), we include an allowable deviation as positioning tolerance within the safety zone.

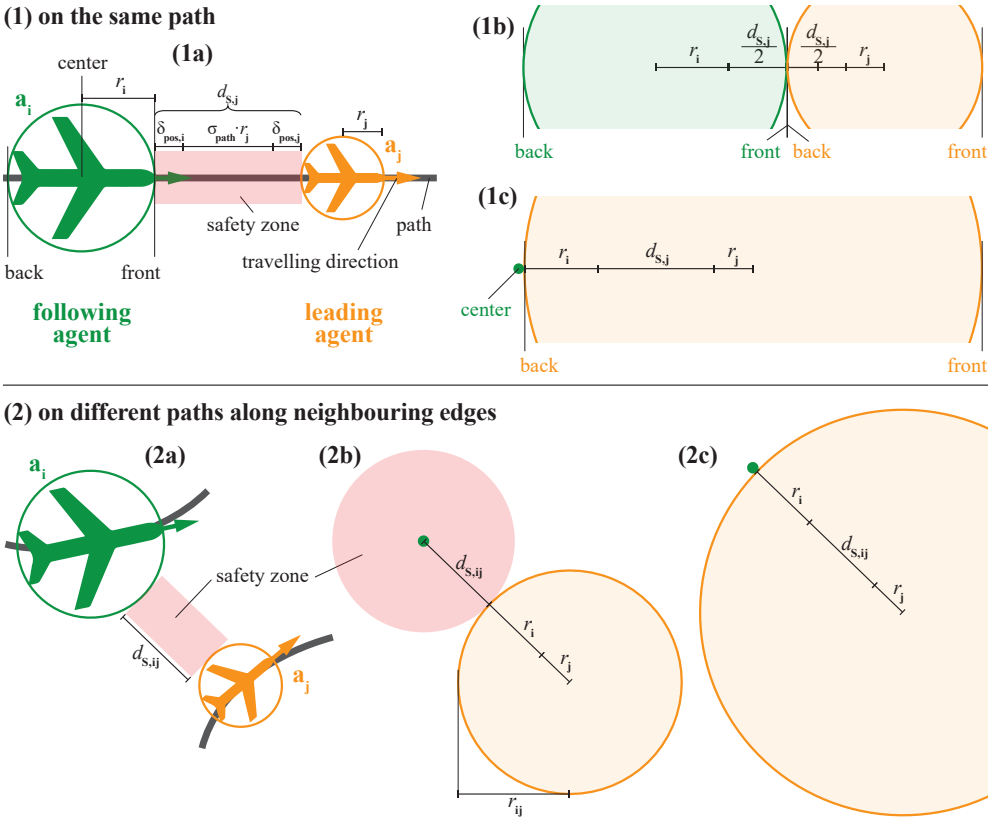


Figure 4.4: Definition of shapes and safety distances when two agents a_i and a_j travel (1) on the same path and (2) on different paths

In AS-MAMP, we hence define that all agents have to keep a minimum safety distance d_S between each other in dependence of their shape radii as well as the positioning tolerance δ_{pos} for each agent. Since the taxiway network is represented as a graph, we distinguish between the minimum safety distance between agents on the same path (see Figure 4.4.1), and that between agents on different paths (see Figure 4.4.2). When travelling *on the same path*, we define the minimum safety distance between two agents in relation to the shape radius of the leading agent. As illustrated in Figure 4.4.1a, for an agent a_i following an agent a_j along the same path, their minimum safety distance $d_{S,j}$ is defined as:

$$d_{S,j} = \sigma_{path} \cdot r_j + \delta_{pos,i} + \delta_{pos,j} \tag{4.14}$$

with the safety factor σ_{path} multiplying the shape radius r_j of agent a_j, and δ_{pos} for each of the two agents. We make use of the safety factor to provide an operational parameter that can be adjusted when studying airport surface movement operations with the multi-agent system. Nonetheless, we recommend to use values for σ_{path} between 2–6 to match the practical considerations of the experts. We will evaluate the effect of different positioning tolerances with a sensitivity analysis of δ_{pos} in Chapter 5.

When travelling *on different paths*, we define the minimum safety distance between two agents a_i and a_j in relation to their average shape radius $\frac{r_i+r_j}{2}$ as:

$$d_{\text{S},ij} = \sigma_{\text{NE}} \cdot \frac{r_i + r_j}{2} + \delta_{\text{pos},i} + \delta_{\text{pos},j} \tag{4.15}$$

with the safety factor σ_{NE}. Since the risk of engine blast is lower when agents do not directly trail each other, we recommend to use values for σ_{NE} between 1 to 3.

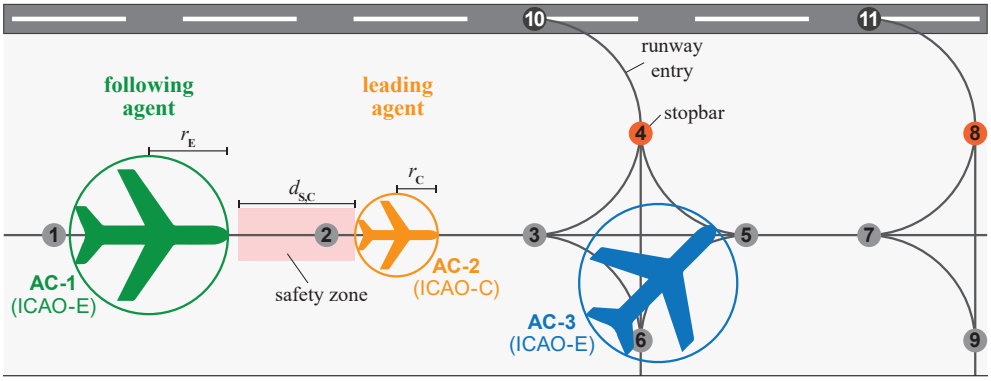
When determining the remaining gap between two moving shapes, different representations of their shapes are possible, which is illustrated in the sub-figures (b) and (c) in Figure 4.4. For two agents on the same path, half of the minimum safety distance can be attributed to each agent, respectively (1b). This representation is useful in conflict detection, as outlined in Section 4.3.5. Since SIMP plans a new trajectory for the centre of a shape, (1c) depicts a representation in which one of the agents is represented as point while both shape radii as well as the minimum safety distance are associated with the other agent. The same approach is feasible for shapes on neighbouring edges (see 2c). Additionally, when setting graph reservations as outlined in Section 4.3.4, the representation in (2b) is needed: it attributes both shape radii as combined shape to a single agent while the other agent keeps the minimum safety distance as safety zone with radius $d_{\text{S},ij}$. To this end, we define the radius of the combined shape as $r_{ij} = r_i + r_j$.

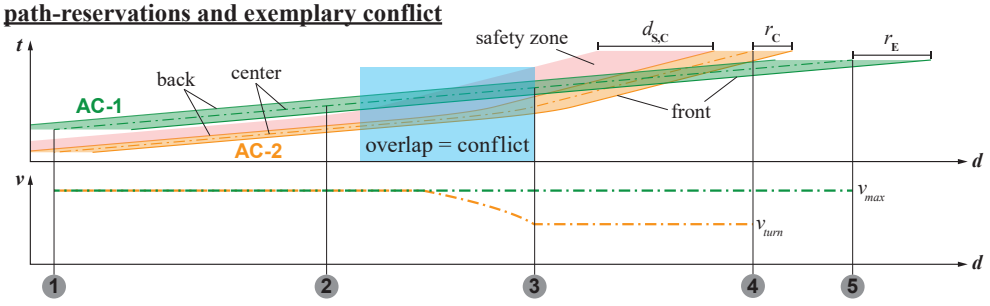
4.3.4 Graph Reservations

To account for dynamic obstacles during single-agent path planning (i.e. the low-level search), the motions of other agents must be known to the planner. While the original SIPP implementation defines collision intervals to do so, these do not account for the shape and kinematics of other agents. As noted by Walker and Sturtevant [134], an algebraic equation to detect collisions between accelerated motions must still be derived. The alternative approaches reviewed in Section 2.3.3 draw on assumptions that are invalid for graphs with arbitrarily long, curved, and oriented edges. Moreover, we must also account for layout constraints such as temporarily blocked edges due to changing RMOs during planning. These considerations motivate a new approach to handle dynamic obstacles and time constraints as graph reservations, as defined in Definition 5 and described in the following.

Once the SIMP algorithm has terminated for an agent, the resulting plan is converted into graph reservations. More specifically, we distinguish between three types of graph reservations:

- **path-reservations** on edges along the path of an agent;
- **neighbourhood-reservations** on all edges not on the path of an agent but swept by its shape, which includes edges on the path in reverse direction;
- **WTC-reservations** on runway entries to ensure sufficient temporal separation according to the wake turbulence category (WTC) of an agent with respect to other agents.





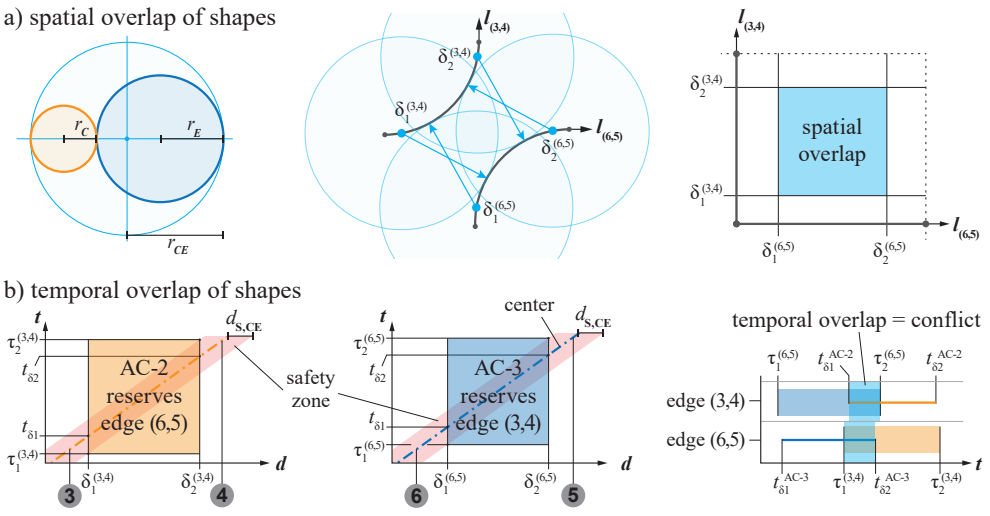


Figure 4.5: Graph reservations and exemplary conflicts (illustrated in top part), with path-reservations on edges along an agent's path (middle part) and neighbourhood-reservations on all other edges that are swept by an agent's shape (bottom part)

Path-reservations are necessary to let an agent follow – but not overtake – a higher prioritised agent that is travelling in front of it, while keeping the minimum safety distance as defined in Section 4.3.3. Such reservations are illustrated in the top part of Figure 4.5: the orange agent *AC-2* (ICAO-C) traverses the edges (1,2), (2,3) and (3,4) towards the runway entry (4,10). It is followed by the green agent *AC-1* that has to use the runway entry (8,11) due to its larger size (ICAO-E). They thus share the edges (1,2) and (2,3). To reserve the traversed path, the motion of an agent's centre as well as the radius of its shape are stored separately per edge as path reservation.

In contrast, **neighbourhood-reservations** are set from an agent traversing an edge onto neighbouring edges. In the illustrated example in Figure 4.5, the orange agent *AC-2* cannot traverse edge (3,4) as long as the blue agent *AC-3* still occupies edge (6,5), since their shapes would collide. A neighbourhood-reservation comprises two parts to represent the space-time collision interval: the reservation is set for the *edge-interval* as reserved segment of the other edge which must not be traversed by other agents during the *time-interval* as its second part.

An edge-interval marks the start and end positions normalised by the length of the edge for which the reservation is set. These positions result from the geometric overlap based on the combined shape of two agents as defined in Figure 4.4.2b. For the illustrated example, the bottom row of Figure 4.5 depicts the radius r_{CE} as combined shape of AC-2 with type ICAO-C and AC-3 with type ICAO-E. With the combined shape, we calculate the geometric overlap between a pair of edges: when the combined shape traverses edge (3,4) it overlaps with edge (6,5) between $\delta_1^{(3,4)}$ and $\delta_2^{(3,4)}$. The overlap-interval of edge (6,5) with edge (3,4) is likewise calculated as $\delta_1^{(6,5)}$ and $\delta_2^{(6,5)}$. These normalised positions on the edge-pair form the edge-intervals of each edge when setting a neighbourhood-reservation from the other edge. Since the spatial overlap between an edge-pair depends on the combined shape of two agents, each combination of two agent types yields one edge-interval-pair per edge-pair. The intervals are pre-calculated for all edge-pairs and shape-combinations.

The time-interval as second part of a neighbourhood-reservation is determined with the motion profile over an edge: as outlined in Section 4.3.3, we determine the time points at which the safety zone around the point-shape of an agent (green agent a_i in shape-representation 2b in Figure 4.4) enters and leaves the normalised positions δ_1 and δ_2 on the edge from which the reservation is set. In the example, agent *AC-2* reserves the edge (6,5) between $\tau_1^{(3,4)}$ and $\tau_2^{(3,4)}$, given the safety distance $d_{\mathrm{S},ij}$ around it. This results in the neighbourhood-reservation on edge (6,5) between the normalised positions $\left(\delta_1^{(6,5)}, \delta_2^{(6,5)}\right)$ as edge-interval during $\left(\tau_1^{(3,4)}, \tau_2^{(3,4)}\right)$ as time-interval. Likewise, agent *AC-3* reserves the edge-interval $\left(\delta_1^{(3,4)}, \delta_2^{(3,4)}\right)$ of edge (3,4) for the time-interval $\left(\tau_1^{(6,5)}, \tau_2^{(6,5)}\right)$.

We account for the required takeoff-separation according to the wake turbulence category (WTC) of a departing aircraft by defining **WTC-reservations**. These are set on the directional edges of all entries to the respective runway as time intervals per type-combination of the agent with any other aircraft type. A time interval consists of the start time point t_{s} and end time point t_{e}:

$$t_{\mathrm{s}} = t_i^{\mathrm{StB}} - t_{j\to i}^{\mathrm{WTC}} \qquad \text{(i departs after j)} \tag{4.16}$$

$$t_{\mathrm{e}} = t_i^{\mathrm{StB}} + t_{i\to j}^{\mathrm{WTC}} \qquad \text{(i departs prior to j)} \tag{4.17}$$

with t_i^{StB} denoting the time at which the centre of agent i is at the stopbar before entering the runway, and t^{WTC} being the durations required for the WTC-separations between agents a_i and a_j. In the example in Figure 4.5, both agents *AC-3* and *AC-1* use edge (8,11) as runway entry. Presuming that AC-3 will be in front, it reserves the runway entries (8,11) and (4,10) between t_s and t_e with $t^{\text{StB}} = t_8$ as the time of passing the stopbar at node (8) and $t_{j\to i}^{\text{WTC}} = t_{i\to j}^{\text{WTC}} = 60\,\text{s}$ from Table 3.2.

Before updating the plan of a de-prioritised agent, all reservations from agents higher in priority are gathered. The illustrated collisions in Figure 4.5 are thus resolved when a priority order between the three agents is set. The next subsection outlines how the necessary conflict detection is carried out.

4.3.5 Conflict Detection

Following Definition 6, we execute a conflict detection scheme to find all conflicts between each pair of agents. The corresponding pseudocode is outlined in Algorithm 2 for reference. We use an approach based on an exhaustive search: the reservations of each agent a_i are compared against those of every other agent a_j. Conflict detection is skipped if a priority order already exists between the two agents, or their plans are disconnected in time. Otherwise, we compare the path-, neighbourhood-, and WTC-reservations of agent a_i with those of agent a_j.

Algorithm 2: Conflict detection: without or with induced constraints (IC)

```
Input: graph reservations of all agents
conflicts ← ∅
foreach agent a_i do
    foreach agent a_j do
        ## check if conflict detection is needed between a_i and a_j
        skipDetection ← checkConflictDetectionNeeded(a_i, a_j)
        if skipDetection then
            continue                    # conflict detection not required
        ## detect path-conflicts between a_i and a_j
        foreach edge in plan of a_i do
            if edge in plan of a_j then
                conflict ← checkMotionOverlap(motions of a_i, a_j along edge)
                conflicts ← conflicts ∪ {conflict}
        ## detect neighbourhood- and WTC-conflicts between a_i and a_j
        foreach edge in plan of a_i do
            foreach reservation of a_j on edge do
                if motion of a_i overlaps with reservation then
                    conflict ← store conflict-information
                    conflicts ← conflicts ∪ {conflict}
if without IC then
    conflict ← conflict in conflicts with min(t_start)
else
    conflict ← conflict with max(IC); breaking ties with min(IC_lower), then min(t_start)
return conflict
Output: agent pair in conflict
```

The agents are in a **path-conflict** if their path-reservations overlap on an edge that both of them traverse. To detect such a conflict, we first determine the sequence of the agents passing the edge. In the example in Figure 4.5, agent *AC-2* is in front of agent *AC-1* on the edges (1,2) and (2,3). Using the shape-representation 1b in Figure 4.4, we then compute the minimal gap between the front-line of the shape of the trailing agent (AC-1) with the back-line of the shape of the leading agent (AC-2). Note that half of the minimum safety distance is included in each shape as depicted in Figure 4.4.1b. The minimal gap between the two shifted motion-profiles (see Section 4.3.3) is analytically calculated by determining the minimum distance between the underlying motion-functions. If the distance is negative, the motions overlap, and thus form a path-conflict. As marked in the example, the motions of AC-1 and AC-2 overlap on edge (2,3), since AC-2 slows down for the curved edge (3,4) while AC-1 can keep its maximum speed to go straight ahead along edge (3,5).

In contrast, agents are in a **neighbourhood-conflict** if the neighbourhood-reservations of one agent overlap with the motion of the other agent's centre on the same edge. Recall that a neighbourhood-reservation consists of a time-interval during which another agent must not traverse the corresponding edge-interval of the edge for which the reservation is set. To detect such conflicts, we iterate over each edge in the plan of agent a_i. For each neighbourhood-reservation of agent a_j, we determine the time-interval at which the centre of a_i enters and leaves the normalised edge-positions according to the edge-interval of the reservation. The agents are in a neighbourhood-conflict if the resulting time-interval of a_i overlaps with the time-interval of the neighbourhood-reservation set by a_j. In Figure 4.5, such a conflict occurs between agent *AC-2* and agent *AC-3* between the edges (3,4) and (6,5): the motion of the safety zone around the centre of AC-2 traversing edge (3,4) overlaps with the neighbourhood-reservation of AC-3 that are set for that edge. Likewise, the reservation of AC-2 on edge (6,5) overlaps with the motion of AC-3 over that edge.

In analogy, agents are in a **WTC-conflict** if the WTC-reservation of one agent overlap with the motion of another agent on any edge that is an entry to the same runway. Using Equation (4.17), the time interval of agent a_i occupying the runway is determined. The agents are in a WTC-conflict if it overlaps with the time interval that agent a_j reserved on all runway entries.

The conflict detection scheme is applied to every combination of two agents and yields all existing conflicts. In general, the first one in time is selected. However, when using a PBS search-mode with induced constraints (IC), we follow the approach of GPBS with IC [62]: the agent pair with the largest IC-set is selected, breaking ties by choosing the pair with the smallest set of agents lower in priority. This is complemented by breaking ties in favour of the earliest conflict in time. To resolve the selected conflict, recall that two new PBS search-nodes are created with each conflicting agent being added to the others priority list. Per search-node, a new list of conditions for the de-prioritised agent is formed and SIMP is invoked.

4.3.6 SIMP: Objective Function and Heuristic Search

Similar to A* and SIPP (see Section 2.3.3), the Safe Interval Motion Planning (SIMP) algorithm searches for a path from a start to a goal location by minimising the associated cost of the resulting trajectory. The f-score $f(S)$ of a state S estimates the path cost:

$$f(S) = g(S) + h(S) \tag{4.18}$$

with the g-score $g(S)$ as accrued cost between the start location and S, and the h-score $h(S)$ as estimation of the remaining cost to reach the goal. Recall from Definition 11 that the heuristic function $h(S)$ also accounts for the vehicle category cat. This determines which edges of the graph can be traversed in general, as some edges are not accessible to all agents: for example, service roads cannot be used by aircraft, and some taxiways are wingspan-restricted.

To force SIMP to follow the agent's activity sequence, we calculate $h(S)$ as sum of the heuristic per activity $h(\alpha_k)$:

$$h(S) = \sum h(\alpha_k) \tag{4.19}$$

For GoTo activities, multiple end nodes may be defined (see Definition 1). In that case, the minimum of each $h(S)$ per combination j of end nodes within the sequence is chosen so that $h(S)$ remains consistent:

$$h(S) = \min(h_j(S)) \tag{4.20}$$

Each $h(\alpha_k)$ is calculated by using Equation (4.1) with $d_{\text{taxi}} = d_{\text{h}}(\alpha_k)$ and $t_{\text{taxi}} = t_{\text{h}}(\alpha_k)$ as well as the c_{d} defined for the respective agent. For $d_{\text{h}}(\alpha_k)$, the node-to-node shortest distance along the graph edges between the start and end node of the respective activity is used in general. However, since aircraft cannot turn on the spot at a state S in their current activity α_{S}, we use an edge-to-node shortest distance for $d_{\text{h}}(\alpha_{\text{S}})$ based on the directional edge in the travelling direction of the agent. These distances are pre-calculated for all edge-node-combinations in the layout. Per $d_{\text{h}}(\alpha_k)$, and depending on the speeds at the start and end nodes of α_k, we apply Equations (4.6) to (4.8) to calculate the corresponding minimal traversal time $t_{\text{h}}(\alpha_k)$. Note that for waiting-activities, $d_{\text{h}} = 0$ and t_{h} equals the waiting duration.

Throughout the search, to ensure that the heuristic is admissible and consistent, we verify that:

$$h(S) \leq c_{\text{m}}(S \rightarrow S') + h(S') \tag{4.21}$$

with $c_{\text{m}}(S \rightarrow S')$ representing the motion cost between state S and its successor S' and $h(S')$ denoting the h-score from S' to the goal. c_{m} is calculated by using Equation (4.13) with $t_{\text{taxi}} = t_{\text{m}}$ as the duration of motion m and $d_{\text{taxi}} = d_{\text{m}}$ as the traversed distance.

4.3.7 SIMP: State Space

In the following, we provide additional details on the state definition in SIMP from Definition 9, and the core part of the SIMP algorithm is provided in Algorithm 3. The algorithm keeps track of its state space with an OPEN-list and a CLOSED-list, akin to A* and SIPP. However, the lists serve a different purpose in SIMP, and states also store additional information, such as a reference to their parent-state, the g-, h-, and f-scores, the arrival distance, and speed v_{mid} to indicate an acc-dec-motion (see Section 4.3.2).

The purpose of the OPEN-list is to mark unexplored states. To follow the logic behind SIPP, OPEN shall contain states that cluster the motions along the following edge towards the subsequent node into intervals to reduce the number of similar states in the search space, motivating the definition of states through a combination of agent configuration S_{cfg} and the state's feasible motion interval S_{FMI}, as defined in Definition 9.

To create a trajectory with minimal cost (see Section 4.3.6), the state S in OPEN with the lowest f-score is picked and removed. Successors of S are generated by executing the

Algorithm 3: Safe Interval Motion Planning (SIMP) Algorithm

```
   Input: route parameters of agent, reservations of higher prioritised agents
1:  CLOSED ← ∅
2:  OPEN ← getInitStates()                          # creates start states S_init
3:  while open ≠ ∅ do
4:      S ← state in OPEN with lowest f-score
5:      if S fulfils goal-conditions then
6:          plan, cost ← getPlan(S)                 # backtracks plan from S to S_init
7:          return plan, cost                       # solution found: SIMP can terminate
8:      if S not in closed then
9:          add S to CLOSED
10:     else
11:         chooseStateToKeep(S)    # same state-ID in closed: keeps the
             state with lower f-score and updates child-states
12:         continue      # successors of this state were already created
             before
13:     successors ← getSuccessors(S)
14:     foreach S* in successors do
15:         cfg, FMI ← extract from S*
16:         S* ← make FMI of S* temporally disjunct
17:         OPEN ← OPEN ∪ S*                        # indexed by (cfg, FMI)
18: return ∅, ∞                                     # no solution found
   Output: plan from start to goal node and its cost, if existent
```

function **getSuccessors**, which is further outlined in the next subsection. S is then moved to the CLOSED-list (Lines 8 to 12). Once a state in OPEN is picked that meets the goal conditions, SIMP terminates and returns the resulting plan and cost (Lines 5 to 7).

In contrast to clustering states, the purpose of the CLOSED-list is to quickly obtain a parent-state during the search as well as generate the final plan once the search has terminated without recalculating the motions between two states. We thus define the unique state-ID S_{ID} for states in CLOSED as:

$$S_{\text{ID}} = (\nu\,,\,\nu'\,,\,t_{\text{S}}\,,\,v_{\text{S}}\,,\,\alpha_k) \tag{4.22}$$

with the arrival time t_{S} and arrival speed v_{S} at node ν as well as the activity α_k that is valid from the state onwards. If a state with identical state-ID is added to CLOSED, SIMP keeps the one with the lower f-score, and updates its state-values as well as those of all its successors accordingly. Correspondingly, all child-states are recursively updated.

4.3.8 SIMP: State Generation

In this subsection, we outline the generation of successors based on the pseudocode of the function **getSuccessors** and an illustrating example provided in Figure 4.6. Due to our state definition, multiple successor states S' may be defined for traversing the edge on which state S is defined. To limit the amount of repetitive computations for the motion along an edge, we first gather the speed restrictions and safe time intervals that must be satisfied to enter the next edges. These form multiple sets of boundary conditions for the motion along an edge. We cluster them along with the next edges that require an identical set of these motion boundaries (Line 5 in Algorithm 4). Furthermore, we sort them based on earliest arrival time within the safe interval, breaking ties in favour of highest speed limit v_{lim}.

Then, following the sorted order, a motion is computed that satisfies the set of conditions, and the respective successors are created (Lines 8 to 22).

Algorithm 4: SIMP function `getSuccessors(S)`

```
   Input: state S
1: successors ← ∅
   ## special case: wait-activity, switching to next activity
2: if S.activity = "wait-activity" then
3:     S' ← getSuccessorWaitActivity(S)
4:     return S'
   ## get motion restrictions for next edges
5: motionBounds ← getMotionBounds(S)            # based on v-restrictions
6: if not motionBounds then
7:     return ∅                                          # no way forward
   ## calc motions for next edges. Start with quickest possible
      motion
8: motionBounds.sorted()                          # sort for quickest motion
9: motionPrev ← ∅
10: foreach motionBound in motionBounds do
11:     if motionPrev slower than motionBound allows then
12:         continue       # previous motion slower than this bound, skip
             recalculation
13:     motion ← calcMotion(S, motionBound)    # calculate motion based on
         bounds
14:     foreach edgeID in nextEdgeIDs do
15:         S' ← createSuccessor(motion)       # create successor based on
             motion
16:         successors ← successors ∪ {S'}
        ## use backtracking iff vFinal < vLimit
17:     if vFinal < vLimit then
18:         S" ← createSuccsBacktrack()
19:         successors ← replace S' with S"
        ## anticipate on upcoming motion restrictions
20:     S' ← createSuccsAnticip()
21:     successors ← successors ∪ {S'}
22:     motionPrev ← motion
23: return successors
   Output: successor-states S' of state S
```

In the example in Figure 4.6, three operations are needed to create five regular successors (ID1 to ID5): first, the motion to enter edge (3,5) is created taking 12 s with $v_\mathrm{i} = v_\mathrm{f} = 15\,\mathrm{m/s}$ (ID1). Then, meeting the speed limit of $v_\mathrm{lim} = 5\,\mathrm{m/s}$ for edges (3,4) and (3,6), the fastest motion takes 16.4 s, arriving within the first safe interval of each edge (ID2 and ID3). Finally, the successors ID4 and ID5 are created to arrive at the beginning of the second SIs on edges (3,5) and (3,6). To form a motion that spans 40 s, it is necessary to immediately start to decelerate from v_i to $v_\mathrm{f} = 1.4\,\mathrm{m/s}$, and keep a constant speed for the remaining length of the edge (2,3). In contrast, a successor of S that enters edge (3,4) at the beginning of its second safe interval is infeasible: its motion would require a duration of 110 s, but the safe interval on edge (2,3) already ends at $t = 100$ s, thus requiring the motion to last less than 90 s.

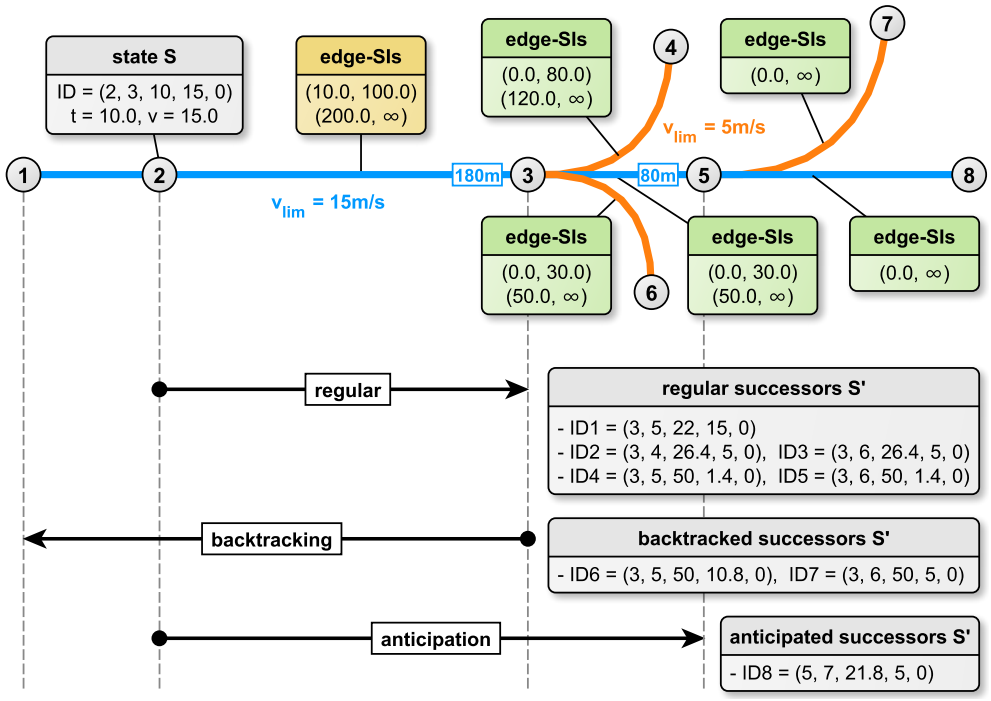


Figure 4.6: Example of successor generation: from state S on edge (2,3), different successors are created on edges (3,4), (3,5), (3,6), and (5,7) due to speed limits at straight segments (blue lines, $v_{\text{lim}} = v_{\text{max}}$), curved segments (orange lines, $v_{\text{lim}} = v_{\text{turn}}$), as well as safe intervals (SIs) on edge (2,3) (shaded in yellow) and for entering the upcoming edges (shaded in green); all resulting successors S' are displayed

The two successors created for the second SIs on edges (3,5) and (3,6) enter these edges with a lower speed than the speed limit of 5 m/s: to reach the required traversal time, the agent has to slow down significantly. In this case, arriving later and/or with a smaller speed at node (2) would yield a higher final speed at node (3). Therefore, for the sake of this example, we use the backtracking scheme (Lines 17 to 19) that was introduced in Definition 10 to optimise the respective final speed for entering the two edges: the function uses an acc-dec-motion explained in Section 4.3.2 from the parent state of S to satisfy the restrictions. In the example, backtracking yields the two successors ID6 and ID7 that replace the successors ID4 and ID5, respectively. Note that the speed limit may not be reachable in all cases due to other constraints on the previous edges.

On the other hand, the final speed of a regular successor may be too high to decelerate sufficiently for the speed restrictions on upcoming edges. We use the anticipation scheme introduced in Definition 10 to check for such cases and to compute a motion from state S directly to the start node of that future edge. In the example, this is illustrated with edge (5,7): the length of edge (3,5) does not suffice for an agent to decelerate from 15 m/s to the turn speed of 5 m/s. Therefore, the function implementing the anticipation scheme (Lines 20 to 21) creates a successor of state S (ID8) directly for entering edge (5,7).

Motion bounds do not have to be calculated when the agent has to wait at the current node due to a `Wait` activity: the successor can directly be created based on the waiting

duration (Lines 1 to 4). If multiple SIs exist in which the agent can enter a subsequent edge, additional successors for holding until each start of these SIs are created as well. Note that holding is only possible until the end of the safe interval at the waiting node.

4.3.9 Replanning

When the AS-MAMP algorithm is invoked by the Routing Agent, some agents may already be taxiing. Thus, these agents are likely in motion and in arbitrary places in the taxiway network at the planning time t_p. Since states are always defined on nodes, the Guidance Agents predict when and with which speed these agents will arrive at the next node on their respective paths (see their P3). The (node, time, speed)-information is then used as initial conditions when updating their plans in SIMP. To set the reservations correctly, the graph reservations explained in Section 4.3.4 include the motion and edges along the already executed part of the route. However, reservations that already ended before t_p are dropped, since these lie in the past and agents cannot infringe them any more. Thus, only those previous reservations that are still relevant during re-planning are accounted for.

This chapter introduced the AS-MAMP algorithm as planning approach in which PBS as high-level coordination mechanism resolves multi-agent conflicts and SIMP as novel low-level planner computes conflict-free 4D trajectories under operational constraints (RQ 3). The next chapter evaluates the algorithmic performance of these components through a structured benchmarking study on a synthetic airport layout: the high-level of AS-MAMP is benchmarked against variants of PBS, while SIMP is compared to SIPP and to A* with kinodynamic constraints. Building on these controlled experiments, the chapter further assesses AS-MAMP on the real-world layout of Amsterdam Airport Schiphol to examine scalability and operational relevance under realistic airport complexity. Finally, the chapter discusses the modelling approach presented in Chapter 3 and the AS-MAMP algorithm in terms of limitations and challenges for implementation in real-world operational environments.

5

Evaluating the AS-MAMP Algorithm within the MAS Model

The previous chapters outlined the MAS model instance as illustration of the generalised architecture (Chapter 3), and specified the AS-MAMP algorithm as core decision-logic of that model (Chapter 4).

In this chapter, the algorithmic performance of the MAS model is evaluated. To this end, the AS-MAMP algorithm is benchmarked by varying its high-level prioritisation scheme, and exchanging SIMP with two existing low-level solvers, namely the Safe Interval Path Planning (SIPP) and A* algorithms. Evaluation scenarios on both a synthetic and a real-world airport layout demonstrate the efficacy of AS-MAMP in planning safe and efficient 4D trajectories. The results further show that the model can handle runway throughput levels matching or exceeding those of large European airports.

This chapter then reflects on the results, limitations, and open research directions of Chapters 3 to 5, and concludes with a summary of the main findings.

This chapter is based on the following publication:

M. von der Burg, J. Kamphof, J. Soomers and A. Sharpanskykh, 'Towards next-generation airport surface movement operations using hierarchical-distributed multi-agent planning', *Transportation Research Part C: Emerging Technologies*, vol. 186, p. 105 583, 2026, ISSN: 0968-090X. DOI: 10.1016/j.trc.2026.105583.

5.1 Model Evaluation

In this section, we evaluate the performance of the multi-agent system model for fully-automated operations with centralised planning, with the main focus on its most important centralised planning component – the AS-MAMP algorithm. To this end, we use multiple scenarios on both a synthetic and a real-world airport layout (Section 5.1.1). First, we benchmark AS-MAMP with PBS variants and FCFS as high-level solvers (Section 5.1.2), combined with SIMP, SIPP, and A* with kinodynamic constraints as low-level solvers in AS-MAMP (Section 5.1.3). Next, we study the influence of key operational parameters (Section 5.1.4), analyse the effect of the planning window (Section 5.1.5), and assess its computational efficiency and scalability (Section 5.1.6). We then evaluate AS-MAMP using the flight schedule of two operational days at Amsterdam Airport Schiphol in Section 5.1.7, simulating both multi-engine taxiing (MET, see Section 2.1) and tug-enabled taxiing (TET). Finally, to stress-test the algorithm, we examine the maximal attainable runway throughput per hour for MET and TET operations on the synthetic layout, and link them to those reported for large European airports (Section 5.1.8).

5

5.1.1 Definition of Test Cases

In taxiway networks, potential bottlenecks may arise in different locations. To assess how AS-MAMP deals with them, we analyse the bottlenecks separately. To this end, we created the synthetic layout visualised in Figure 5.1 which accentuates the bottlenecks, and define two schedules each with 7 flights in total as listed in Table 5.1.

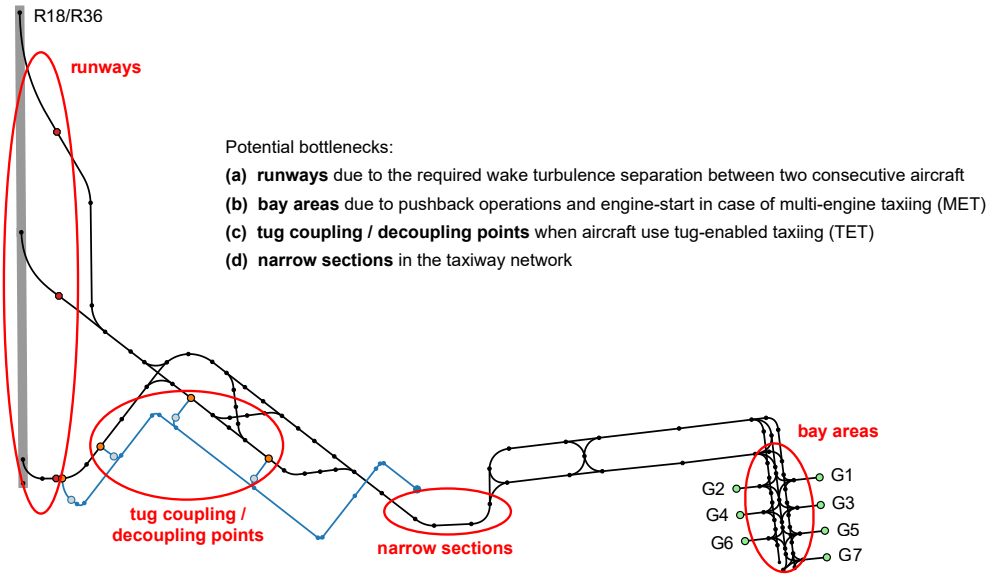


Figure 5.1: Synthetic layout with potential bottlenecks marked by red circles that form the basis of the test cases

In the scenario *Queuing*, the 7 aircraft depart from all available gates in the layout (see Figure 5.1). To isolate the runway as bottleneck, **test case A** considers MET without pushback operations and with engines being already started. To optimise the trajectory

Table 5.1: Flight schedules for the sensitivity analyses on the synthetic layout, with T as fictive reference time

(a) Queuing scenario

ID	Type	At	From	To
1	ICAO-C	$T + 0\,\mathrm{s}$	G1	R36
2	ICAO-E	$T + 20\,\mathrm{s}$	G2	R36
3	ICAO-C	$T + 40\,\mathrm{s}$	G3	R36
4	ICAO-E	$T + 60\,\mathrm{s}$	G4	R36
5	ICAO-C	$T + 80\,\mathrm{s}$	G5	R36
6	ICAO-E	$T + 100\,\mathrm{s}$	G6	R36
7	ICAO-C	$T + 120\,\mathrm{s}$	G7	R36

(b) Crossflow scenario

ID	Type	At	From	To
1	ICAO-E	$T + 0\,\mathrm{s}$	R18	G2
2	ICAO-E	$T + 60\,\mathrm{s}$	R18	G4
3	ICAO-E	$T + 120\,\mathrm{s}$	R18	G6
4	ICAO-C	$T + 0\,\mathrm{s}$	G1	R36
5	ICAO-C	$T + 40\,\mathrm{s}$	G3	R36
6	ICAO-C	$T + 80\,\mathrm{s}$	G5	R36
7	ICAO-C	$T + 120\,\mathrm{s}$	G7	R36

costs, the AS-MAMP algorithm must determine a suitable runway sequence so that the WTC-separation (see Table 3.2) leads to minimal queuing delay before entering the runway (see Figure 5.2). For **test case B**, pushback operations and engine-start are included, congesting the bay area: decoupling from the pushback tractor takes 60 s, and engine-start either 180 s (ICAO-C) or 360 s (ICAO-E). Thus, AS-MAMP must find a suitable pushback sequence. Note that agents can either hold at their gate, or on the bay area after pushback. To investigate the bottleneck of tug-decoupling locations, **test case C** uses the same setup as case A but with tug-enabled taxiing. AS-MAMP has to choose one of the four decoupling locations and must account for the decoupling duration of 120 s. In **test case D**, we use the *Crossflow* scenario with three large arriving and four medium-sized departing aircraft, without considering pushback operations, to isolate the narrow section in the layout as bottleneck. Therefore, the AS-MAMP algorithm must optimise the sequence in which the departing and arriving aircraft pass through it.

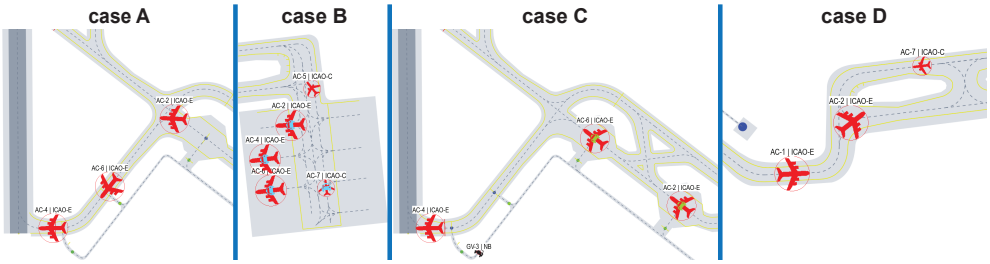


Figure 5.2: Simulation snapshots illustrating the bottleneck of each test case on the synthetic layout

First, AS-MAMP is benchmarked against PBS variants as well as alternative low-level solvers based on these synthetic test cases using a *single* planning round. However, to evaluate the influence of its windowed planning (Definition 3), continuous schedules are needed. The schedule generation is outlined next to the respective assessment in Section 5.1.5. Moreover, in Section 5.1.7, we examine the performance of high-level variants in AS-MAMP on the real-world layout of Amsterdam Airport Schiphol using the flight schedules of two of the busiest days at the airport to date. Our previous studies provide more information as well as the required data comprising the layout, runway configurations, and flight schedules to test both MET and TET scenarios [131], [132]. In

Section 5.1.8, we stress-test the AS-MAMP algorithm by evaluating the potential runway throughput on the synthetic layout with continuous schedules for departing flights of MET (test case A) and TET operations (test case C).

The values for key AS-MAMP parameters such as the kinodynamic properties, cost factors, safety factors, and positioning tolerance are provided in Table 5.2. The maximum speeds for straight and curved segments were determined through expert interviews and current technological limitations of tugs. To evaluate the capabilities of the AS-MAMP algorithm, we chose rather low and differing absolute values for the acceleration and deceleration of the agents, which have a large influence on the KPIs (see Section 5.1.4). As specified in Section 4.3.3, the safety radii along an agent's path and the neighbouring edges are defined as factors in relation to the shape radius (see Table 3.1) of the leading aircraft. Their respective values listed in Table 5.2 resulted from expert interviews.

Table 5.2: Default values for the key parameters of AS-MAMP

parameter	aircraft	tugs	unit
maximum speed v_{max}	15	11.8*	m/s
maximum turn speed $v_{\text{turn,max}}$	5	5	m/s
minimum speed v_{min}	1.5	1.5	m/s
acceleration acc	0.25	0.25	m/s^2
deceleration dec	-0.75	-0.75	m/s^2
cost factor c_{d}	0.1	0.1	s/m
cost factor c_{prio}	1	1	–
safety factor for path edges σ_{path}	3	3	–
safety factor for neighbouring edges σ_{NE}	1	1	–
positioning tolerance δ_{pos}	0	0	m

*: as current limit of the reference TaxiBot system [12]

As operational key performance indicators (KPIs), we mainly use the sum of taxi times t_{taxi} and taxi distances d_{taxi} of all flights. However, to show the relative increase in taxi time or distance between different parameter settings, the KPIs are normalised by the lowest value of the tested set. When assessing the real-world layout, we use the taxi time distributions per runway. Furthermore, we track the number of collisions between vehicles during the execution of the planned routes, and ensure that none exist. As search-related KPIs, we use the cost, computational runtime, assigned priority pairs, as well as the amount of explored high-level nodes and low-level states. Additionally, we use the number of operationally obsolete stops exceeding 10 s to assess the secondary planning objective of mitigating stop-and-go movements (see Section 3.2).

5.1.2 Benchmarking the High-level Search

To benchmark the high-level search of the AS-MAMP algorithm, we compare its *depth-first* search against PBS variants such as Greedy PBS [62] without and with the use of induced constraints (IC), as well as the first-come-first-served (FCFS) scheme often used in more application-oriented studies. To establish an optimal priority ordering, we use a *best-first* search in PBS. Table 5.3 provides a comparison of these when applied to the four test cases. First, we analyse the results when running PBS without IC, followed by searching with IC.

Table 5.3: AS-MAMP benchmark using high-level variants, with cost ratios listed w.r.t. 'best-first'

	case	cost	approx. runtime	prio pairs	high-level nodes	low-level states	extra stops (> 10 s)
AS-MAMP: high-level variants							
best-first	A	4718.1	200 s	11	571	357 103	0
	B	7601.5	43 s	13	247	78 847	0
	C	6089.4	580 s	11	1 529	1 083 529	0
	D	5172.9	110 s	11	452	206 106	0
depth-first	A	100 %	3 s	12	25	10 686	0
	B	100 %	3 s	13	27	8 067	0
	C	102 %	7 s	15	31	16 658	0
	D	105 %	4 s	14	24	9 386	0
greedy	A	100 %	3 s	9	19	8 875	0
	B	126 %	3 s	8	17	6 496	0
	C	113 %	4 s	9	19	8 356	0
	D	100 %	3 s	10	18	6 217	0
FCFS	A	110 %	1 s	21	1	2 508	0
	B	130 %	1 s	21	1	1 818	0
	C	103 %	1 s	21	1	2 016	0
	D	113 %	1 s	21	1	1 895	0
AS-MAMP: high-level variants with induced constraints (IC)							
best-first with IC	A	100 %	240 s	13	627	416 354	0
	B	100 %	87 s	13	415	145 138	0
	C	100 %	750 s	11	1 757	1 348 195	0
	D	100 %	110 s	12	468	198 606	0
depth-first with IC	A	100 %	5 s	13	27	11 829	0
	B	100 %	7 s	13	27	7 930	0
	C	114 %	14 s	18	37	20 479	0
	D	107 %	19 s	13	53	24 077	0
greedy with IC	A	101 %	5 s	10	21	10 875	0
	B	102 %	9 s	10	21	10 400	0
	C	112 %	6 s	10	21	11 595	0
	D	100 %	4 s	10	18	6 829	0

As expected, while the best-first search yields the highest solution quality, it requires a large number of explored PBS nodes, total sum of SIMP states as well as a long runtime. Substantially less time is required for the depth-first and greedy searches to find a solution, with only a fraction of PBS nodes being explored. Since the runtime correlates with the total sum of SIMP states, the greedy search terminates faster (cases C and D) or slightly faster (cases A and B). Furthermore, due to picking the child-node with the lowest number of remaining conflict pairs, it finds solutions requiring less priority relations. However, in comparison to the solution cost of the best-first search, its solution quality varies more between the cases than with a depth-first search. With a pre-defined priority sequence, the FCFS approach results in the fewest explored SIMP states, and thus the lowest runtimes for all test cases. However, in contrast to the depth-first and greedy searches, it also leads for all cases to suboptimal solutions, ranging between 3 % (case C) to 30 % (case B).

When running PBS with IC, recall that conflict detection returns the conflict pair with the highest number of induced constraints, likely selecting a different pair than running PBS without IC. Despite returning the optimal priority ordering among those explored, also the best-first search does not explore all possible priority pair sequences. Therefore, when using IC, the number of PBS nodes, total sum of SIMP states, and runtime change in general. Furthermore, the conflict free solutions contain a different sequence of priority pairs, and the final count of pairs may differ. While 'best-first with IC' returns identical solution costs for all test cases, the resulting priority sequences indeed differ w.r.t. 'best-first'. Likewise, using induced constraints in the depth-first and greedy searches leads to different solution costs. For test cases C and D, the solution quality degrades using 'depth-first with IC', while it improves for cases B and C when IC is used in the greedy search. Nonetheless, the returned solution quality of 'greedy with IC' continues to fluctuate more than that of the depth-first search. Regarding runtime, using IC seems to increase the number of explored PBS nodes and SIMP states, leading to higher computational times.

Overall, the comparison among the different high-level search strategies without and with IC suggests that the conflict selection procedure can be further optimised to combine the advantages of a fast greedy search with a more consistent solution quality returned by a depth-first search. The runtime could also be reduces with effective partial priority relations handed to PBS, as confirmed by the low runtime of the FCFS approach. In Section 5.1.7, we complement the evaluation of the high-level variants by testing AS-MAMP on a real-world airport layout.

5.1.3 Benchmarking the Low-level Search

To benchmark the low-level search of the AS-MAMP algorithm, we run AS-MAMP with two alternative low-level solvers, each necessitating different additional assumptions:

- SIPP: extends the regular SIPP states to accommodate activity-IDs; motions are instantaneously accelerated and assume constant speed according to the edge's speed limit; agents can hold on nodes between the motions;
- Spacetime-A*: extends the regular A* states to accommodate the arrival time, arrival speed, and activity-IDs; arrival speeds are discretised, with an increment of 1.5 m/s; agents can only hold at nodes after finishing a `Wait` activity, with the hold-time discretised into 10 s increments, and another hold-state only created upon exploring the previous one.

Table 5.4: AS-MAMP benchmark using low-level alternatives (cost ratios w.r.t. 'best-first' in Table 5.3)

	case	cost	approx. runtime	prio pairs	high-level nodes	low-level states	extra stops (> 10 s)
best-first + SIPP	A	90 %	110 s	15	1 293	445 462	14
	B	91 %	15 s	12	217	42 103	5
	C	90 %	250 s	16	2 245	1 025 266	12
	D	86 %	32 s	12	455	141 674	7
depth-first + SIPP	A	90 %	2 s	16	33	7 336	14
	B	100 %	1 s	12	25	4 949	6
	C	94 %	2 s	15	31	9 607	15
	D	87 %	1 s	10	18	4 564	8
greedy + SIPP	A	93 %	1 s	7	15	3 161	16
	B	109 %	2 s	9	19	5 257	5
	C	101 %	2 s	9	19	5 662	24
	D	87 %	2 s	10	18	4 948	7
FCFS + SIPP	A	97 %	< 1 s	21	1	1 107	19
	B	118 %	< 1 s	21	1	1 094	7
	C	93 %	< 1 s	21	1	1 316	19
	D	96 %	< 1 s	21	1	1 188	12
depth-first + Spacetime-A*	A	105 %	220 s	12	25	307 244	0
	B	105 %	140 s	13	27	218 002	0
	C	118 %	470 s	16	33	590 362	0
	D	107 %	110 s	13	24	225 988	0
greedy + Spacetime-A*	A	110 %	180 s	10	21	317 768	0
	B	119 %	86 s	8	17	187 079	0
	C	110 %	440 s	11	23	542 321	0
	D	107 %	79 s	11	21	185 523	0
FCFS + Spacetime-A*	A	116 %	35 s	21	1	69 083	0
	B	128 %	25 s	21	1	52 343	0
	C	109 %	52 s	21	1	69 815	0
	D	122 %	28 s	21	1	60 374	0

Table 5.4 lists their results when applied to the four test cases in combination with the high-level variants from the previous section. In comparison to SIMP, the adapted SIPP solver requires less low-level states to find a solution, but often needs to explore more high-level nodes. Nonetheless, it has a lower runtime for all test cases across all examined high-level variants. Furthermore, SIPP appears to return a better solution quality in all cases. However, this is a result of the instantaneously accelerated motions, leading to trajectories that are impossible to be executed in real-world airport operations. Moreover, the computed trajectories require frequent stop-and-go movements with holding times exceeding 10 s. In contrast, the Spacetime-A* solver produces more realistic trajectories due to the same kinodynamic agent properties applied as in SIMP. However, its search is highly inefficient, requiring a large amount of low-level states to find solutions due to its large search space. To conclude, the low-level benchmark emphasises that SIMP combines the advantages of returning realistic and operationally-efficient 4D trajectories from a large search space with the low computational runtimes of SIPP-based search.

5.1.4 Influence of Key Operational Parameters

In this subsection we analyse how the acceleration and deceleration, minimum speed, cost-factor c_d in the cost function in SIMP, and positioning tolerance δ_{pos} influence the KPIs. To this end, we simulate the four synthetic test cases with changing values for each of these parameters. All simulations are run with a best-first search without IC to yield the highest solution quality.

The choice of acceleration and deceleration has a strong impact on the taxiing duration, as shown exemplary for case A in Figure 5.3: the higher the acceleration and/or deceleration, the faster the aircraft are able to complete their taxiing. In contrast, the influence on the travelled distance is minimal. This trend holds for the other cases as well. Thus, with the values of $acc = 0.25\,\mathrm{m/s^2}$ and $dec = -0.75\,\mathrm{m/s^2}$, approximately 10 % higher taxi times result in comparison to the highest considered absolute value of 2 m/s² for both parameters.

Since the maximum speeds are operational and/or technological limits, we did not vary them. Nonetheless, it is self-evident that changing these values will affect the taxiing duration. When varying the minimum speed v_{min} between 0.5 m/s to 5.0 m/s, the taxi time increased by more than 7 % for $v_{min} \geq 2.0\,\mathrm{m/s}$ in test case D. While case C showed a small increase up to 1 % at $v_{min} = 5.0\,\mathrm{m/s}$, test cases A and B were unaffected by the different minimum speeds. This suggests that the AS-MAMP algorithm is capable of creating trajectories without the need for agents to stop at arbitrary locations to resolve conflicts, as long as $v_{min} \leq 1.5\,\mathrm{m/s}$. The operational benefit is that pilots have to apply breakaway-thrust less often, resulting in fuel savings and reduced engine-blasts that lower the associated risks.

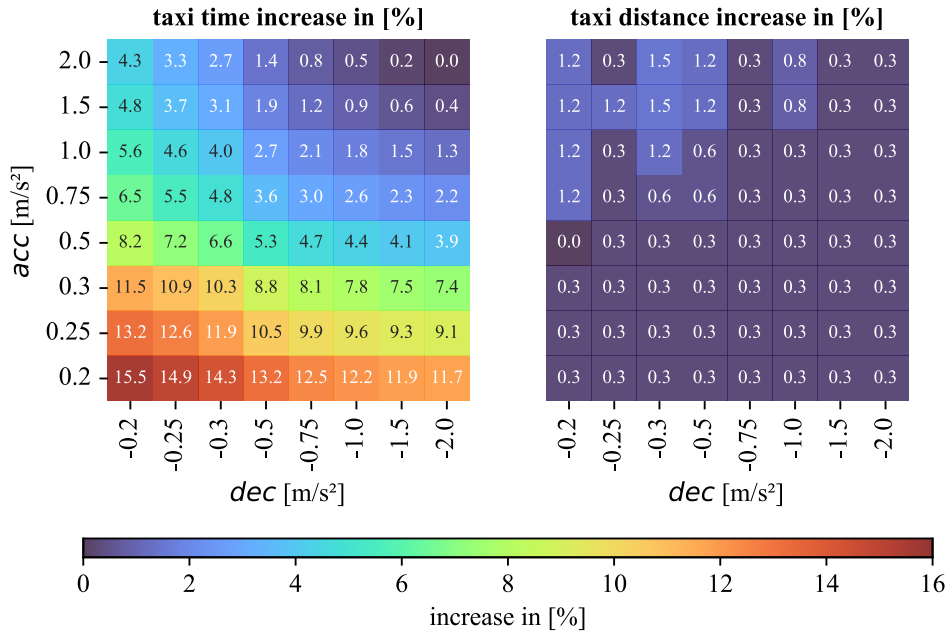


Figure 5.3: Sensitivity analysis of case A depicting how acceleration acc and deceleration dec influence the taxi time and distance, shown as increase in [%] w.r.t. the lowest sum over all flights.

Regarding the cost function in SIMP (see Equation (4.1)) our hypothesis was that small values of c_d have a positive impact on the KPIs: while for $c_\mathrm{d} = 0$ the algorithm may choose long detours to save a few seconds in taxi time, slightly including the travelled distance in the objective function should mitigate these. In contrast, high values for c_d should result in shorter paths but increasingly long taxi times, since the secondary objective of optimising for distance may outweigh the primary one of low taxi times. When varying c_d between 0 s/m to 0.5 s/m, the hypothesis is supported: setting $c_\mathrm{d} = 0$, the taxi distance increases by 1.7 % for test case A and 3.1 % for test case C, while cases B and D are unaffected. For $c_\mathrm{d} = 0.5$, especially case D is affected: both taxi time and taxi distance increased by 3.9 % and 0.8 %, respectively. For cases A and C, the taxi distances were unaffected, and the taxi times showed a small increase of 1.7 % (case A) and 3.1 % (case C).

Increasing the positioning tolerance δ_pos of each agent from 0 m to 50 m, showed a negligible influence on test cases A and B. In contrast, the taxi time abruptly increased to $>$ 6 % for $\delta_\mathrm{pos} >= 20$ m in test case C, and > 4 % for $\delta_\mathrm{pos} >= 10$ m in test case D. In both cases, the taxi time further increased with rising positioning tolerances adding around 2 % at $\delta_\mathrm{pos} = 50$ m, while the increase in taxi distance remained < 0.8 %. This suggests that higher positioning tolerances likely impact confined passages in airport layouts. Nonetheless, the overall impact on the KPIs should be reasonably small for $\delta_\mathrm{pos} < 20$ m.

5.1.5 Influence of Planning Window

As the flight schedules of the test cases described in Section 5.1.1 contain merely 7 flights, a single planning round sufficed to route all aircraft in the simulations conducted up to this point. To assess how the AS-MAMP algorithm deals with continuous schedules on the synthetic layout and thus the necessity to re-plan, we extended the flight schedules in Table 5.1 to over 90 minutes by repeating the seven flights: for each replicate, the parameter t_cycle is added to the fictive reference time T. Table 5.5 provides an overview of the number of replications and the resulting flights of the flight schedules created for different values of t_cycle.

Table 5.5: Extending the flight schedules from Table 5.1: each base-schedule is repeatedly appended with a time shift of t_cycle

t_cycle	replications of schedule in Table 5.1a	resulting flights
15 min	6	42
12 min	8	56
10 min	9	63
9 min	10	70
8 min	12	84

For the analysis, we varied w_p between 10 min to 120 min and h_p between 20 % to 80 % in relation to w_p. With the number of agents exceeding the feasibility of a best-first search, we used a *best-of* depth-first search as slight modification in PBS: after finding the first solution, we store it, let PBS pick the lowest-cost search-node, and continue the depth-first search for an alternative solution. This procedure is repeated until a maximum number of solutions (in this case 24) is found, returning the one with the lowest cost.

For test case A, the average taxi time remained unaffected. Likewise, for test cases C and D, the average taxi time did not change significantly for all considered values of w_p and h_p when $t_\mathrm{cycle} \leq 10\,\mathrm{min}$. With $t_\mathrm{cycle} > 10\,\mathrm{min}$, the simulations showed stronger taxi time variations across varying w_p and h_p while increasing in general for lower t_cycle. This is further discussed in Section 5.1.8. Based on the chosen t_cycle values, all simulations of test case B failed: subsequent aircraft were spawned at stands still occupied with holding aircraft, due to the confined bay area of the synthetic layout combined with the long engine-start times of up to 6 min. However, a separate analysis with $t_\mathrm{cycle} \geq 22\,\mathrm{min}$ showed results similar to those of the other test cases.

Regarding the total planning time measured as sum over all planning-rounds, the expected trend is observed for all test cases, aligning with the observations from our previous study [132]. For smaller w_p, AS-MAMP has to assign fewer priority-relations since conflicts occurring beyond w_p are ignored, reducing the overall runtime. Regarding h_p, more frequent replanning leads to an increase in total planning time. In summary, the windowed planning approach greatly improves the runtime performance of the AS-MAMP algorithm without sacrificing its solution quality.

5.1.6 Computational Efficiency and Scalability of AS-MAMP Algorithm

Using the 200 simulations conducted per test cases A and C for the sensitivity analysis of the continuous schedules presented above, we study the computational efficiency and scalability of the AS-MAMP algorithm in this subsection. The simulations result in 3 854 planning rounds in total. Per planning round, the traffic load as well as the traffic situation vary, depending on the time point of replanning determined by w_p and h_p of the respective simulation. For the following analysis, we use the performance data of the first conflict-free solution that PBS found in each of the planning rounds, visualised through multiple log-log plots in Figure 5.4.

We can decompose the total time that the AS-MAMP algorithm requires to find a conflict-free solution into three component runtimes: for creating the initial trajectories in the PBS root-node, for detecting conflicting agent-pairs throughout the PBS search, and for subsequently resolving each conflict by invoking SIMP. The upper part of Figure 5.4 shows this decomposition in dependence of the size of the agent population across all planning rounds for case A and case C, respectively. For both test cases, finding the initial trajectories has almost a linear time complexity, matching the expectation following from calling SIMP n-times to find n agent trajectories. The effort to detect conflicts between the trajectories increases with the number of agents to be routed concurrently. For conflict detection, we use an exhaustive search as noticeable in its pseudocode provided in Algorithm 2 in Section 4.3.5. This approach leads to an almost quartic time complexity, resulting in the largest increase in the AS-MAMP runtime for growing agent populations. Thus, for a large number of agents, the time required by the conflict detection mechanism predominantly drives the total runtime to find a conflict-free solution.

While the conflict resolution for test case A has an almost linear time complexity, it increases beyond a quadratic time complexity for case C, and varies more between planning rounds with equal agent population. As shown on the bottom left in Figure 5.4, a comparison of the necessary number of assigned priority-pairs in PBS to find a conflict-free solution reveals a similar trend: for equal agent populations, test case C shows higher values

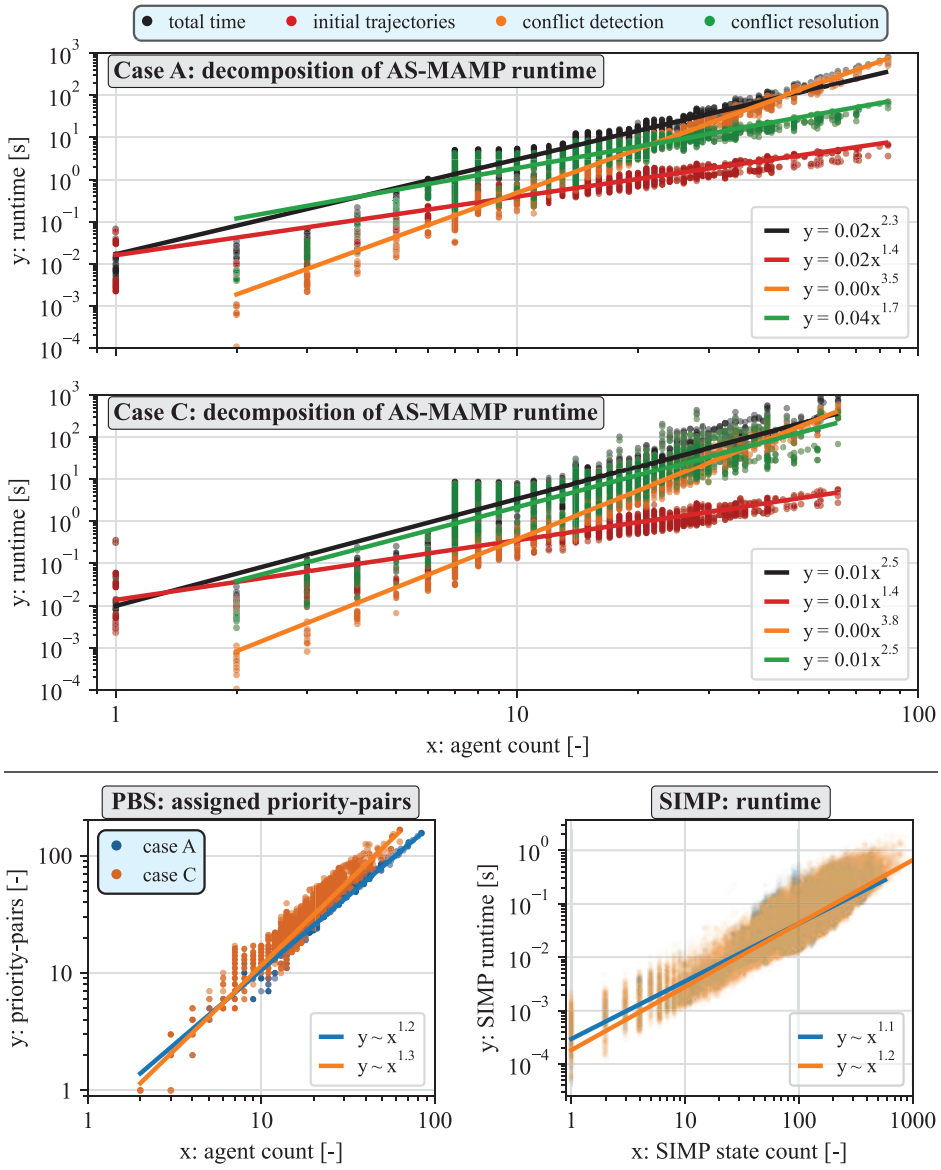


Figure 5.4: AS-MAMP computational performance based on 200 simulations per test cases A and C, resulting in total in 3 854 planning rounds with approx. 695 k SIMP-calls

that vary more between planning rounds. In general, this means that more PBS nodes need to be explored with SIMP being called more often, leading to an increase in total time.

To generate initial as well as updated agent trajectories across the 400 simulations, the SIMP algorithm is called approx. 695 k times in total. As visualised in the bottom right plot in Figure 5.4, it has an almost linear time complexity over the number of generated states, regardless of the test case. Furthermore, it requires approx. 120 states and 0.07 s to calculate an agent trajectory that adheres to the issued activity sequence and constraints set for the agent's route on the exemplary layout.

5.1.7 Testing AS-MAMP on Real-world Airport Layout

Turning to a real-world airport layout, we examine the scalability of the AS-MAMP algorithm based on previous operational studies as mentioned in Section 5.1.1. In the following, we evaluate the different high-level variants benchmarked in Section 4.1 for both MET and TET operations, based on simulations with $w_\mathrm{p} = 20\,\mathrm{min}$ and $h_\mathrm{p} = 10\,\mathrm{min}$. Figures 5.5 and 5.6 provide the resulting taxi time distributions as boxplots per runway for both arriving and departing flights, respectively for MET and TET operations. Furthermore, the taxi time distributions of the high-level searches 'depth-first', 'greedy with IC', and 'FCFS' are compared against the historical operations in Figure 5.7.

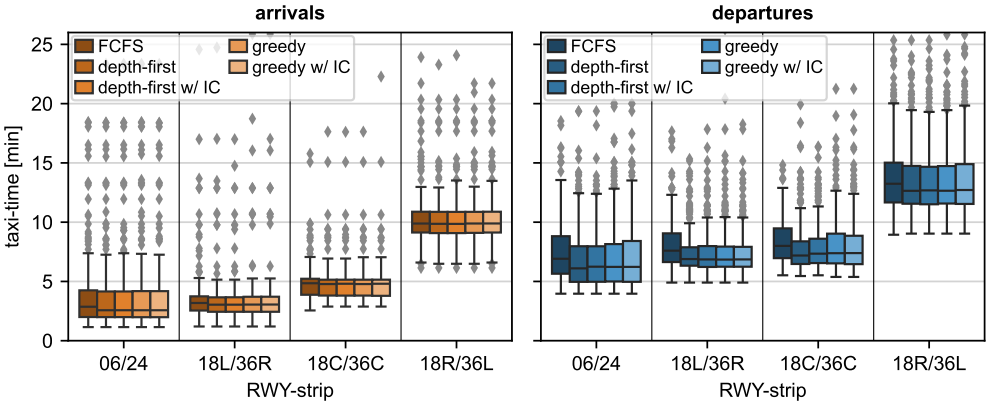


Figure 5.5: Box-and-whisker plots per runway for both inbound and outbound flights, showing the taxi time distributions of simulations of multi-engine taxiing (MET) operations using either FCFS or PBS variants as high-level in AS-MAMP

For inbound flights, the different PBS variants show almost identical taxi time distributions across all runways as well as between MET and TET (used only by outbound flights), while FCFS leads to slightly longer taxiing durations. Nonetheless, in comparison to the historical operations, the centralised path planning in AS-MAMP produces consistently lower and less varying taxi times. For outbound flights, the depth-first search outperforms the other variants. Except for FCFS, the alternative high-level variants produce similar distributions, yet with slightly elevated median or interquartile range marked by the boxes in the three figures. In contrast, the FCFS variant results in higher average taxi times with larger variations between agents for both MET and TET operations. However, similar to inbound flights, any of the tested high-level variants shows an overall reduction in taxi time in comparison to the historical operations.

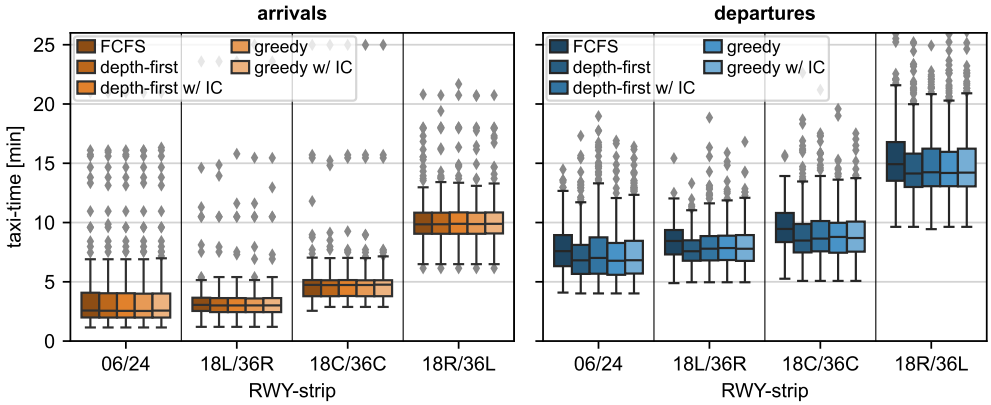


Figure 5.6: Box-and-whisker plots per runway for both inbound and outbound flights, showing the taxi time distributions of simulations of tug-enabled taxiing (TET) operations using either FCFS or PBS variants as high-level in AS-MAMP

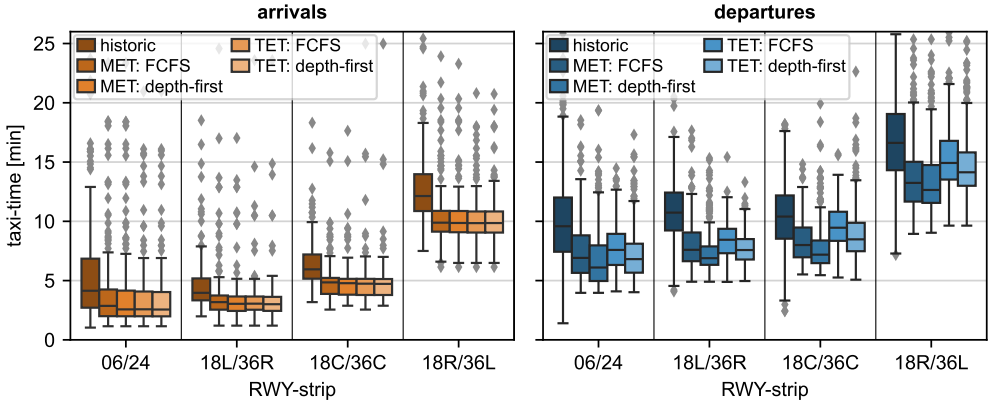


Figure 5.7: Box-and-whisker plots per runway for both inbound and outbound flights, showing the taxi time distributions of historical data, as well as simulations of multi-engine taxiing (MET) and tug-enabled taxiing (TET) operations using either FCFS or depth-first search as high-level in AS-MAMP

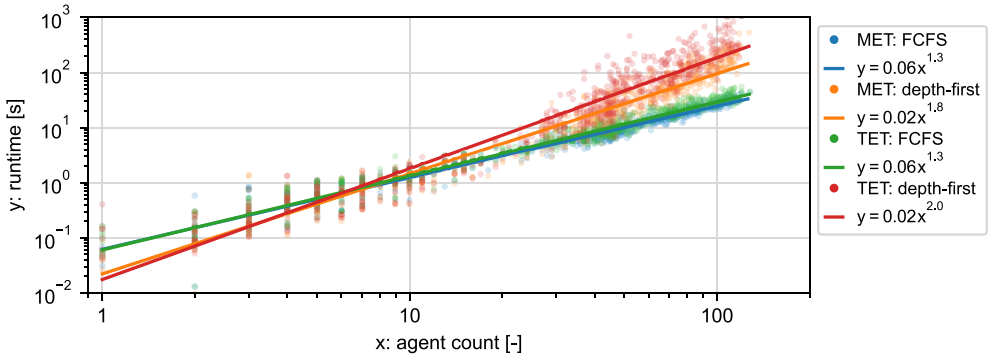


Figure 5.8: Computational runtimes of planning rounds of simulations of multi-engine taxiing (MET) and tug-enabled taxiing (TET) operations using either FCFS or depth-first search as high-level in AS-MAMP

To examine the computational runtime and scalability of AS-MAMP on real-world layouts, we use the data of multiple simulations with differing values of the planning window. The three component runtimes resulting from these simulations have a similar time complexity than those visualised in Figure 5.4. The resulting overall runtime is shown in Figure 5.8 for both MET and TET operations simulated with the depth-first search and FCFS approach as high-level variants. The latter has almost a linear time complexity with increasing agent populations, and thus scales better than any of the tested PBS variants. Nonetheless, the approximately quadratic time complexity of the depth-first search indicates that AS-MAMP scales in general well with respect to both the size of the agent population as well as the complexity of the airport environment.

To summarise, the depth-first without induced constraints showed the best performance across the simulations carried out on the synthetic as well as the real-world layout. In comparison, the FCFS approach often used in application-oriented studies resulted in longer average taxi times of up to 30 % for the synthetic test cases, and 10 % for the real-world scenarios. Moreover, AS-MAMP is more robust than a FCFS approach which more frequently failed to return a solution, especially in high traffic demands. Nonetheless, based on the conducted runtime analyses, the current computational efficiency of AS-MAMP leaves room for improvements, which we further discuss in Section 5.2.

5.1.8 Stress-test: Potential Runway Throughput

In this part, we focus on the potential maximal runway throughput per hour to demonstrate the applicability of AS-MAMP for busy airports. To this end, we fix $w_\mathrm{p} = 30\,\mathrm{min}$ and $h_\mathrm{p} = 15\,\mathrm{min}$, vary t_cycle from 15 min to 7.5 min, and compare multi-engine taxiing (MET, test case A) to tug-enabled taxiing (TET, test case C). Figure 5.9 shows the average taxi times, the maximal runway throughput, and maximal occupancy rate for increasing traffic loads (i.e. lower t_cycle) for both test cases as well as either FCFS or depth-first search used as high-level in AS-MAMP.

For each of the four scenarios, the taxi distance remains unaffected across the tested traffic loads, while the taxi time increases significantly for smaller t_cycle, indicating a rising congestion. However, this also maxes out the achievable runway throughput within 60 min as well as the occupancy rate of the four scenarios. While the simulations of MET operations reach the limit of the occupancy rate of 100 % irrespective of either a 15 min or 60 min frame, the depth-first search results in a higher maximal hourly throughput of 54 flights, with FCFS maxing out at 49 flights.

For TET operations (test case C), the occupancy rate for a 15 min frame shows that its highest value of around 98 % is approached for $t_\mathrm{cycle} \geq 10\,\mathrm{min}$, matching the increase in taxi time. For lower traffic loads, the TET case consistently shows higher runway occupancy rates, while having identical runway throughput compared to the MET case. The hourly occupancy rates are in general higher for FCFS, limiting its achievable runway throughput to 43 flights in comparison to 47 flights with the depth-first search. Overall, this means that the AS-MAMP algorithm cannot optimise the runway sequence of TET as much as in the MET case.

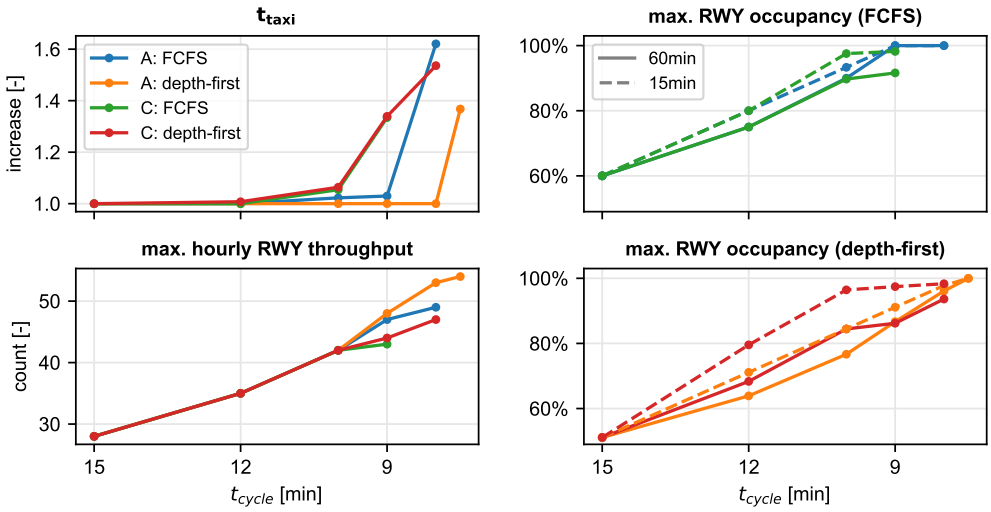


Figure 5.9: KPIs w.r.t. t_{cycle} for continuous schedules of test cases A and C, comparing the high-level variants FCFS and depth-first

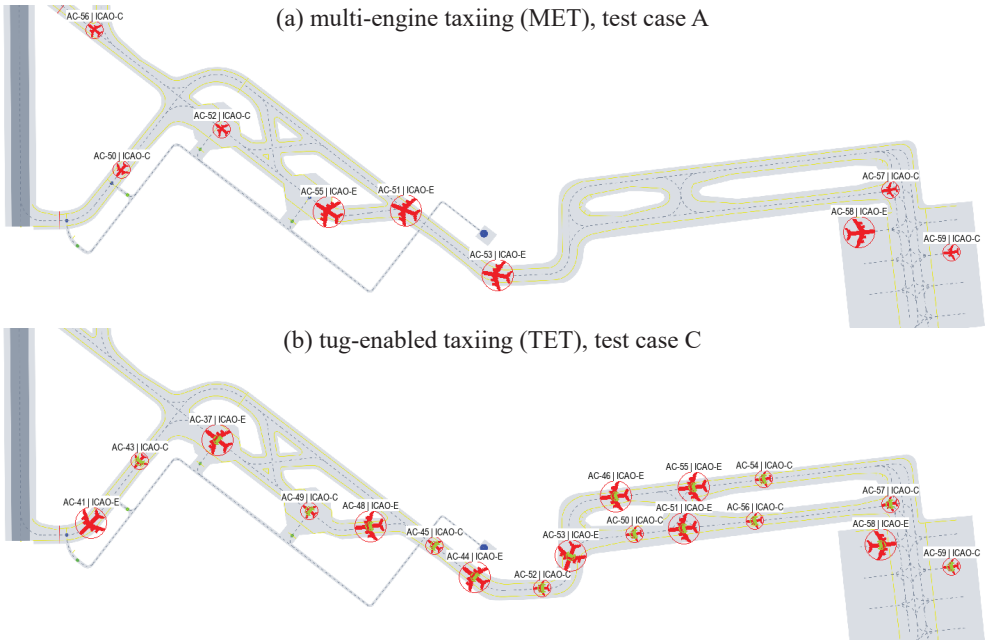


Figure 5.10: Simulation snapshots at $\approx$ 60 min into continuous schedule with $t_{\text{cycle}} = 7.5$ min and depth-first search

This is also reflected in the most congested TET scenario with $t_{\text{cycle}} = 7.5\,\text{min}$: as visualised in Figure 5.10 for a depth-first search, aircraft queue almost until the bay area when the simulation crashes, around one hour into the 90 min flight schedule. Also the choice of simulation parameters plays a role: with $v_{\text{min}} = 1.5\,\text{m/s}$, the AS-MAMP algorithm eventually runs out of route options for the agents, and thus fails to find a solution. In contrast, without the necessity to decouple, minimal queuing occurs in test case A.

As mentioned above, using a depth-first search, the chosen set of parameters results in a maximal runway throughput within 60 min of 54 and 47 takeoffs for MET and TET, respectively. To put this into perspective, Table 5.6 lists the maximal runway throughput of all possible runway configurations for the five largest airports in Europe [135]: the highest throughput per runway is reached by London Heathrow with 54 departing flights per hour. Therefore, when aircraft use MET, the multi-agent system is capable of matching or exceeding the maximal runway throughput seen at these large European airports. For TET, the runway throughput in the simulated setting still exceeds those of most European airports. Note that the potential throughput can be increased when the taxiway network is adapted: with either better accessible or more decoupling locations, the AS-MAMP algorithm would have more options to optimise the takeoff sequence and thus the potential runway capacity.

Regarding the runway throughputs and occupancy rates of simulations on the real-world layout, both MET and TET operations reached similar maximal values of around 46 hourly takeoffs with up to $\approx 80\,\%$ occupancy rates, irrespective of the high-level search in AS-MAMP as well as the runway configuration of the scenarios. In turn, we believe that this validates that the two cases analysed above are suitable stress-tests for AS-MAMP: all scenarios with $t_{\text{cycle}} \geq 10\,\text{min}$ resulted in higher occupancy rates than those found in the historical flight schedules. This underscores that the AS-MAMP algorithm is able to coordinate ASM Ops in an efficient way, and only returns planned trajectories when these are indeed conflict-free.

Table 5.6: Maximal runway throughputs and occupancy rates for departures within 60 min of the 5 largest European airports in (a), as well as scenarios simulated with the AS-MAMP high-level variants FCFS in (b) and depth-first search in (c)

(a) historical data (runway occupancy rate not reported)

airport code	ICAO IATA	EDDF FRA	EHAM AMS	LFPG CDG	EGLL LHR	LEMD MAD
departure runways		2	1	2	1	2
max. throughput		68	46	75	54	70

(b) simulated data: FCFS

layout	EHAM		case A	case C
operations	MET	TET	MET	TET
max. throughput	45	46	49	43
max. occupancy	76 %	79 %	100 %	92 %

(c) simulated data: depth-first

EHAM		case A	case C
MET	TET	MET	TET
45	46	54	47
77 %	80 %	100 %	94 %

5.2 Discussion

The results obtained in the simulation experiments presented above demonstrated the ability of the AS-MAMP algorithm to coordinate agent trajectories safely and efficiently, as backbone of a MAS model to study a next-generation concept for fully-automated surface movement operations with centralised planning. Despite the promising results, many real-world challenges and limitations remain before the ConOps underlying this MAS model could be deployed in live airport operations. In turn, numerous research questions remain to be explored in future work, of which we consider only a few in the following subsections: we first reflect on the aspects related to the AS-MAMP algorithm in Section 5.2.1, and then discuss the limitations of the MAS model in Section 5.2.2. Broadening the perspective, we discuss open research directions regarding the proposed generalised architecture for studying diverse concepts for next-generation airport surface movement operations in Section 5.2.3.

5.2.1 Reflection on the AS-MAMP Algorithm

We formulated the AS-MAMP algorithm as centralised decision-logic to conduct multi-agent coordination for all concurrent ground movements. A fast response time of the AS-MAMP algorithm is crucial for enabling human-in-the-loop experiments with real-time adjustments from human operators, probabilistic delays, and disruptions, and is ultimately necessary for deployment in live operations. Therefore, the AS-MAMP algorithm should provide conflict-free solutions within mere seconds regardless of traffic density, necessitating further improvements to its current computational complexity. The exhaustive search currently deployed as conflict detection mechanism turned out to drive the required time when coordinating the trajectories among large agent populations representative of busy traffic scenarios, and thus needs refactoring or replacement. Parallelisation of the algorithm, e.g. to compute the initial trajectories or to recalculate them during conflict resolution, could further reduce its computational complexity. In turn, the total runtime of AS-MAMP shown in Figures 5.4 and 5.8 would likely remain below the green line, i.e. <1 s to deconflict the routes for up to 20 agents and <10 s for up to 50 agents.

As highlighted by the comparison to a first-come-first-served approach, the conflict selection procedure could be optimised to minimise runtime while preserving a high-quality solution. Despite SIMP having an almost linear time complexity, the number of explored states could be minimised by further improving the heuristics as well as introducing suitable pruning strategies. To this end, future research should also investigate the potential of enhancing both high-level and low-level in AS-MAMP for example with machine learning or reinforcement learning (ML/RL) techniques: while still in its infancy, a growing body of literature explores how these can be combined with and integrated into MAPF algorithms to augment their path planning capabilities as well as help in the execution of the generated routing plans [136]. Given the almost linear computational complexity of SIMP in relation to the number of explored states observed in Figures 5.4 and 5.8, a considerable improvement in runtime is to be expected. In turn, deconflicting up to 50 agents might be achievable within 1 s.

Regarding the evaluation of AS-MAMP, the stepwise approach of first using a simplified, synthetic layout and then shifting to a real-world airport layout proved helpful to benchmark both high-level and low-level of AS-MAMP. To further investigate the efficacy

of AS-MAMP, simulations could be conducted on airports with differing layouts across varying traffic densities. Moreover, other ground movements such as towing traffic as well as other ground vehicles using the taxiways could be included, which is closely linked to the overall model instance and thus further discussed in the next subsection.

In summary, potential directions for future work on the AS-MAMP algorithm include:

- to minimise the runtime of AS-MAMP with enhancements to the conflict detection scheme, the conflict selection and prioritisation mechanism, and to SIMP using e.g. focal search, pruning strategies, or improved heuristics;
- to explore how ML/RL techniques could be utilised in AS-MAMP;
- to extend the evaluation of AS-MAMP to other real-world airport layouts.

5.2.2 Reflection on the MAS Model

In general, developing realistic models for real-world applications such as ASM Ops is a challenging and labour-intensive task. While we elaborated one model instance, simulating all its underlying conceptual aspects is beyond the scope of this thesis. Thus, the ability of the MAS model to generalise to the large variability in airport sizes, their taxiway structures, and specific operational constraints remains untested and requires further validation. Furthermore, additional operations such as towing movements could be included in this MAS model instance.

Despite the promising simulation results, the feasibility of deploying this model instance with the AS-MAMP algorithm as a centralised path planner into live operations remains limited at this stage. The reliance on a single coordination entity introduces potential scalability and fault-tolerance concerns, especially under high traffic density or degraded system conditions. Moreover, the simulations presented in this chapter relied on precise localisation and reliable communication between agents to ensure an execution according to the conflict-free routing plans. However, deviations during execution can occur for example due to mechanical issues, human interventions, or obstacles on the taxiways unknown to the agents in the MAS model. Furthermore, highly dynamic, unforeseen events such as sudden weather changes or emergency reroutings may challenge the resilience of the MAS model, requiring quick actions of agents. As first steps towards a potential real-world integration of the underlying ConOps, both adaptive replanning capabilities as well as empowering the agents to carry out local adjustments need to be thoroughly studied.

In summary, potential directions for future work on the MAS model instance include:

- to assess the operational challenges and bottlenecks faced at different airports through interviews with operational experts;
- to conduct further simulations of differing airport layouts;
- to include additional operations such as towing movements into the MAS model;
- to evaluate its real-time adaptability by implementing the conceived local adjustment mechanisms based on distributed control elements.

5.2.3 Reflections on Generalised Hierarchical-Distributed Modelling Architecture

In general, an important future research direction is to study various decision-making, coordination, and control mechanisms to explore the large design space for next-generation airport surface movement operations. To name one direction, ML/RL techniques could boost the abilities of MAS model instances, for example by providing predictions based on historical airport data that the different agents can use in their coordination, communication, and decision making. While ML/RL techniques could prove helpful in addressing real-world challenges, the implications of using such techniques must be thoroughly studied: for example, as key difference to AS-MAMP that anticipates on conflicts and mitigates deadlocks through its centralised coordination mechanism, implicit approaches such as ML/RL techniques must provably meet the stringent safety requirements of ASM Ops.

Besides, real-world ASM Ops are a complex socio-technical system, and current regulations as well as liability concerns pose constraints on introducing autonomy into the operations. While we recently explored how interactions between human operators and the MAS can be incorporated into such model instances [137], many fundamental questions remain, e.g. the desirable level of automation, the allocation of tasks between humans and automated systems, and questions related to shared situation awareness. In general, human–autonomy teaming must be carefully designed to ensure effectiveness, trust, and accountability, particularly when air traffic controllers remain responsible for safety-critical decisions.

Moreover, real-world airport environments are subject to uncertainties such as weather disruptions, equipment failures, or ad-hoc ground vehicle movements, which demand robust fallback mechanisms and real-time adaptability. On the other hand, integrating automated planning systems into existing air traffic management workflows requires alignment with current operational procedures and regulatory constraints, as well as certification to be provably safe. Overall, we strongly agree with the view expressed in e.g. [31], [138] that human–autonomy teaming is required in an operational setting. As outlined in this chapter, we believe that this should be realised as a hierarchical-distributed system. Connecting to the discussion on the algorithmic runtime in Section 5.2.1, distributed coordination schemes could keep the number of agents per coordination task well below 20. Since the envisioned runtime improvements would allow for sub-second calculations of deconflicted trajectories, many of the above mentioned real-world challenges as well as the algorithmic scalability issues could be accounted for when shifting to distributed coordination.

Collectively, addressing these points will require a joint effort between researchers, industry stakeholders, and policymakers. Likely, an iterative approach is necessary to design, test, and refine operational concepts, and to contrast the efficacy of operational ideas. We hope that the generalised architecture proposed in this chapter will support these efforts as well as provide a basis for discussion between the parties involved.

In summary, potential directions for future work regarding the generalised MAS architecture include:

- to instantiate other models that satisfy the modelling requirements provided in Section 2.2, across ConOps addressing e.g. divergent decision-making mechanisms or degrees of control centralisation;
- to incorporate stochastic aspects representing e.g. sensor noise, communication delays, or operational deviations and disruptions into model instances, and evaluate how these affect agent decision-making;
- to explore approaches for adaptive human–autonomy teaming, tested in human-in-the-loop experiments;
- to compare and contrast the operational performances of various model instances to evaluate their potential implications for being introduced into real-world operations, and to synthesise these findings and lessons learned for improvement of the generalised modelling architecture.

5.3 Conclusions

As one of the key contributions of this thesis, we proposed a generalised multi-agent modelling architecture with a hierarchical-distributed structure. Four agent categories (Scheduling Agents, Routing Agents, Guidance Agents, and Moving Agents) are organised in hierarchical layers. Within each category, multiple agents may operate and engage in centralised and/or distributed control mechanisms, informed by communication and coordination within and across the hierarchical levels. Through their perception, all agents can potentially receive relevant data from environmental objects to inform their decision-making processes. Based on the generalised architecture, a large variety of MAS models can be instantiated that are tailored to a specific ConOps, while retaining the overall system structure. Nonetheless, significant gaps remain in demonstrating how such control system models could be adapted to the constraints of live operations, as pointed out in Section 5.2. Therefore, the generalised architecture should be seen as a research tool to structure and analyse models of future operational concepts, rather than a deployable solution in its present form.

We elaborated one model instance based on a ConOps with centralised planning and a fully-automated control system. The centralised Routing Agent issues conflict-free trajectories by assigning priorities between the agents that move concurrently over the airport surface. It shares the routing plans with the distributed Guidance Agents that subsequently instruct the Moving Agents which in turn execute the commands accordingly. As backbone of this model instance, we formalised the decision-logic of the Routing Agent as the Multi-Agent Motion Planning on Airport Surfaces (AS-MAMP) algorithm. This two-level solver uses windowed prioritised planning to compute conflict-free 4D agent trajectories. As ASM Ops are a safety-critical system, we designed the AS-MAMP algorithm to return a solution only when it is certainly conflict-free. Moreover, as key contribution to the MAPF-field, we showed how the SIPP algorithm [60] can be enhanced to work with finite accelerations as well as upper and lower speed restrictions of agents. To this end, we introduced the novel Safe Interval Motion Planning (SIMP) algorithm that is used as low-level solver in AS-MAMP.

We evaluated the AS-MAMP algorithm using multiple scenarios on both synthetic and real-world airport layouts: we benchmarked its high-level search using PBS variants and the first-come-first-served (FCFS) approach often used in application-oriented studies, as well as SIMP as its low-level search against SIPP and A* with kinodynamic constraints. The conducted simulations highlighted that AS-MAMP scales with approximately quadratic time complexity, with SIMP scaling alomst linearly in time, also for the tested traffic scenarios at Amsterdam Airport Schiphol with up to 120 hourly movements. Finally, we showed that the attainable maximum hourly runway throughputs match or exceed those at the largest European airports. Furthermore, contrasting PBS with FCFS underscored that the latter trades fast solutions for up to 30 % loss in efficiency as well as 10 % attainable runway throughput.

Overall, the obtained simulation results demonstrate the effectiveness of the proposed MAS model with centralised path planning in enabling safe, efficient, and scalable autonomous surface movement operations under realistic airport conditions, making it a viable benchmark model for studying next-generation airport surface movement operations.

This concludes the development phase of the thesis (Chapters 2 to 5), in which the modelling requirements, MAS architecture, and AS-MAMP algorithm were established and benchmarked. As part of the validation phase, Chapters 6 to 8 focus on operational studies, starting with an investigation of fully-automated ASM Ops under multi-engine taxiing (RQ 4). To support this case study, the next chapter refines the MAS model and calibrates key movement parameters. This provides a far-term operational outlook under the assumed automation capabilities.

6

Autonomous Multi-Engine Taxiing Operations

In this chapter, the operational implications are explored that result from fully-automated airport surface movement operations modelled with high realism. The underlying MAS model further enhances the one presented in Section 3.2. As brief summary, the centralised Airport Operations Agent handles the flight schedule and runway configuration, while the centralised Routing Agent computes conflict-free trajectories. The Guidance Agents instruct and monitor the Aircraft Agents while these execute the planned routes. The model incorporates the decisive processes and elements of airport surface movement operations such as pushback, engine-start, inbound and outbound holding, compliance to CTOT-slots, and wake turbulence separation for takeoffs. Recall from Chapter 4 that the AS-MAMP algorithm accounts for different sizes of aircraft and minimum safety distances between them. In this chapter, the speed limits in curves are calibrated with historical ADS-B data. Using the real-world flight schedules of two of the busiest days at Amsterdam Airport Schiphol, including different runway configurations, we examine the performance of the fully-automated taxiing system with respect to the historical operations. Under the considered simulation conditions, conflict-free routing within the MAS model reduces taxi times by 30 %, decreases their variability, improves predictability, and increases runway capacity.

An initial version of this chapter was published in:

M. von der Burg and A. Sharpanskykh, 'Multi-agent planning for autonomous airport surface movement operations', in *SESAR Innovation Days 2023*, 2023, pp. 1–9. DOI: 10.61009/SID.2023.1.27.

An extended version, as presented in this chapter, was published in:

M. von der Burg and A. Sharpanskykh, 'Modelling and analysis of autonomous airport surface movement operations based on multi-agent planning', *European Journal of Transport and Infrastructure Research*, vol. 25, no. 1, pp. 1–23, 1 2025, ISSN: 1567-7141. DOI: 10.59490/ejtir.2025.25.1.7459.

The air traffic demand is predicted to exceed 10 billion yearly passengers by 2050 [1], more than twice the amount of 2019. However, it is expected that infrastructural expansions of airports are insufficient to facilitate this growth [8]. Therefore, large airports are facing challenges to improve the efficiency of their operations and, on top of that, to reduce their environmental footprint to achieve the industry-wide goal of net-zero emissions by 2050 [1].

When the congestion at airports increases, the taxi time of an aircraft, i.e. the time that it travels over the airport surface from runway to gate or vice versa, becomes harder to predict. This may affect the respective flight, but may also lead to network-wide knock-on effects [10]. Moreover, when Air Traffic Control Officers (ATCOs) have to handle more and potentially less predictable traffic, their workload is amplified [9]. Consequently, the taxiing operations may become less efficient.

To deal with these issues, previous and ongoing SESAR projects considered how to increase the efficiency and predictability of taxiing operations through automation in general [139], [140]. Other projects examined more specifically how to reduce emissions by integrating engine-off taxiing techniques [34], or how to enable human–automation teamwork in the operations through higher levels of automation [35]. Moreover, different aspects of airport surface movement operations (ASM Ops) were studied in previous work: for example, [6], [141] studied automation within ASM Ops, [71] reviewed optimisation approaches, [142], [143] investigated how to create optimal trajectories under multiple objectives, and [144] assessed metrics to evaluate predictability when automating airport surface movements.

In practice, the Advanced-Surface Movement Guidance and Control System (A-SMGCS) provides specifications for four services to increase efficiency through the use of automation: the surveillance service to track vehicles, the airport safety support service to alert controllers of potential conflicts, the routing service to determine conflict-free trajectories, and the guidance service to guide the vehicles during taxiing [68], [92]. Moreover, towards 2050 and beyond, both EASA and SESAR JU define a vision and roadmap to eventually reach fully-automated air traffic management, i.e. Level 3 in EASA's AI roadmap [18], or Level 4–5 in the roadmap of SESAR JU [145]. However, the challenges to achieve this long-term vision are manifold. For instance, the role of the human during and beyond the transition as well as the implications of such fully-automated operations remain largely unknown.

In this chapter, we explore which operational consequences may result from fully-automated surface movement operations at large airports: in terms of efficiency, predictability, delays on taxiways, and runway capacity, while sustaining safety levels. We represent such operations as multi-agent system (MAS) model that plans conflict-free routes for all aircraft on the ground and controls their execution. Multi-agent systems modelling and simulation allow for inherent modularity, flexibility, and expressiveness of a system's structure and dynamics. Both heterogeneous agent properties and different types of interactions between actors as well as randomness can be integrated into an agent-based model [46].

However, to evaluate the operational impact of such a new operational concept, the model must represent the airport surface movement operations realistically. As pushback and engine-start operations considerably affect delays in planning systems [146], we explicitly model these processes in the MAS control model. Furthermore, we include inbound and outbound holding as these can considerably affect the taxi times of the

respective flights: some arriving aircraft must wait at a remote holding location until their stand is free, while some departing aircraft must be delayed to comply with the Calculated Take-Off Time (CTOT) issued by Eurocontrol. Moreover, the taxi times are strongly influenced by the chosen agent kinematics as pilots speed up on straight segments, and slow down in front of curved taxiways. To this end, we define a maximum and minimum speed, and use speed limits along curves dependent on the radius of curvature that we calibrate using historical ADS-B data. To the best of our knowledge, this is the first study to use such detailed kinematics to compute trajectories of taxiing aircraft. We thus provide an overview of values found in the literature when describing the model calibration in Section 6.1.4.

Airport surface movement operations must be safe at all times. To plan conflict-free routes for a set of agents, many different multi-agent path finding (MAPF) algorithms have been developed that model agents as a point [102]. However, to include agent shapes and kinematics in path planning, the domain of multi-agent motion planning (MAMP) offers more suitable algorithmic concepts [95]. We combine and extend such state-of-the-art MAMP algorithms to form a routing algorithm that addresses the requirements of airport surface movement operations. Section 6.1.3 summarises how it computes conflict-free trajectories for all aircraft.

With an implementation of the MAS model in Python, we simulate the fully-automated taxiing operations using the flight schedule of two of the busiest days at Amsterdam Airport Schiphol to date. Section 6.2 outlines the experimental setup. We then analyse and discuss key performance indicators in relation to their historical counterparts in Section 6.3, and end with concluding remarks in Section 6.5. With an implementation of the MAS model in Python, we simulate the fully-automated taxiing operations using the flight schedule of two of the busiest days at Amsterdam Airport Schiphol to date. Section 6.2 outlines the experimental setup. We then analyse and discuss key performance indicators in relation to their historical counterparts in Section 6.3, list directions for future work in Section 6.4, and end with concluding remarks in Section 6.5.

6.1 Multi-Agent System Model

The developed multi-agent system model for fully-automated aircraft taxiing operations has a hierarchical-distributed structure of both centralised and distributed agents, which is illustrated in Figure 6.1. The centralised Airport Operations Agent defines and updates the flight schedule and runway configuration, the centralised Routing Agent plans conflict-free trajectories for all Aircraft Agents which are instructed and monitored by distributed Guidance Agents while executing their planned routes. For the fully-automated operations considered in this chapter, it is assumed that the full control and decision making is done by the agents. Furthermore, we assume that digital means of communication via a datalink such as AeroMACS as well as the surveillance service of the A-SMGCS specification are fully operational.

6.1.1 Model specification

The airport taxiing infrastructure is represented by a graph G = (V, E) comprising vertices V and directional edges E. As example, the layout of Amsterdam Airport Schiphol, which is also used in the simulations presented in this chapter, is shown in Figure 6.2. Vertices

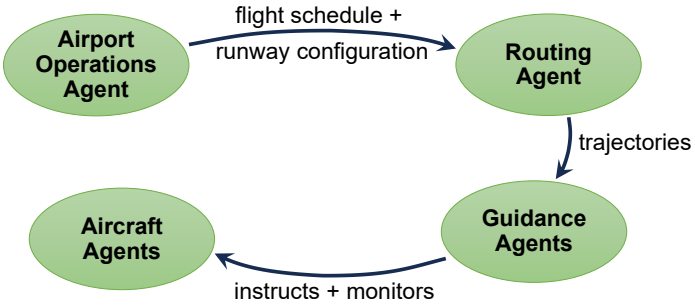


Figure 6.1: Overview of multi-agent system for fully-automated airport surface movement operations

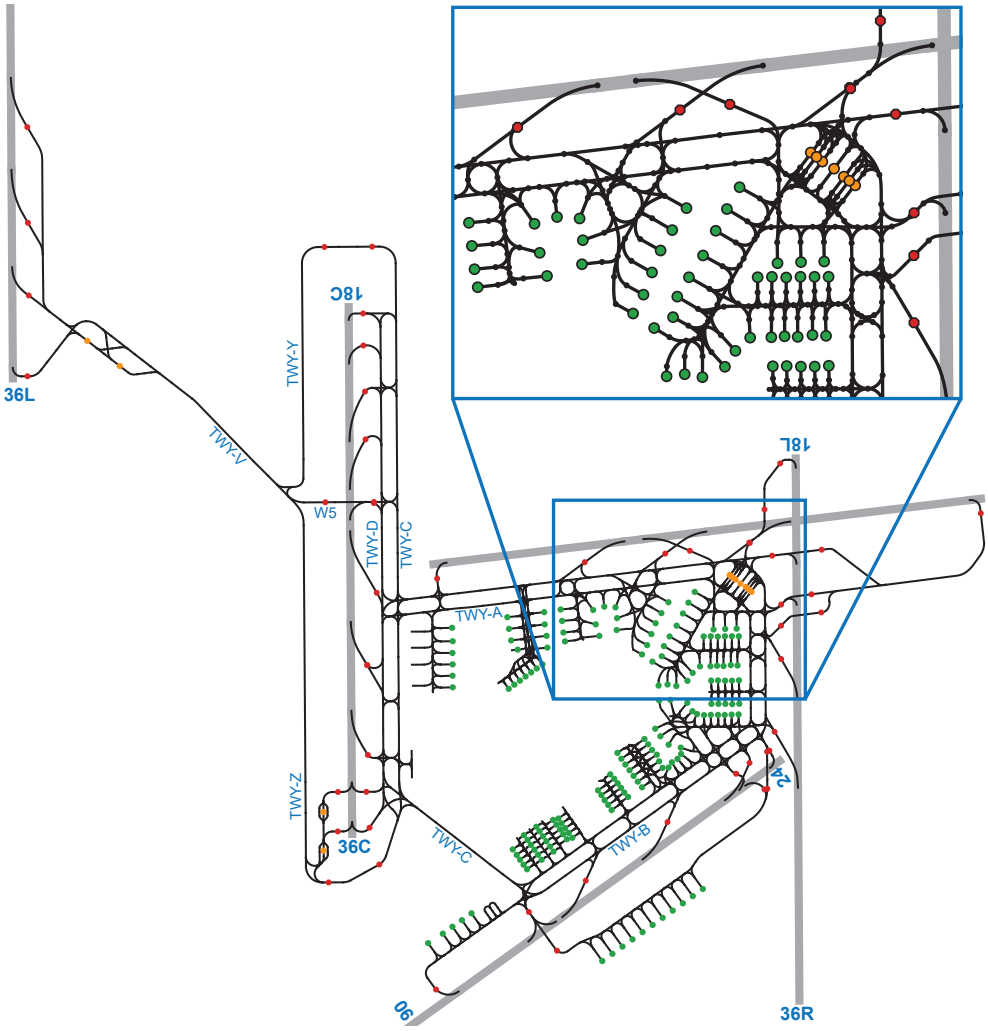


Figure 6.2: Graph of layout of Amsterdam Airport Schiphol with edges for runways (grey) or taxiways (black), and vertices for taxiway intersections (black), stopbars (red), holding points (orange), or gates (green)

denote aircraft stands (green), taxiway intersections (black), holding points (orange), or stopbars in front of runway entries (red). Each bidirectional taxiway segment between two vertices is constructed from two unidirectional edges that connect the vertices. Taxiway edges (black) are constructed using Bézier curves that closely match the taxiway centrelines from a satellite image of Schiphol[1].

The Airport Operations Agent schedules all flights, and updates them whenever new predictions of the underlying A-CDM milestones are available. When the allocated stand of an arriving flight is still occupied by a departing aircraft, or Eurocontrol issued a Calculated Take-Off Time (CTOT) for a departing aircraft, the agent marks the corresponding flight. Such flights are subject to special routes assigned by the Routing Agent to account for the necessary holding, detour, or prioritisation during taxiing, as outlined in Section 6.1.2. Furthermore, the Airport Operations Agent defines the runways in use, i.e. the runway mode of operation (RMO). Active runways and the resulting flight path of arriving or departing flights must not be crossed. Thus, the Airport Operations Agent blocks such taxiway segments by setting layout constraints on them. This mechanism is also applicable for taxiway segments that are temporarily unavailable.

Both the flight schedule and constraints are shared with the Routing Agent that computes conflict-free routes for all taxiing aircraft within the upcoming planning window w_{p}. It re-computes the routing plans when it receives updates from the Airport Operations Agent, or latest after the replanning period h_{p} has passed. We use motion planning to account for vehicle kinematics and shapes in planning. To ensure conflict-free paths, we deploy a two-level search based on Priority-Based Search (PBS) [61] with an augmented version of the Safe Interval Path Planning (SIPP) algorithm [60]. This routing algorithm is presented in Section 6.1.3, and we describe how we calibrate the model in Section 6.1.4.

The resulting trajectories are sent to the Guidance Agents which are positioned at every intersection in the taxiway system. Each Guidance Agent controls those Aircraft Agents that are moving towards its location. It instructs them to execute the next part of the planned trajectories, and monitors that the instructions are carried out accordingly. To do so, the Guidance Agents use the airport radar, which reports the position, speed, and heading of all Aircraft Agents while they move over the airport surface. In case the executed movements deviate from the planned routes, the Guidance Agents locally adjust the trajectories to minimise these deviations. However, when the impact becomes too extensive, they request central replanning from the Routing Agent. Once one of the Aircraft Agents has passed the location of a Guidance Agent, it passes the guidance responsibility for that aircraft to the next Guidance Agent along the aircraft's route.

Aircraft Agents represent the aircraft (auto-)pilots and are modelled to be fully cooperative: they thus carry out the instructions as accurately as possible. To account for the different sizes of aircraft, all flights are categorised as one of the 6 aircraft types from the ICAO aerodrome reference codes [69]. They are assumed to have a circular shape with a pre-defined radius according to the type. Table 6.2b in Section 6.2 lists these shape radii.

When planning the trajectories, a safety zone is added around all agents. To this end, we define a general safety distance, as well as a safety distance that an agent has to keep when it is trailing another aircraft. Both safety measures are defined in relation to the shape radii of the corresponding pair of agents. Moreover, two aircraft that consecutively take off from

[1]The Bézier curves are constructed using the X-Plane WorldEditor WED 2.5 [147].

the same runway must have a minimum separation to mitigate the wake turbulence of the preceding aircraft. We use the time-based separation minima from RECAT-EU for that [133].

6.1.2 Activity sequence of Aircraft Agents

To take the various surface movement operations into account during path planning, the route of an Aircraft Agent is expressed as a combination of the following three activities:

- GoTo activities have one start vertex and a set of goal vertices. Thus, the routing algorithm gets two degrees of freedom: the path between the vertices, and the time to traverse this path. The regular taxiing between one point to another point at the airport is an exemplary GoTo activity.

- `Follow` activities comprise a predefined ordered list of edges that must be part of the route. Therefore, during routing, time is the only remaining variable as the path cannot be changed. Pushback and push-pull manoeuvres of departing aircraft are examples of such.

- `Wait` activities define a vertex at which an agent has to wait for a fixed duration. For instance, a `Wait` activity is used to specify the place at which the pushback tractor is decoupled from the aircraft, or the necessary direction-switch of the push-pull manoeuvre within the pushback operations occurs.

Using a combination of these activities, the Routing Agent defines an activity sequence for both departing and arriving aircraft, as depicted in Figure 6.3.

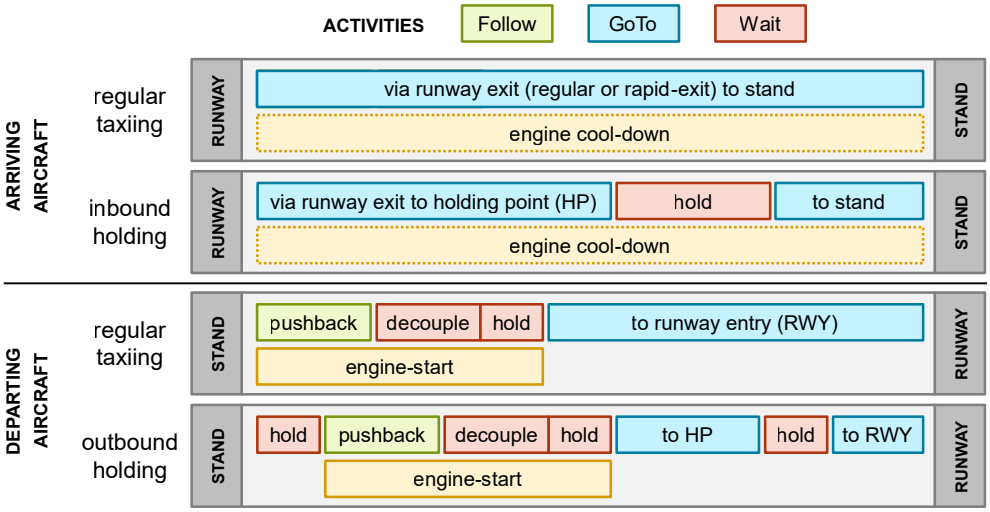


Figure 6.3: Activity sequences for regular taxiing of arriving and departing aircraft as well as inbound and outbound holding; while engine-start (orange box) of departing aircraft is accounted for, engine cool-down (dotted orange box) is neglected

Engine warm-up and cool-down

In the sequence, the warm-up and cool-down of the engines represent special cases. The routing algorithm takes the warm-up phase as part of the engine-start manoeuvre and on basis of the aircraft-specific engine-start duration as input value into account. Therefore, if this duration exceeds the time needed till decoupling from the pushback tractor, additional waiting in form of holding is added to the route. We do not model engine cool-down, as it does not have an influence on the routing regarding the kinematics, since the engines are switched off after standstill at the gate.

Inbound holding

When an aircraft arrives at the airport, but its stand is still occupied by a departing flight, the Routing Agent has three options to resolve the anticipated stand-conflict: for long conflict durations (case 1), it sends the arriving flight to the remote holding platform (see Figure 6.2). Otherwise, it defines a detour along the taxiways (case 2), or reduces the agent's taxi speed for short conflicts (case 3). To this end, the Routing Agent first calculates the single-agent route directly to the stand, i.e. the trajectory without accounting for other aircraft agents, to estimate the severity of the stand-conflict. Then, it computes a single-agent trajectory via the remote holding points. When this detour is insufficient to resolve the stand-conflict, the Routing Agent assigns the remaining time as remote holding duration (case 1), and updates the agent's activity sequence accordingly. In contrast, when the taxi duration now exceeds the time at which the departing aircraft has cleared the stand (case 3), it keeps the original activity sequence of the agent.

Outbound holding to comply with CTOT-slots

Similar to inbound holding, the Routing Agent deals with departing flights for which Eurocontrol issued Computed Take-Off Times (CTOT-slots). However, as long as no arriving flight requires the stand, it assigns a holding duration at the agent's stand so that the agent arrives at the runway at the beginning of the CTOT-slot. In case an arriving flight is scheduled for the stand, the Routing Agent sends the departing flight to a remote holding location close to the scheduled runway. It updates the activity sequence of the departing flight accordingly.

6.1.3 Routing algorithm

The Routing Agent carries out multi-agent motion planning for all Aircraft Agents that taxi within the planning window. This two-level routing algorithm uses a low-level search to calculate individual trajectories per aircraft, and coordinates all agents in its high-level search to yield conflict-free trajectories. For the low-level planner, we extended the Safe Interval Path Planning (SIPP) algorithm [60], and adapted the Priority-Based Search (PBS) algorithm [61] to serve as high-level solver.

PBS carries out multi-agent coordination by assigning a priority order between agents to deconflict their space-time trajectories. To this end, PBS constructs a binary priority tree: first, individual agent trajectories are planned by the low-level search, and are stored as the PBS root-node along with the sum-of-cost of all agent trajectories. We define the cost of a trajectory as sum of the taxiing duration and travelled distance. Then, all conflicts between the trajectories of each agent pair are detected, and one of these conflicting

agent pairs is selected (we use the pair of the first conflict in time). To resolve this conflict, a priority-relation between the conflicting pair of agents is established by creating two PBS child-nodes. In each, one of the agents is given priority, and the other agent must give way along its entire trajectory. Per child-node, the low-level solver is invoked as further outlined below, returning a new trajectory with an updated cost. PBS then continues by following a depth-first search: between the two child-nodes, the one with the lowest sum-of-cost is picked. Any remaining conflicts between those agents that do not yet form a priority-relation with each other are detected, and the conflict selection and resolution scheme is repeated. In case no conflicts remain, PBS can terminate as the established priority order results in conflict-free trajectories, which are returned as the path planning solution.

In the low-level search, the route of a deprioritised agent has to be adapted, either by changing its path or altering the speed profile along the path. To this end, we translate all paths into a set of graph reservations: an aircraft temporarily blocks a set of edges during each movement between one vertex and another. The blockage times and set of blocked edges are dependent on the agent's shape, speed profile, the shapes of other agents, and the safety zone between the shapes.

The SIPP algorithm represents moving obstacles as collision intervals and subsequently defines a set of safe intervals (SIs) per graph location, representing time intervals during which an agent can occupy that location. Furthermore, states are defined on vertices and motion profiles with piecewise constant acceleration map the trajectory between states. We augmented SIPP to facilitate the activity sequence of an aircraft as defined by the Routing Agent, and to take the travelling direction as well as the kinematic agent properties into account. Additionally, we use SIs also on edges to deal with the reservations of agents higher in priority.

In the motion generation, we are bound to the agent's kinematic properties for the current activity and the speed in the current state. A motion that is part of the follow-activity for pushback is for example constrained by a lower maximum speed than regular taxiing in a go-to activity. In addition, vehicles that have maximum speed in the current state, might not be able to decelerate enough to satisfy a reservation on the next edge or vertex. In this case, it might be required to start decelerating on the edge before the current state. To efficiently account for this, we anticipate based on the agent's current speed, braking distance, and reservations or speed restrictions within the braking distance.

6.1.4 Model calibration

In the MAS model, the agents' motions during route planning are modelled based on constant longitudinal acceleration/deceleration and do not account for slip, i.e. are steady-state motions. Nonetheless, to the best of our knowledge, this is the first study to use such detailed kinematics to compute trajectories of taxiing aircraft. In the following, we thus include an overview of related values found in the literature.

We define a general speed limit of 15 m/s in line with the design taxi speed given in the A-SMGCS manual from ICAO [92]. Except for the designated waiting locations, agents must taxi at least with the minimum speed of 1.5 m/s to avoid stop-and-go during taxiing. For curved taxiway segments, the ICAO manual mentions that speeds up to 10 m/s may occur. Most previous studies on airport surface movement operations define curved segments as turns with a maximum speed of 5 m/s [33], [89], [148]. With taxiway curves modelled

explicitly through Bézier curves (see Section 6.1.1), a speed limit v_{curve} is defined per edge:

$$v_{curve} = \sqrt{a_{lat} \cdot r_{curve}} \tag{6.1}$$

with the lateral acceleration a_{lat} and the radius of curvature r_{curve} of the respective edge. To obtain r_{curve}, we use the median value of all curvatures per 1 m-segment of the underlying Bézier curve.

For passenger comfort in public transport, Bae, Moon, Jhung *et al.* [149] provide a range for both longitudinal and lateral accelerations of $\pm 0.9\,\mathrm{m/s^2}$. Furthermore, they claim that a car driver with a normal driving style experiences a lateral acceleration of up to $\pm 4\,\mathrm{m/s^2}$ and a longitudinal acceleration of $-2\,\mathrm{m/s^2}$ to $1.47\,\mathrm{m/s^2}$. In contrast, De Winkel, Irmak, Happee *et al.* [150] found in empirical studies that the acceptable limits for passenger comfort are $1.23\,\mathrm{m/s^2}$ for longitudinal and $0.98\,\mathrm{m/s^2}$ for lateral acceleration. As noted above, previous studies did not consider lateral accelerations to define turn speeds. For longitudinal acceleration/deceleration, different values are reported: for example, $\pm 0.98\,\mathrm{m/s^2}$ [89], [148], or $0.26\,\mathrm{m/s^2}$ as acceleration and $-0.8\,\mathrm{m/s^2}$ as deceleration [33].

To find realistic values for the longitudinal and lateral accelerations of taxiing aircraft, we use historical track data from Schiphol captured by ADS-B receivers that record the aircraft positions during taxiing with a rate of 1 Hz. To this end, we map the positions onto the graph representing the taxiway centrelines, and smooth the resulting trajectories with a Savitzky-Golay filter (window length of 11 s, linear polynomial). This yields the travelled taxi distance along the graph edges as well as the speed and acceleration at each time point of the trajectory. However, we only use the data on the edges that correspond to the main taxiways: while the tracks become too noisy in the bay areas and at aircraft stands, the accelerations on runways for takeoff and landing are not representative of those experienced during taxiing.

In Figure 6.4, the acceleration over speed of each data point is visualised as 2d-histogram. The 1 % and 99 % percentile lines of the acceleration values per 0.5 m/s step show that the longitudinal acceleration/deceleration remain similar across different taxi speeds. Therefore, we set the acceleration to $0.4\,\mathrm{m/s^2}$ and deceleration to $-0.5\,\mathrm{m/s^2}$ independent of an agent's speed. While these values seem low compared to those mentioned in the literature, we argue that using these in planning increases the flexibility during execution: the Guidance Agents have more options to locally adjust the trajectories if necessary (see Section 6.1.1).

Figure 6.5 visualises the historical curve speeds of different aircraft types as average speed along a curved edge with radius r. The average speed is calculated as $\bar{v} = \frac{\Delta d}{\Delta t}$ with the time difference Δt and travelled distance Δd of the data points per edge along each trajectory. Although higher curve speeds exist, we define the speed limit in curves based on a lateral acceleration of $1.5\,\mathrm{m/s^2}$ for small aircraft (left plot in Figure 6.5) and $1.125\,\mathrm{m/s^2}$ for large aircraft (right plot). Using Equation (6.1), we assign a speed limit per edge within the speed bounds of 1.5 m/s to 15 m/s.

Table 6.1 summarises the kinematic values and lists the main algorithmic parameters used by the routing algorithm. In general, two aircraft agents have to keep a minimum safety distance between them equal to the average of their shape radii. However, when an aircraft is trailing another agent, it has to keep a safety distance of at least three times the shape radius of the preceding aircraft, based on discussions with operational experts at Amsterdam Airport Schiphol. The planning window w_p and replanning period h_p are provided as ranges with the requirement that $h_\mathrm{p} < w_\mathrm{p}$.

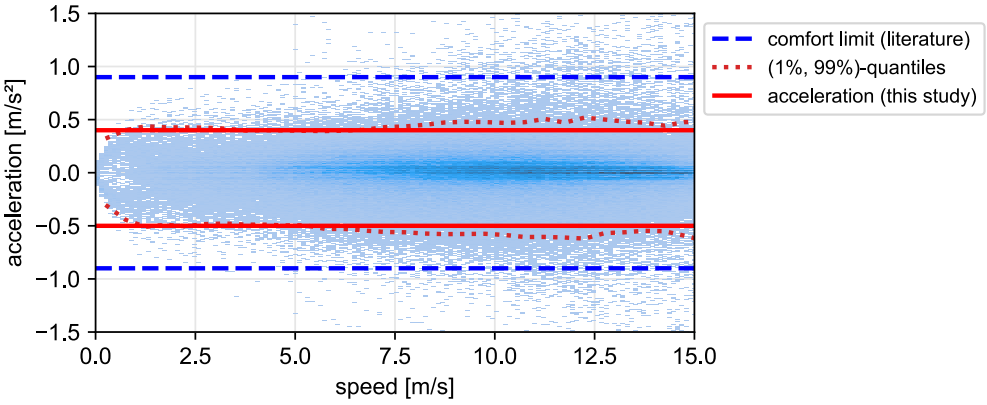


Figure 6.4: Calibration of longitudinal acceleration and deceleration values with historical track data. Comfort limit obtained from [149]

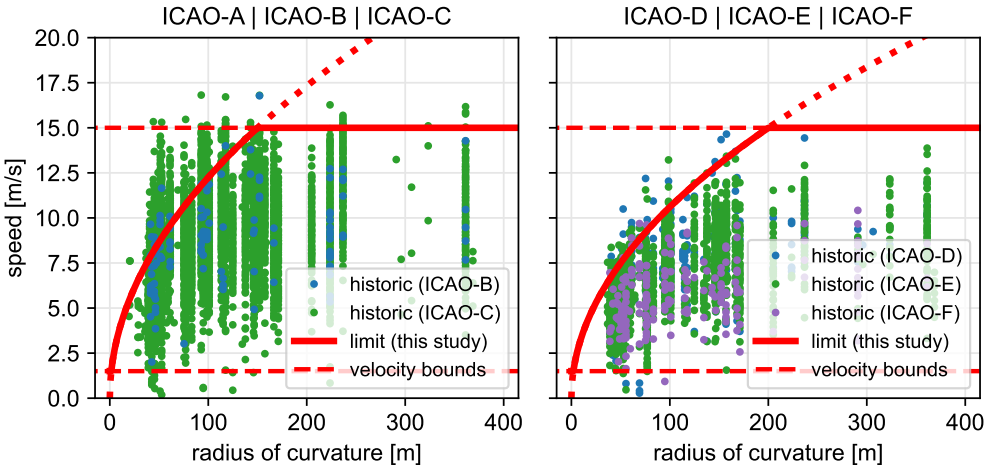


Figure 6.5: Calibration of curve speed with historical track data; the red dotted lines mark the speed limits computed using Equation (6.1)

Table 6.1: Kinematic and algorithm parameters that are used in the routing algorithm

parameter	value	unit
maximum speed $v_{\max}$	15	m/s
minimum speed $v_{\min}$	1.5	m/s
curve speed v_{curve}	per edge	m/s
acceleration acc	0.4	m/s²
deceleration dec	-0.5	m/s²
safety distance in general	1	averaged shape radius
safety distance trailing	3	shape radii of preceding aircraft
planning window w_{p}	15 to 60	min
replanning period h_{p}	3 to 48	min (= 20 % to 80 % of w_{p})

6.1.5 Verification and validation

Verification and validation of the simulation model were performed in accordance with validation techniques and tests as described by Sargent [151]. To validate the conceptual model, operational experts from Amsterdam Airport Schiphol were consulted. During implementation, continuous verification was performed. The model was developed in different modules, allowing for the independent testing of the building blocks. In addition, assertion conditions were added to ensure correctness of the internal processes in the code and compiler errors were resolved. With visual animations, we verified that the routes were executed as planned. Furthermore, small test scenarios were created to verify the model's behaviour in the bay areas for pushback, push-pull, and engine start manoeuvres. The activity-based path planning was verified with small test scenarios ensuring correct timings and kinematics. With these scenarios, face validation was performed to ensure that the model performance was as expected. Finally, individual agent behaviour was carefully followed throughout the system to ensure correctness. As safety must not be compromised, we confirmed that indeed no collisions between agents occurred during the execution of the planned conflict-free routes.

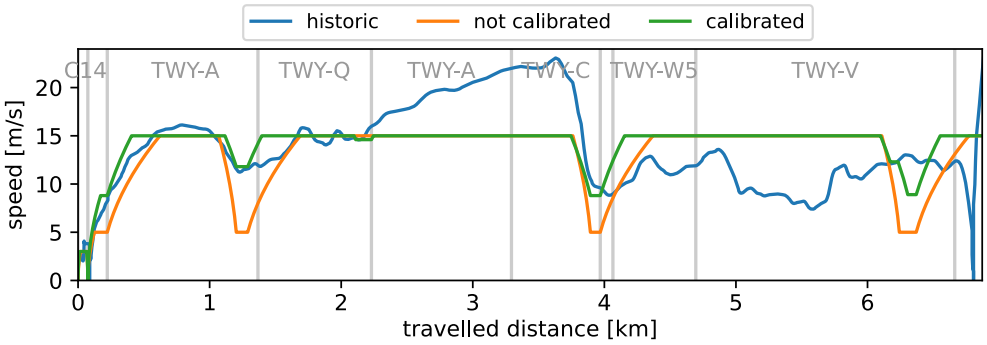


Figure 6.6: Exemplary historical, un-calibrated, and calibrated speed profiles of a departing flight travelling from stand C14 to runway 36L

As validation of the calibrated agent kinematics, Figure 6.6 shows a representative historical speed profile over travelled distance (black line) as well as simulated single-agent trajectories along the same path for both the un-calibrated agent kinematics used in [152] (green line) and the calibrated ones of this work (orange line). Especially on the long straight taxiway segments towards runway 36L, the historical trajectories indicate that aircraft often taxi faster than the speed limit of 15 m/s.

6.2 Experimental Setup

In this section, we present the experimental setup to simulate ASM Ops using the historical flight schedules for 17–18 July 2019, which were two of the busiest days to date at Amsterdam Airport Schiphol. To this end, we outline additional assumptions for our study and give an overview of the traffic situation on these two days. Furthermore, we describe the analyses and key performance indicators to evaluate the study.

6.2.1 Study assumptions

We draw the following additional assumptions with respect to the concept of the control architecture outlined above:

- the final flight schedule of the two operational days is used and remains static throughout the simulation;
- arriving aircraft are spawned at their historical Actual Landing Time (ALDT); their speed at rapid-exit taxiways $v_{\text{RET}} = v_{\max}$, or at regular exits $v_{\text{exit}} = v_{\text{curve}}$;
- departing aircraft are spawned at their historical Actual Off-Block Time (AOBT), but are allowed to hold at the stand; they use the standard pushback path of the stand according to the airport manuals (see [153]);
- the routing algorithm can freely determine the takeoff sequence, but has to comply with the CTOT-slots of the historical data;
- all vehicles execute the instructions from the Guidance Agents perfectly, i.e. no deviations to planned routes;
- the simulation is executed sequentially, i.e. paused when routes are planned;
- the routing algorithm must always return a solution for the simulation to continue, i.e. we do not consider any degradation modes for path planning;
- the standard taxiway flow directions at Schiphol are ignored.

These assumptions decrease the complexity within the simulation, and give the routing algorithm more freedom to optimize, limiting the risk of an incomplete solution, i.e. situations in which the routing cannot be done. Moreover, they exclude any uncertainty or noise, making the simulations deterministic. As outlined below, the simulated scenarios are based on the most common runway configurations, and we use flight schedules from two of the busiest days at Schiphol to date. Despite the assumptions made, the simulated MAS model represents the airport surface movement operations in detail. We thus deem the conducted simulations suitable to study the key operational consequences when automating ASM Ops. Furthermore, we discuss the main aspects influencing the comparison between historical and simulated operations in Section 6.3.1.

6.2.2 Traffic situation

At Schiphol, two main runway modes of operation (RMO) exist: **RMO North** (active on 17 July 2019), and **RMO South** (active on 18 July 2019). During each day, different runway combinations are set. These RMO phases are visualised in Figure 6.7.

In the simulation, the same runways are activated according to those that were active in the historical operations. The runway 09/27 was not active during the two days. Aircraft landing on or departing from runway 04/22 are not modelled as these general aviation flights remain foremost within Schiphol East. Table 6.2a lists the total number of flights, arrivals, departures, the main RMO, and the number of RMO phases for the two simulated days. As mentioned above, each aircraft is categorised as one of the six ICAO-types with an associated shape radius and wake turbulence category (WTC). Table 6.2b lists these parameters along with the count over the two days.

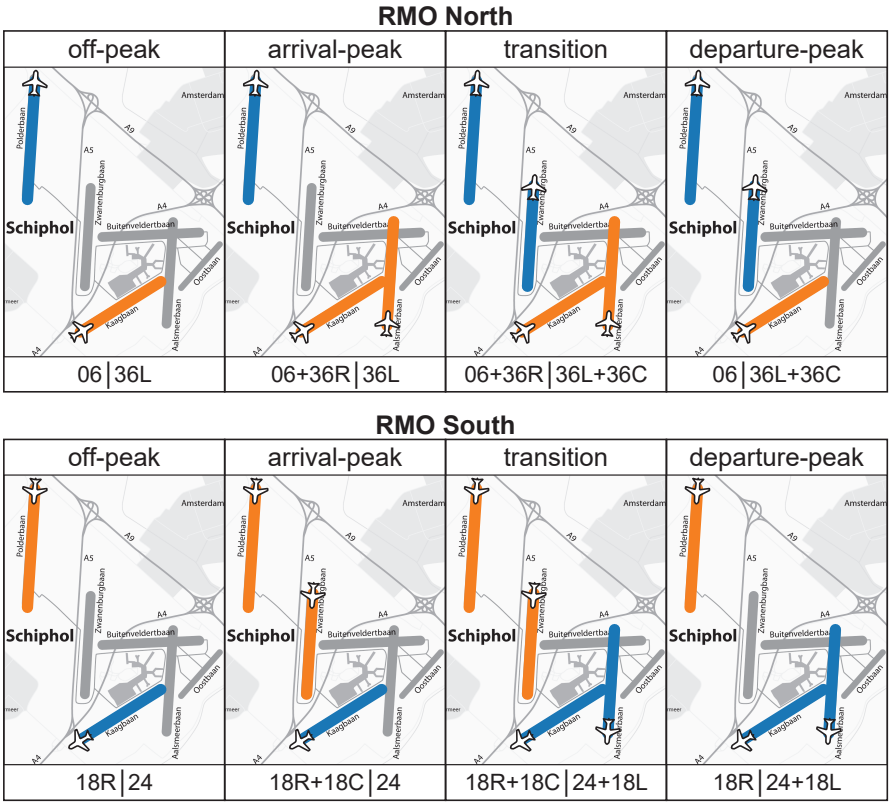


Figure 6.7: Runway modes of operation (RMO) at Schiphol: different phases of RMO North (top) and RMO South (bottom) with active runways for arrivals (orange) and departures (blue); background map adapted from [154]

Table 6.2: Overview of (A) traffic data and (B) parameters and daily counts of ICAO-types

(a) traffic data from July 2019

date	17 July	18 July
flights	1 489	1 492
arrivals	745	744
departures	744	748
RMO	North	South
RMO phases	19	19

(b) ICAO-types; WTC = wake turbulence category

	parameters		count per day	
	shape	WTC	17 July	18 July
ICAO-A	12 m	CAT-F	0	0
ICAO-B	25 m	CAT-E	22	20
ICAO-C	40 m	CAT-D	1 195	1 198
ICAO-D	54 m	CAT-C	37	43
ICAO-E	72 m	CAT-B	213	206
ICAO-F	80 m	CAT-A	22	25

6.2.3 Overview of analysis and key performance indicators

We perform two types of analyses to explore the operational consequences of fully-automated ASM Ops, as outlined below. All conducted simulations use the historical flight schedules and RMOs defined above. The taxi time distributions form the basis of our analyses as they are directly linked to the efficiency, predictability, and resulting emissions of airport surface movement operations. Since the taxi times are not necessarily normally distributed, we report the averages, medians, and interquartile ranges. Furthermore, we exclude flights with a hold-type assigned by the Routing Agent due to the potential impact on their taxi times.

Analysis of algorithmic parameters to investigate the influence of the planning window w_p and replanning period h_p on the trade-off between:

1. Efficiency of operations: How do the parameters influence the distribution of taxi times? Intuitively, we expect that longer planning windows combined with shorter replanning periods yield lower and less varying taxi times (Hypothesis H1).

2. Predictability of taxi times: In the simulated operations, how predictable is the remaining taxi time when the planning window changes? To this end, we define the predictability of the taxi time as the actual vs. predicted value with respect to the remaining time till the end of taxiing: recall that per planning round, the MAS plans the routes for all aircraft that are scheduled to start within the planning window and those that are already taxiing. Conflicts with other agents that occur beyond w_p are ignored. This yields a predicted taxi time that is updated in the subsequent planning round, which takes place latest after h_p has passed. We hypothesise that longer planning windows combined with more frequent plan updates (lower h_p) yield more accurate predictions of the remaining taxi time (Hypothesis H2).

3. Computational efficiency of routing algorithm: How long does it take to coordinate the agents per planning round with respect to different sets of w_p and h_p? Intuitively, we expect that longer planning windows require more computational time to reach a conflict-free solution as more agents need to be coordinated (Hypothesis H3).

To this end, we simulate the operations with different values of w_p and h_p within the ranges specified in Table 6.1. We analyse the first part with the taxi time distributions, the second with the deviations between predicted and actual taxi times, and the third with the computational time in relation to the number of agents per planning round. As outcome of the trade-off, we choose a set of parameters to conduct the following analysis.

Analysis of operational consequences of fully-automated airport surface movement operations with respect to:

1. Delay hotspots in the simulated operations: In the routing algorithm, three mechanisms exist to let an agent avoid the reservations of other agents: an alternative path is chosen, its speed is reduced down to the minimum speed of 1.5 m/s, or the agent has to hold at the stand or engine-start location. The latter two create additional taxi time that we sum per 5 m-segment of the graph underlying the airport layout. This yields the locations where most delays across all flights occur. Per simulated day of operations, we visually examine these congestion areas in the airport layout.

2. Efficiency gain on each of the two days: We compare the distribution of taxi times between historical and simulated operations, and summarise the efficiency gain as change in the key performance indicators related to the taxi time.

3. Predictability of simulated operations compared to the historical operations: we assess the predictability of the taxi times by defining metrics similar to those used by Liu, Hansen, Gupta *et al.* [144]:

   - The predictability of takeoff times as difference between the Estimated taXi-Out Time (EXOT) and Actual taXi-Out Time (AXOT) of both historical and simulated operations: to this end, we calculate the root-mean-square prediction error $RMSE$ and the inequality coefficient U as defined below.
   - Likewise, the predictability of in-block times as difference between the Estimated taXi-In Time (EXIT) and the Actual taXi-In Time (AXIT) of both historical and simulated operations.

   The prediction error $RMSE$ is defined as:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(AXT_i - EXT_i)^2} \tag{6.2}$$

   with the corresponding actual taxi time (AXT) and estimated taxi time (EXT) per flight i as well as the total number of flights n. The inequality coefficient U quantifies the relative prediction error and is calculated as:

$$U = \frac{\sqrt{\sum_{i=1}^{n}(AXT_i - EXT_i)^2}}{\sqrt{\sum_{i=1}^{n}(AXT_i)^2}} \tag{6.3}$$

   When this ratio is zero, all predictions are perfect, while unity denotes that the RMSE equals the theoretical error of constantly predicting a taxi time of zero.

4. Inbound and outbound holding: We assess how many arriving and departing aircraft must hold, and the effect on the taxi times. To this end, we compare the taxi time distributions between historical and simulated operations of all flights that must hold during taxiing.

5. Compliance with CTOT-slots: We analyse the compliance with CTOT-slots and the distribution of takeoff times within the CTOT window between historical and simulated operations.

6. Runway sequencing and usable capacity: We provide an exemplary takeoff order as comparison of the runway sequence between the historical and simulated operations. As indication of the runway use, we provide the maximal hourly throughput and occupancy rate of any departure or arrival runway. The occupancy rate is calculated as the relative time that the runway is blocked due to the minimum wake turbulence separation between two consecutive aircraft. Furthermore, we calculate the remaining takeoff slots per day that comply with the wake turbulence separation of the flights before and after. We compare the historical to the simulated slot count to estimate the surplus runway capacity.

6.3 Experimental Results

To give an overview of the traffic situation, Figure 6.8 shows the hourly count of all flights for both the historical (black dotted line) as well as simulated operations (grey line) over the 17–18 July 2019. The two curves almost match each other, with the simulated operations showing a slightly lower total count due to the lower taxi times as discussed below. Furthermore, the chart visualises the count of arriving vs. departing flights: the alternating trend between landings and takeoffs that is characteristic for a hub-and-spoke airport such as Schiphol is clearly visible. This is also reflected in the frequently changing RMO phases over the course of the two days, illustrated by the colored shades in the figure.

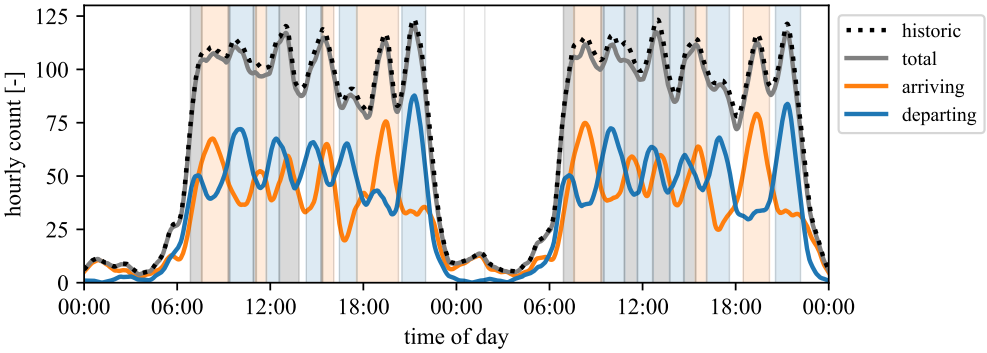


Figure 6.8: Hourly count of flights over the two days; shades denote the RMO phase: off-peak (white), arrival-peak (orange), transition (grey), and departure-peak (blue)

Table 6.3: Comparison of historical and simulated operations with $w_p = 20$ min, $h_p = 10$ min

	date	17-07-2019		18-07-2019	
	operations	historical	simulated	historical	simulated
ARR	mean taxi time	04:38	03:03	10:15	07:40
	median taxi time	04:00	02:53	10:44	08:37
	IQR taxi time	02:51	01:40	05:59	05:13
	RMSE taxi time prediction	01:47	00:02	02:16	00:02
	U taxi time prediction	33.9 %	1.1 %	20.6 %	0.5 %
	RWY throughput*	42		38	
	RWY occupancy*	68.7 %		69.3 %	
DEP	mean taxi time	14:29	11:18	10:20	07:05
	median taxi time	14:32	11:20	10:06	06:34
	IQR taxi time	06:32	04:56	04:20	02:29
	RMSE taxi time prediction	02:45	00:40	02:44	00:35
	U taxi time prediction	18.1 %	5.5 %	25.2 %	7.9 %
	RWY throughput*	45	45	43	42
	RWY occupancy*	74.7 %	76.1 %	73.5 %	77.0 %
	CTOT-slot violations	5	3	0**	0

*: maximal hourly value for any runway
**: corrected after checking historical A-CDM milestones

Table 6.3 lists the mean, median, and interquartile range (IQR) of the taxi times for inbound and outbound flights. The underlying distributions exclude flights that hold explicitly during taxiing, i.e. arriving flights without a free gate and departing flights with an issued CTOT-slot exceeding the required taxi time. The distributions are further discussed in Section 6.3.2, while the excluded flights are analysed separately in Section 6.3.6. As predictability metrics, we list the prediction error $RMSE$ and inequality coefficient U, and refer back to these in Section 6.3.3. Moreover, for any runway in use, we report the maximal throughput and occupancy rate per hour. For arrivals, the indicators are identical between the historical and simulated operations as we used the actual landing time (ALDT) as spawn-time in the simulation (listed as assumption in Section 6.2). For departures, the maximal hourly throughput is similar between historical and simulated operations, while the maximal hourly occupancy rate has increased for the simulated operations. Section 6.3.8 provides more details on the runway sequence and capacity. The listed CTOT-slot violations are further discussed in Section 6.3.7.

6.3.1 Influencing factors on taxi time comparison

As visualised in Figure 6.9, most A-CDM milestones of landings and takeoffs, i.e. Actual Landing Time (ALDT) and Actual Take-Off Time (ATOT), are located on the runway and are thus viewed as sufficiently precise. Likewise, most block-times, i.e. Actual In-Block Time (AIBT) of arriving and Actual Off-Block Time (AOBT) of departing flights, are fairly accurate. However, the tracks of some flights end (1.6 %) or start (4.9 %) further away from the respective stand. Especially for departing flights, an AOBT situated outside the bay area may lead to significantly lower taxi times: the pushback and engine-start procedures require multiple minutes, but are likely not captured in the resulting historical taxi time.

Moreover, the historical tracks of departing flights often do not follow the standard pushback paths that the simulated flights must follow. Additionally, the actual engine-start duration cannot be extracted from the historical tracks. In the simulation, aircraft are only allowed to enter the runway when they can immediately take off, i.e. use a rolling takeoff. However, we do not model the takeoff, and instead use the time point of entering the runway as ATOT. While all of these factors introduce potential offsets between the historical and simulated taxi times, we believe that those leading to shorter historical times have a greater impact. Therefore, we did not attempt to correct the historical A-CDM milestones, but use them as such when comparing the taxi times in the following.

6.3.2 Comparison of taxi time distributions per runway

Figure 6.10 displays the distributions of taxi times for each arrival and departure runway for the historical and different simulated operations as box-and-whisker plots. These represent the median, first and third quartiles as box, while outliers are marked by whiskers and points. Like Table 6.3, all distributions exclude flights that hold explicitly during taxiing, which are analysed separately in Section 6.3.6.

In general, the taxi times from the simulated operations are shorter and vary less. Since the runway 18R/36L is far away from the central part of Schiphol, taxiing to/from this runway takes more time than to any of the other runways. As the departing aircraft have to start their engines to taxi after pushback, their taxi time to any runway is in general longer than for aircraft that land on the same runway. Furthermore, since we used an

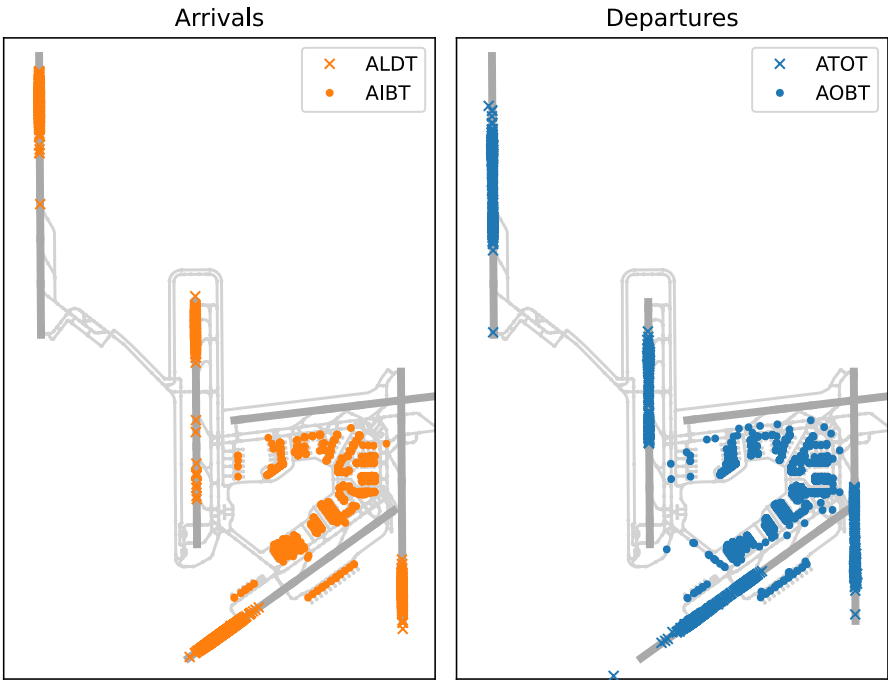


Figure 6.9: Aircraft positions at historical A-CDM milestones for inbound (left) and outbound flights (right)

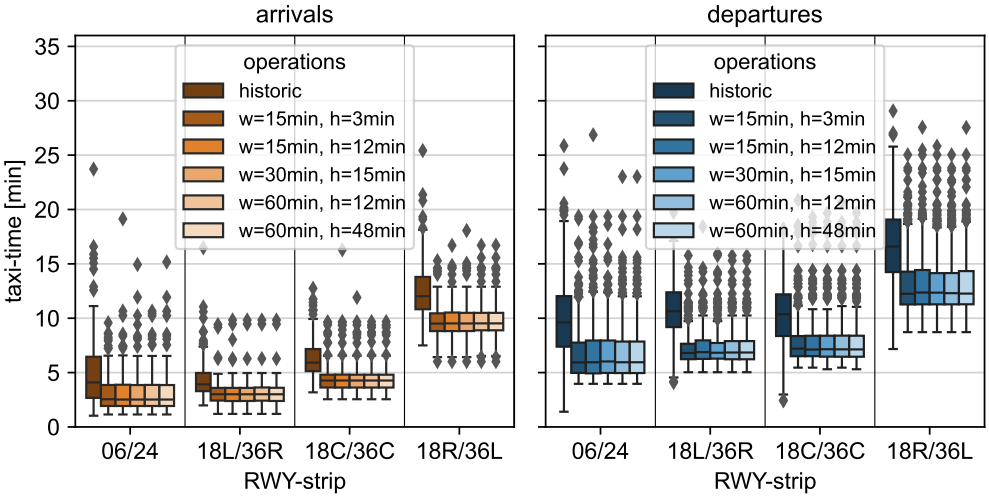


Figure 6.10: Box-and-whisker plot of historical and simulated taxi times for arrivals and departures per runway-strip

engine-start time of 6 min for large aircraft (ICAO-D to ICAO-F) in comparison to 3 min for small aircraft, the taxi times of departing aircraft vary more than those of arriving aircraft in the simulated operations.

Of the simulations carried out for the algorithmic analysis, the five shown in Figure 6.10 represent different combinations for w_p and h_p. We also tested values for w_p below 15 min, but the routing algorithm did not succeed to find a solution throughout each of the two days. The taxi times do not differ significantly between the simulations, both for arriving and departing flights from all runways. Therefore, we conclude that both the planning window w_p as well as the replanning period h_p do not impact the efficiency of operations within the ranges that we tested, and we reject Hypothesis H1.

6.3.3 Predictability of taxi times

Figure 6.11 shows the variability of the taxi time predictions with respect to the remaining actual taxi time for four sets of w_p and h_p. The red lines mark the 1 % and 99 % quantiles as indication of the accuracy over the remaining time. For all four simulations, the first predictions underestimate the actual taxi time: the aircraft start taxiing almost at the end of the planning window and most conflicts are thus not yet resolved. The deviation to the actual taxi time decreases strongly in the following planning rounds. When the remaining taxi time is less than w_p, the difference between predicted and actual taxi time is negligible for more than 50 % of all flights. The accuracy further increases towards the end of taxiing. Longer planning windows yield accurate predictions within a longer duration till the end of taxiing, supporting Hypothesis H2, while h_p has a subordinate effect on the predictability. As listed in Table 6.3, the $RMSE$ and U values decrease significantly in comparison to the historical operations. Note that this may change when deviations to the planning arise during execution, which we did not model in this work.

6.3.4 Computational efficiency of routing algorithm

The multi-agent system including the routing algorithm are implemented in Python. We ran the simulations on a Windows 10 laptop equipped with a 1.80 GHz Intel Core i7-10610U CPU and 16 GB RAM. In Figure 6.12, the runtime of each planning round is plotted over the number of agents that had to be routed for different sets of w_p and h_p. The planning rounds on day 1 (RMO North) are marked by circles, and those of day 2 (RMO South) by crosses. The algorithm scales with approx. quadratic time complexity in relation to the number of agents (supporting Hypothesis H3), while the RMO has a subordinate effect. From the parameter settings we tested, we deem $w_\mathrm{p} = 20\,\mathrm{min}$ with $h_\mathrm{p} = 10\,\mathrm{min}$ to yield a good trade-off between a fast algorithmic runtime and accurately predicting the remaining time within the last 10 min of taxiing. Thus, we chose these parameters to analyse the operational consequences of the fully-automated operations.

6.3.5 Delay hotspots in the taxiway network

Figure 6.13 visualises congested areas in the taxiway network on each of the two days. On both days, hotspots form in front of the runway stopbars, the bay areas and stands, as well as on some taxiway segments. On 17 July, one of the latter hotspots is located in front of the crossing of the second departure-runway 36C, which is active only in departure-peaks and transition-phases (see Figure 6.7). Aircraft slow down in front of the crossing to await

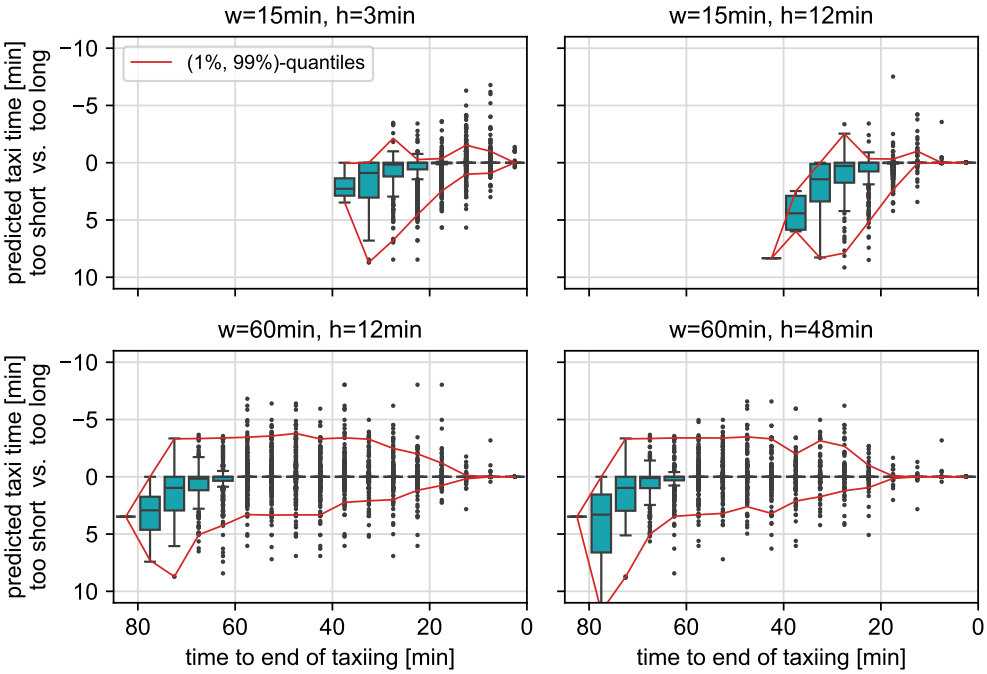


Figure 6.11: Predictability of taxi times: in each planning round, only the conflicts within the planning window w_p are resolved; the estimated taxi times are then updated in the subsequent planning rounds; based on the remaining taxi time, the predictions as difference between estimated and actual taxi times are visualised as box-and-whisker plots, and red lines marking the 1 % and 99 % quantiles of the underlying distributions

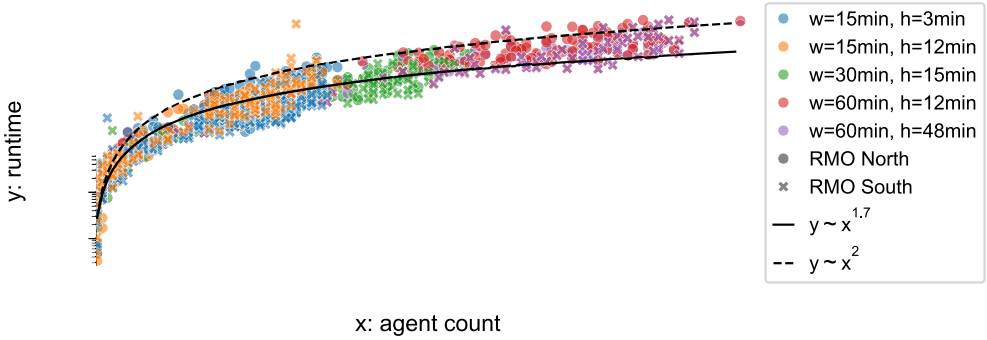


Figure 6.12: Runtimes of planning rounds over the number of agents; the solid and dashed black lines indicate polynomial time complexities of $\mathcal{O}(x^{1.7})$ and $\mathcal{O}(x^2)$, respectively

its opening, which the Routing Agent determined to be faster than letting the aircraft go around the Southern end of 18C/36C. In front of the runways, aircraft queue with reduced speed until the minimum separation time due to wake turbulence of the preceding aircraft has passed. In the bay areas, aircraft mostly hold at their stand, with some more holding after engine-start occurring on the second day.

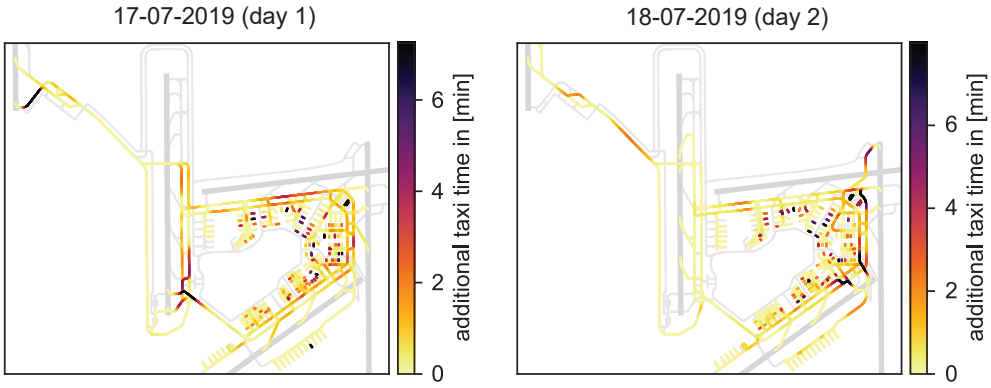


Figure 6.13: Additional taxi time per layout location on each of the two days, simulated with $w_p = 20\,\text{min}$, $h_p = 10\,\text{min}$

6.3.6 Holding of inbound and outbound flights

In the taxi time analysis in Section 6.3.2, we excluded flights with a hold-type assigned by the Routing Agent. Figure 6.14 compares the taxi times of these flights between historical operations and the different hold-types of the simulated operations as box-and-whisker plot. In general, not many flights are holding. In comparison to Figure 6.10, the taxi times of flights with inbound holding are significantly higher, and are similar between historical and simulated operations, also considering that some of the historical A-CDM milestones end at the holding locations. For most outbound aircraft that must be delayed due to their CTOT-slots, the Routing Agent lets them hold at their stand. The taxi times of the historical operations are slightly longer, which has an influence on the moment that the aircraft take off within the CTOT-slot, as we analyse further below.

6.3.7 CTOT-slots of outbound flights

Over the two days, a total number of 442 CTOT-slots are assigned by Eurocontrol. Figure 6.15 visualises the compliance between the takeoff times to the CTOT-slots of both historical and simulated operations. While the historical times almost follow a normal distribution centred around the time issued by Eurocontrol, those of the simulated operations are skewed towards the beginning of the CTOT-window. In its current implementation, the routing algorithm optimises for lowest taxi times and does not attempt to let aircraft take off closer to their calculated takeoff time. From the 442 flights, 8 historical flights (1.8 %) do not comply with their CTOT-slots. In comparison, only a single simulated flight takes off outside its CTOT-slot. However, we noticed that for this flight, the historical A-CDM milestone occurs multiple minutes after the actual pushback, rendering it a faulty outlier.

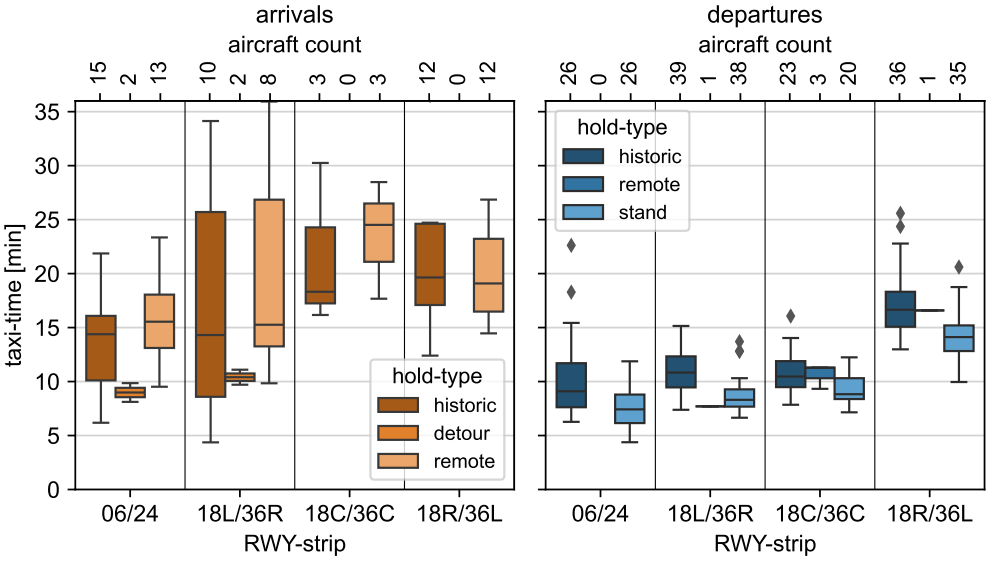


Figure 6.14: Box-and-whisker plot of historical and simulated taxi times dependent on the hold-type of arrivals and departures per runway-strip; simulation with $w_p = 20$ min and $h_p = 10$ min. The hold-type 'remote' denotes holding at one of the remote holding locations, 'detour' marks trajectories that go via one of these holding points. The hold-type 'stand' denotes explicit holding time at the stand, in contrast to implicit holding at the stand as emergent property to resolve conflicts during path planning

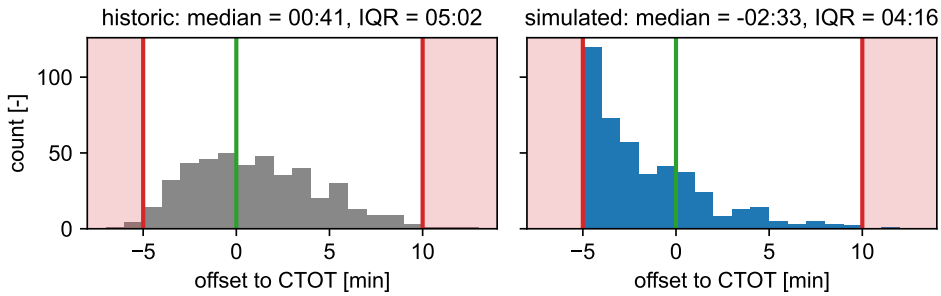


Figure 6.15: Compliance to CTOT-slot (white area) of historical (left) and simulated operations (right)

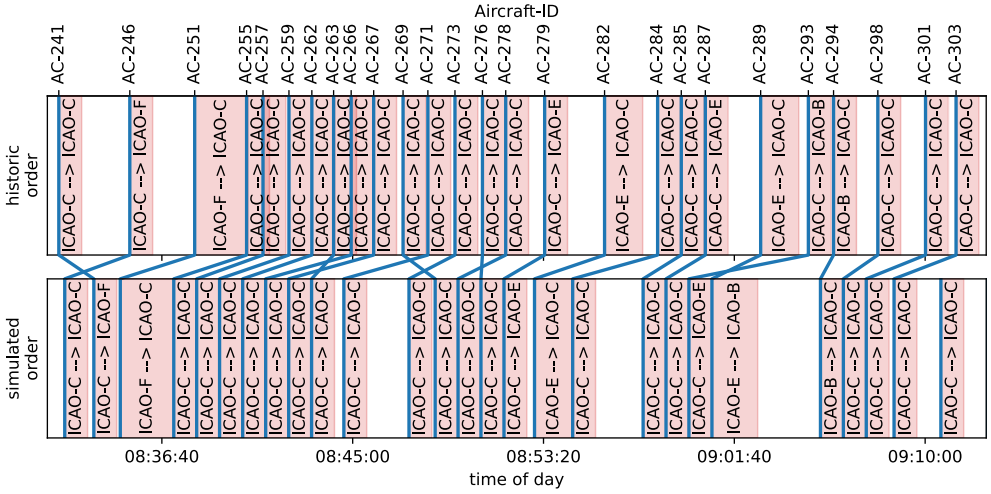


Figure 6.16: Comparison of exemplary takeoff order at runway 36L between historical (top) and simulated operations (bottom), with actual takeoff time (blue lines) and minimum WTC separation (red shades)

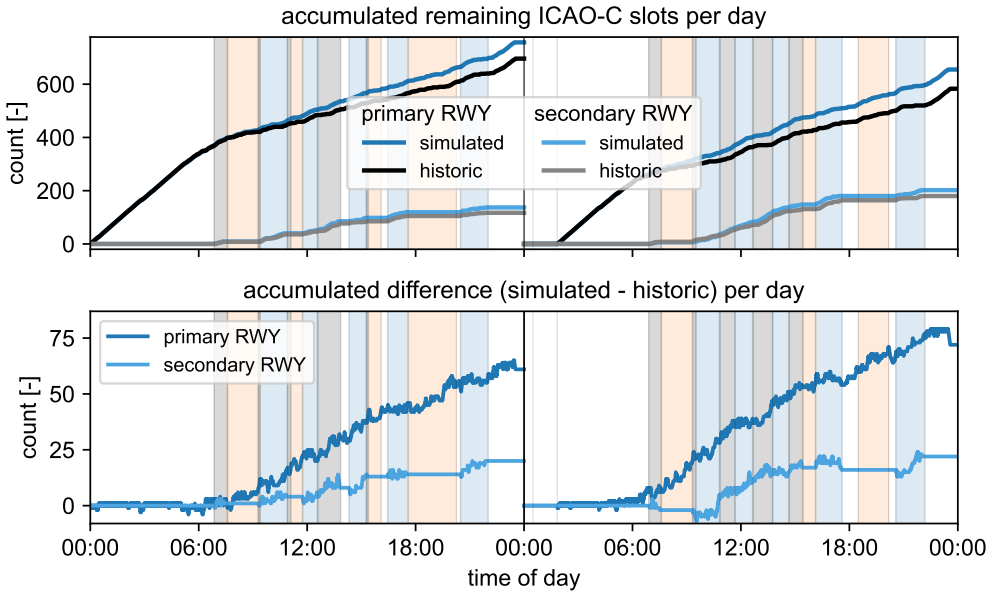


Figure 6.17: Remaining takeoff slots for ICAO-C aircraft per day for both historical and simulated operations; shades denote the RMO phase: off-peak (white), arrival-peak (orange), transition (grey), and departure-peak (blue)

6.3.8 Runway sequence and capacity of outbound flights

As defined in Section 6.2, we do not use a pre-defined takeoff sequence. Thus, the order of aircraft departing from a runway is different between the historical and simulated operations, as exemplary shown in Figure 6.16. The minimum separation between flights according to RECAT-EU is illustrated by the red shades. While the historical order mostly adheres to the RECAT-EU separation, in some cases, two aircraft are separated less than the minimum. We confirmed with the track data that indeed some flights take off around 52 s after the previous one, despite having a larger separation than indicated in the figure due to inaccuracies of the A-CDM milestones.

As emergent property of the MAS, whenever possible, flights are grouped together with minimum separation time between each takeoff. Based on this observation, we count the remaining slots per departure runway for each of the two days, as visualised in Figure 6.17. For both historical and simulated operations, takeoff slots remain even during busy departure peaks marked by the blue shades. However, throughout the two days, more slots remain for the simulated operations: in relation to the total number of departing flights, 10.9 % and 12.6 % additional slots are available on each of the two days, respectively. Thus, the MAS better utilises the potential runway capacity.

6.4 Future Work

In this chapter, we considered a futuristic scenario of fully-automated operations. However, before real-world airport surface movement operations are fully-automated, a long transition period will be necessary with many challenges to investigate and resolve. In general, it needs to be explored which tasks could be automated and how controllers could then interact with the automated support tools. Moreover, a suitable architecture of the human–autonomy teaming that keeps controllers in the loop is needed. The underlying model should also handle automation failures in an effective way, e.g. by defining degradation modes for path planning in case the routing algorithm cannot find a conflict-free solution. We will focus on creating a suitable framework and evaluating it in human-in-the-loop simulations in future work.

In the presented study, the Aircraft Agents were assumed to execute all commands exactly as instructed. However, many uncertainties arise in real-world operations. Future research should determine how severe different sources of uncertainty are, and how to include these in the model. Moreover, it would be valuable to compare the centralised optimisation of the concurrent routes presented in this chapter to a baseline-model that represents the current working procedures of ground controllers more closely. As such, we plan to implement a distributed control model with local adjustments in future work.

Furthermore, as part of surface movement operations, various types of ground vehicles may come in close contact to parked and moving aircraft, foremost in the aprons. Their movements must be coordinated with each other and the aircraft. Future versions of the model should include such operations to explore the operational consequences that may result from their automation.

6.5 Conclusion

In this chapter, we presented a hierarchical multi-agent system model for fully-automated taxiing operations at large airports in which different agents coordinate and control all movements on the airport surface. As centralised agents, the Airport Operations Agent handles the flight schedule and runway configuration, while the Routing Agent computes conflict-free trajectories for all Aircraft Agents. Their execution is then instructed and monitored by the Guidance Agents. We accounted for aircraft shapes and kinematics during path planning on a high-resolution airport layout. To this end, we calibrated the model using historical ADS-B data. Furthermore, important airport surface movement elements and processes were explicitly included in the model such as pushback, engine-start, inbound holding, complying with CTOT-slots, and adhering to a minimum safety distance during taxiing as well as minimum wake turbulence separation during takeoff. The routing algorithm was evaluated to be well suited for planning conflict-free trajectories of all aircraft.

We analysed the proposed model using the real-world schedules of two of the busiest days at Amsterdam Airport Schiphol, including 19 different runway configurations per day. For the considered simulation conditions, the fully-automated operations controlled by the MAS model reduced the average taxi time per flight by around 3 min, or 30 %. Furthermore, the simulated operations decreased the taxi time variability and yielded a high accuracy in predicting the remaining time till the end of taxiing. Counting the remaining takeoff slots at departure runways yielded an increase of approximately 11 % in comparison to the historical operations, meaning that the MAS model better utilises the available runway capacity. All obtained results demonstrate the potential of the MAS as control model for fully-automated airport surface movement operations that are more efficient, predictable, and hence produce less emissions.

> The multi-engine taxiing case study presented in this chapter addresses RQ 4. Building on the calibrated baseline established here, the next chapter turns to emission reduction by investigating tug-enabled taxiing for all outbound flights under the same far-term automation assumptions (RQ 5), and assesses its system-level implications for highly-automated, lower-emission ASM Ops. Subsequently, based on feedback from operational experts, Chapter 8 revisits the model calibration of the movement parameters introduced in this chapter to better align the MAS model with the historical operations.

7

Engine-Off Taxiing of Outbound Flights

Airports worldwide face a dual challenge towards 2050: meeting the predicted annual demand of 10 billion passengers, while striving for net-zero emissions throughout their operations. For airport surface movement operations, engine-off taxiing methods have the potential to cut the associated aircraft emissions. One promising technique is to use specialised tugs to tow aircraft to or from locations close to the runways. However, the implementation of tug-enabled taxiing (TET) introduces operational complexities, necessitating advanced planning, guidance, and control.

In this chapter, the multi-agent system model for automated taxiing operations is further extended to explore the operational implications posed by TET of all outbound flights. To this end, simulated TET operations are compared to the historical as well as automated multi-engine taxiing operations reported in Chapter 6. Under the considered simulation conditions, TET substantially reduces emissions in bay areas and decreases fuel consumption by up to 69 %, corresponding to estimated annual fuel-cost savings of € 13.8 million at Schiphol. These results indicate that TET operations can provide both environmental and economic benefits, while potentially lowering health-related risks for ground personnel.

This chapter is based on the following publication:

M. von der Burg and A. Sharpanskykh, 'Studying the operational consequences of automated engine-off taxiing using multi-agent planning: Explorative case study at Amsterdam Airport Schiphol', in *Proceedings of ICRAT-24*, Singapore, 2024, pp. 1–8.

Airports around the world are facing two conflicting challenges on the path to 2050: they must cope with the predicted demand of more than 10 billion yearly passengers, and must reduce their environmental footprint to fulfil the industry-wide goal of reaching net-zero emissions by the same time [1]. Achieving these opposed goals is even more challenging for large airports as it is expected that these cannot facilitate this growth with infrastructural expansions [8].

Emissions during taxiing can effectively be reduced when aircraft do not use their engines during taxiing. The TaxiBot concept from IAI [12] is one of the most promising of such engine-off taxiing techniques [15]. A tug replaces the pushback tractor, and after pushback, continues to tow departing aircraft to a decoupling location close to the runway where the aircraft is decoupled from the tug. Since the tugs provide sufficient towing power, the aircraft engines can remain off, until the pilots have to start the engines to be ready to taxi onto the runway after decoupling.

In a feasibility study, Schiphol explored the potential consequences and open challenges to implement such tug-enabled taxiing (TET) for outbound flights throughout their operations [155], [156]. The study was based on the current infrastructure and processes, and highlighted the necessity for an advanced guidance and control system to counteract challenges such as queues forming on active taxiways due to tug decoupling.

Different aspects of airport surface movement operations (ASM Ops) were previously studied, including multi-objective routing [142], [143], automation [6], [141], and review of optimisation approaches [71]. In Chapter 6, we examined the operational consequences of automated ASM Ops, controlled by a multi-agent system with integrated routing algorithm to plan conflict-free trajectories and execute them accordingly. As suggested by [146], we included various processes such as pushback and engine-start explicitly in our model. Simulating the operations of two consecutive days at Amsterdam Airport Schiphol in July 2019 (two of the busiest days to date) resulted in 15 % reduction of taxi times. However, to assess whether this operational control system is also suitable for TET operations, multiple questions remain, some of which are:

- How do TET operations affect key performance indicators like the taxi times of inbound and outbound flights as well as runway capacity?
- Does the long tug decoupling time of around two minutes lead to congestion and delays, especially when in-flow decoupling is performed on busy taxiways?
- Do TET operations offer the potential to reduce fine particle emissions in the bay areas that pose a health risk for ground personnel [157]?

In this chapter, we explore the potential operational impact that engine-off taxiing of outbound flights through the use of tugs poses. To this end, we extend the multi-agent system model from our previous work to simulate tug-enabled taxiing operations for the same flight schedule used in that study. The adapted model is described in Section 7.1, and Section 7.2 explains the experimental setup. In Section 7.3, based on key performance indicators, we then compare tug-enabled taxiing to both historical and simulated multi-engine taxiing (MET) operations. Relating back to the open questions described above, we summarise our findings in Section 7.5.

7.1 Multi-Agent System Model

The architecture of the multi-agent system (MAS) comprises both centralised and distributed agents that are organised in a hierarchical-distributed structure. Since we consider automated operations in this chapter, we assume that the agents have full control and decision-making over the taxiing operations. Moreover, it is assumed that communication is done digitally through a datalink like AeroMACS, and that the A-SMGCS surveillance service is fully functional. The MAS is depicted in Figure 7.1 and summarised in the following. Further details on the MAS, including its verification and validation, are provided in Chapter 6.

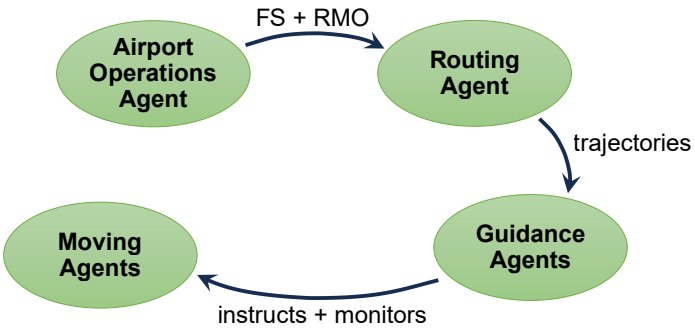


Figure 7.1: Overview of multi-agent system for automated multi-engine and tug-enabled taxiing operations

As shown in Figure 7.2, we represent the taxiway layout of Amsterdam Airport Schiphol, which is used as case study in this chapter, as graph G = (V, E) with vertices V and directional edges E. Vertices denote stands (green), intersections of taxiways (black) and service roads (blue), tug decoupling points (orange), tug all-clear points (grey, explained below), or stopbars in front of runway entries (red). The vertices are connected by two unidirectional edges to form a bidirectional segment of a taxiway (black) or service road (blue).

The centralised Airport Operations Agent determines the active runways based on the runway mode of operation (RMO). Since these must not be crossed by any vehicle, the respective taxiway segments are blocked by layout constraints. Furthermore, it sets the flight and tug schedules (FS), and revises them whenever an A-CDM milestone is updated.

The schedules and constraints are shared with the centralised Routing Agent that calculates conflict-free trajectories for all Moving Agents. A two-level motion planning search is deployed based on Priority-Based Search (PBS) [61] combined with an extended version of the Safe Interval Path Planning (SIPP) algorithm [60]. To ensure conflict-free paths, the algorithm accounts for the kinematics and shapes of agents, and resolves any conflicts within the planning window w_p. The Routing Agent re-computes all plans once every replanning period h_p, or after receiving updates from the Airport Operations Agent. Further details on the routing algorithm are provided in Chapter 4.

The conflict-free routes are shared with the Guidance Agents that are located at every intersection in the taxiway network and control those Moving Agents that are travelling towards their location. They send instructions to the Moving Agents under their control so that these are able to execute the route as planned. When the executed movements diverge from the planned trajectories, the Guidance Agents modify the instructed route to minimise

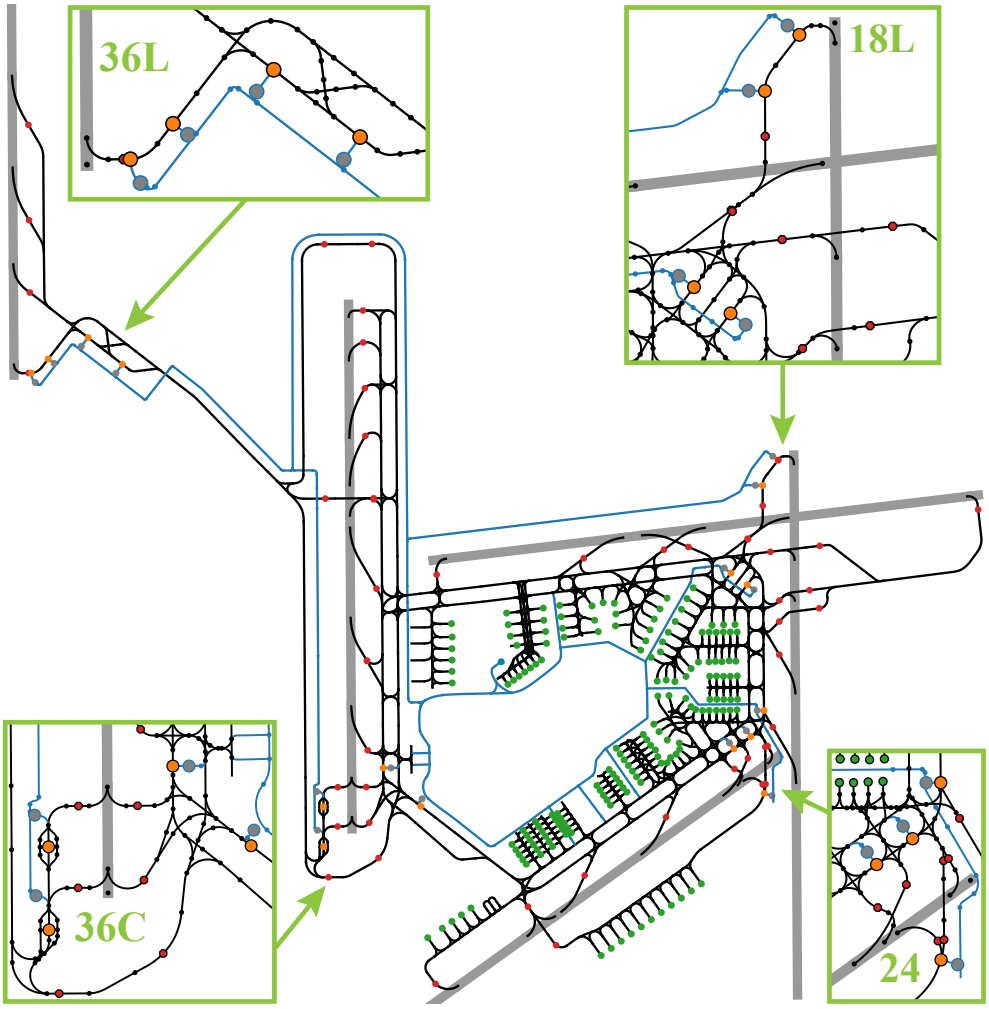


Figure 7.2: Graph of layout of Amsterdam Airport Schiphol with edges for runways (grey), taxiways (black), or service roads (blue), and vertices for gates (green), taxiway intersections (black), tug decoupling (orange) and all-clear points (grey), or stopbars (red)

the deviations. When the executed route differs much from the planned one, they request the Routing Agent to centrally replan the affected routes.

Moving Agents represent the (auto-)pilots and drivers of aircraft and tugs. They are assumed to be fully cooperative, i.e. execute instructions as closely as possible. To account for the various surface movement operations in path planning, the path of a Moving Agent is split into multiple activities based on `GoTo`, `Follow`, and `Wait` activities, see Definition 1 in Section 3.1 for further details.

For each Moving Agent, the Routing Agent uses a combination of these activities to define an activity sequence. Figure 7.3 illustrates these sequences for arriving aircraft using MET, and departing aircraft using either MET or tug-enabled taxiing. To finish the decoupling process as part of TET, a tug drives to a near all-clear point, and gives a signal that the pilot may continue taxiing. The tug then either drives back to its base or to the location of its next assignment.

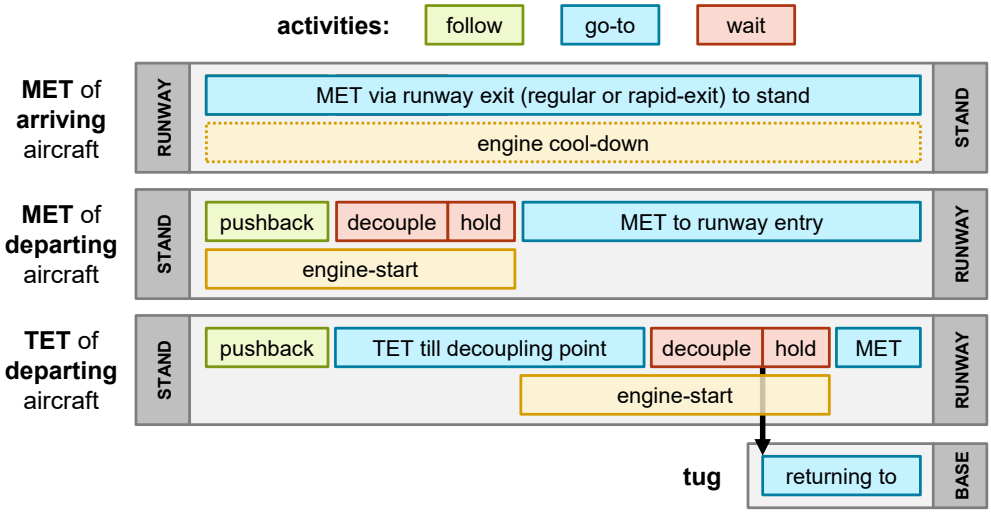


Figure 7.3: Activity sequence for arriving and departing aircraft; while engine-start (orange box) of departing aircraft is accounted for, engine cool-down (dotted orange box) is neglected

The engine warm-up and engine cool-down are special cases in the activity sequence of aircraft. During routing, the warm-up phase is accounted for as part of the engine-start manoeuvre, which is based on the aircraft-specific duration of its engine-start. In case that it takes longer to start the engine than the time till decoupling from the tug or pushback tractor, the trajectory includes additional holding either at the stand, the place of switching direction during pushback, or the decoupling location. Since the engine cool-down of arriving aircraft does not have an influence on the routing, we neglect it and assume that it takes place after standstill at the stand.

In path planning, we represent the Moving Agents as circular shapes with a fixed radius, as listed in Table 7.2 in Section 7.2. To limit the variety, we categorise all aircraft according to the six types listed in the ICAO aerodrome reference codes [69]. Tugs are assigned a fixed shape based on the TaxiBots as reference tug-system [12]. Moreover, we include a safety zone between agents, and account for minimum separation between takeoffs according to RECAT-EU [133].

7.2 Experimental Setup

In this section, we outline the experimental setup to compare the two taxiing modes of operation at Amsterdam Airport Schiphol based on the historical flight schedules for 17–18 July 2019. All experimental assumptions listed in Section 6.2.1 are valid for this study as well. Specifically for TET operations, the following additional assumptions are made:

- all departing flights use TET, while arriving flights use MET;
- tugs are fully electric, no fuel is required;
- sufficient tugs are available, no task assignment is done;
- tugs have to give way to all aircraft, i.e. path planning is done separately for tugs;
- after decoupling, tugs can stay as long as necessary at all-clear points and are routed back to the tug base.

Two primary runway modes of operation (RMO) exist at Schiphol: **RMO North** (active on 17 July 2019), and **RMO South** (active on 18 July 2019). Throughout each day, different runway combinations are set, as described and illustrated in Section 6.2.2. For the simulated operations, the same runways are active as during the historical operations. Runway 09/27 remained inactive during the two days. Flights to and from runway 04/22 are excluded, as these general aviation flights are primarily confined to Schiphol East. For both days, Table 7.1 provides details on the total number of flights, arrivals, departures, the main RMO, and the number of RMO phases.

As discussed earlier, each aircraft is assigned one of six ICAO aircraft types, each associated with a shape-based radius and a wake turbulence category (WTC). These parameters are listed in Table 7.2 along with the count per day. Tugs have a shape radius of 6 m and an unlimited number is available. Table 7.3 lists the kinematic properties of the Moving Agents and key algorithmic parameters. When aircraft are towed by a tug, the maximum speed of tugs is applied. All agents have to keep a minimum safety distance from each other based on the average radius between each pair. Furthermore, when one agent is trailing another vehicle, it must maintain a safety distance of at least three times the shape radius of the preceding vehicle, determined through expert interviews.

To simulate tug-enabled taxiing operations, we made changes to the airport layout corresponding to the schedule data from 2019 to incorporate tug decoupling locations, all-clear points, and service roads for the return movements to their base, derived from interviews with operational experts. Furthermore, we included the second Quebec taxiway that was finished in 2022, which is important to facilitate tug decoupling close to the runway 36C. We thus refer to this layout as 2022+. The underlying graph has 1 255 vertices and 3 468 edges.

In the analysis below, several operational cases are compared. Historical operations and the automated multi-engine taxiing operations from Chapter 6, both using layout 2019, serve as references. The results show that layout 2022+ has a negligible influence on MET operations, which enables TET operations on layout 2022+ to also be compared with historical operations. For TET operations, we use the operational limit of TaxiBots of $v_{\max} = 11.8\,\mathrm{m/s}$. However, to aid our analysis, we also conducted a TET simulation using $v_{\max}$ from MET operations, and refer to this case as 'TET (15m/s)'.

Table 7.1: Overview of operational data

date	17-07-2019	18-07-2019
flights	1 489	1 492
arrivals	745	744
departures	744	748
RMO	RMO North	RMO South
RMO phases	19	19

Table 7.2: Vehicle types: name, shape diameter, wake turbulence category (WTC) for aircraft, and count in operational data

	parameters		count per day	
	shape	WTC	17-07-2019	18-07-2019
ICAO-A	12 m	CAT-F	0	0
ICAO-B	25 m	CAT-E	22	20
ICAO-C	40 m	CAT-D	1 195	1 198
ICAO-D	54 m	CAT-C	37	43
ICAO-E	72 m	CAT-B	213	206
ICAO-F	80 m	CAT-A	22	25
tug	12 m	-	unlimited	

Table 7.3: Kinematic properties of Moving Agents and algorithm parameters that are used in the routing algorithm

parameter	value	unit
maximum speed aircraft $v_{\text{max,AC}}$	15	m/s
maximum speed tugs $v_{\text{max,GV}}$	11.8	m/s
maximum turn speed $v_{\text{turn,max}}$	5	m/s
minimum speed v_{min}	1.5	m/s
acceleration acc	0.25	m/s^2
deceleration dec	-0.75	m/s^2
planning window w_{p}	30	min
replanning period h_{p}	15	min

To provide a rough estimate of the total fuel consumption, fuel costs, and CO_2 emissions, we use a fuel flow rate of 0.1 kg/s per engine based on the value for an A320 from [158]. We assume that all smaller aircraft (ICAO-A, ICAO-B, and ICAO-C) use two engines, while the larger aircraft use four engines. During the engine-start duration of 3 min for smaller and 6 min for larger aircraft, we assume that half the fuel flow rate is necessary. To convert the accumulated fuel consumption into CO_2 emissions and fuel cost, we use 3.16 kg CO_2/kg fuel [159] and 0.8 €/kg [160], respectively.

7.3 Simulation: Results

To provide an overview of the traffic situation, Figure 7.4 presents the hourly count of all flights for both multi-engine taxiing and tug-enabled taxiing operations over 17–18 July 2019. The two curves almost align, with MET operations displaying a slightly lower total count due to shorter taxi times, as discussed below. Additionally, the figure illustrates the count of arriving versus departing flights, showcasing the characteristic alternating trend between landings and takeoffs typical for a hub-and-spoke airport like Schiphol. In turn, this results in frequently changing RMO phases throughout the two days, represented by the colored shades in the figure.

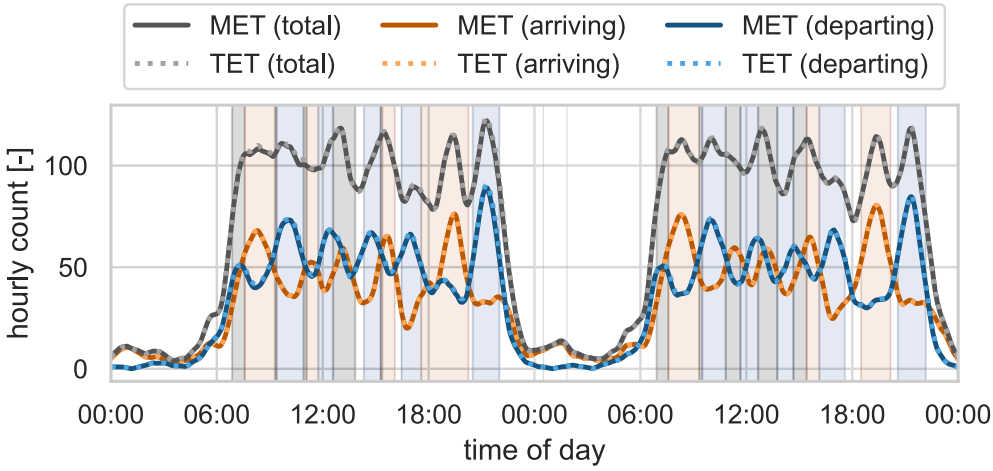


Figure 7.4: Hourly count of flights over the two days. Shades denote the RMO phase: off-peak (white), arrival-peak (orange), transition (grey), and departure-peak (blue)

Table 7.4 lists multiple key performance indicators (KPIs) that we will use in the following sections to compare the historical, MET, and TET operations for both days. Besides the median taxi time $\tilde{t}_{taxi}$, we report the maximal throughput and occupancy rate per hour for any runway in use, and the total estimated consumption of jet fuel in tonnes. Note that the throughput and occupancy of arriving flights are identical for all three cases, since the simulated operations are based on the historical landing times.

Table 7.4: KPIs of historical, multi-engine taxiing (MET), and tug-enabled taxiing (TET): median taxi time $\tilde{t}_{taxi}$, runway throughput (RWY thrpt.) and occupancy (RWY occ.) as maximal hourly value for any runway, and estimated fuel consumption in $[t]$

date		17-07-2019			18-07-2019		
	operations	hist.	MET	TET	hist.	MET	TET
ARR	$\tilde{t}_{taxi}$	04:06	03:18	03:15	10:46	09:28	09:23
	RWY thrpt.		42			38	
	RWY occ.		68.7 %			69.3 %	
	fuel [t]	-	38.6	37.8	-	92.7	92.2
DEP	$\tilde{t}_{taxi}$	14:32	12:35	14:00	10:08	07:31	07:49
	RWY thrp.	45	44	46	43	42	42
	RWY occ.	74.7 %	76.1 %	78.8 %	73.5 %	76.6 %	77.2 %
	fuel [t]	-	115.0	35.8	-	62.3	35.8

7.3.1 Comparison of Taxi Time Distributions per Runway

Figure 7.5 shows the taxi times of the different operational cases both for arrivals (left) and departures (right) per runway strip as box-and-whisker plots, illustrating the median, first and third quartiles as box, as well as outliers as whiskers and points. For MET operations, the observations from [152] hold for both layout versions: the simulated operations have shorter and less varying taxi times for all runways, with taxi times of departing aircraft varying more due to the difference in engine-start durations between small and large aircraft. Since the taxi times are almost identically distributed, we conclude that the changes to the airport layout have a negligible effect on MET operations.

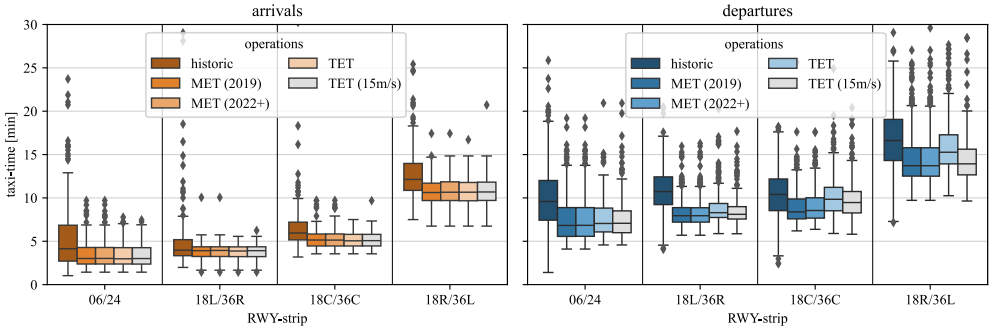


Figure 7.5: Box-and-whisker plot of historical and simulated taxi times for arrivals and departures per runway-strip

This allows us to compare TET operations on the adapted layout not only to MET but also to the historical values. Comparing MET to TET first, the taxi times of arriving aircraft are not affected by departing aircraft using TET. In contrast, the impact of TET varies for each departure runway strip: for runway 06/24, the taxi times are slightly higher but vary also slightly less. For 18L/36R and 18R/36L, the longer taxi times are mainly due to the smaller maximum speed of tugs, confirmed by experiment 'TET (15m/s)' that shows similar distributions to the MET case for both runways. The taxi times towards runway 18C/36C are longer and vary more, which is not only related to the smaller maximum

speed (as confirmed by experiment 'TET (15m/s)'). This means that tug-enabled taxiing leads to higher inefficiencies specifically at this runway, most likely due to the placement and accessibility of the tug decoupling points.

In summary, while the TET operations do not have an impact on arriving flights compared to MET operations, they result in 11.3 % and 4.0 % higher median taxi times of departing aircraft on day 1 and day 2, respectively (see Table 7.4). However, in comparison to historical operations, the taxi times decrease by 3.7 % on day 1 and 22.8 % on day 2, and vary less between flights for all runways. Furthermore, as listed in Table 7.4, the maximal runway throughput and occupancy rate do not change significantly between historical, MET, and TET operations. The multi-agent system is thus able to keep the runway capacity not only for MET but also for TET operations.

7.3.2 Predictability of Taxi Time

For the MET operations reported in Chapter 6, we argued that accurate A-CDM milestones help to keep the different airport processes synchronised. Consequently, we analysed the predictability of the remaining taxi time provided by the Routing Agent throughout the operations. The same trends as depicted in Figure **6.11** and discussed in Section 6.3.3 hold for TET operations as well. In summary, 30 min before reaching the taxiing destination, the predicted taxi time matches the actual taxi time of most flights, with some outliers showing deviations within ± 5 min. No deviations occur within the last 10 min of taxiing.

7.3.3 Delay Hotspots in the Taxiway Network

As depicted in Figure 7.6 and already observed in our previous study [152], MET operations show delay hotspots mostly in the bay areas due to engine-start and holding, the runway entries due to the necessary wake turbulence separation, and in front of the runway crossing of 18C/36C on the way to 18R/36L. In contrast, delays in TET operations occur mostly on the taxiways leading to a tug decoupling location. Furthermore, the delays are more concentrated as the accumulated delay is higher. While some holding still occurs in the bay areas, it is visibly much less than in MET operations.

Although the pattern close to runway 06/24 on day 2 differs between MET and TET, it does not impact the operations as confirmed by the similar taxi times discussed above. In comparison to day 1, more holding occurs in the bay areas: the decoupling locations are close to the bay areas and the flows to the two departure runways are more intertwined. Thus, the Routing Agent lets aircraft hold in bay areas to deconflict the concurrent routes.

7.3.4 Emission Hotspots in the Taxiway Network

For each day and for both MET and TET, Figure 7.7 shows the accumulated time per layout location that engines of departing aircraft are starting or running, which provides insights into the areas in which exhaust emissions including fine particles are mostly produced during taxiing operations. For MET operations, the highest emissions are observable in the bay areas and the taxiways leading to the runway entries. In contrast, TET operations shift the emissions close to the runways. This is especially the case on day 1, in which only in one bay area low emissions (below 10 min over 24h) occur, while high emissions are concentrated in front and after the tug decoupling locations.

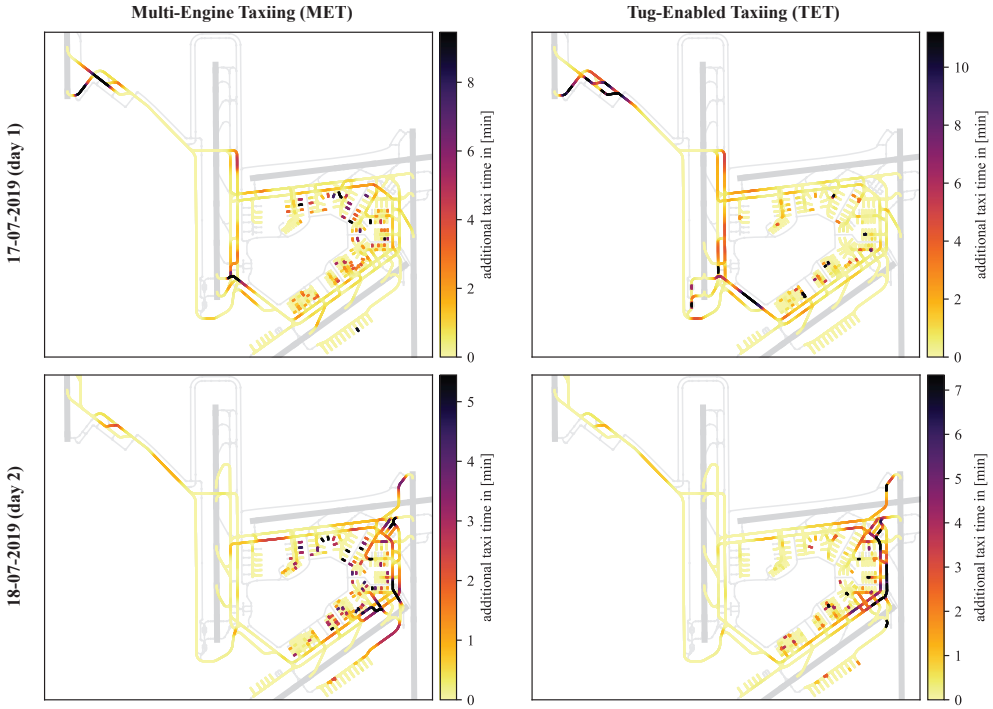


Figure 7.6: Delay hotspots depicted through additional taxi time per layout location for each day and each taxiing mode of operation

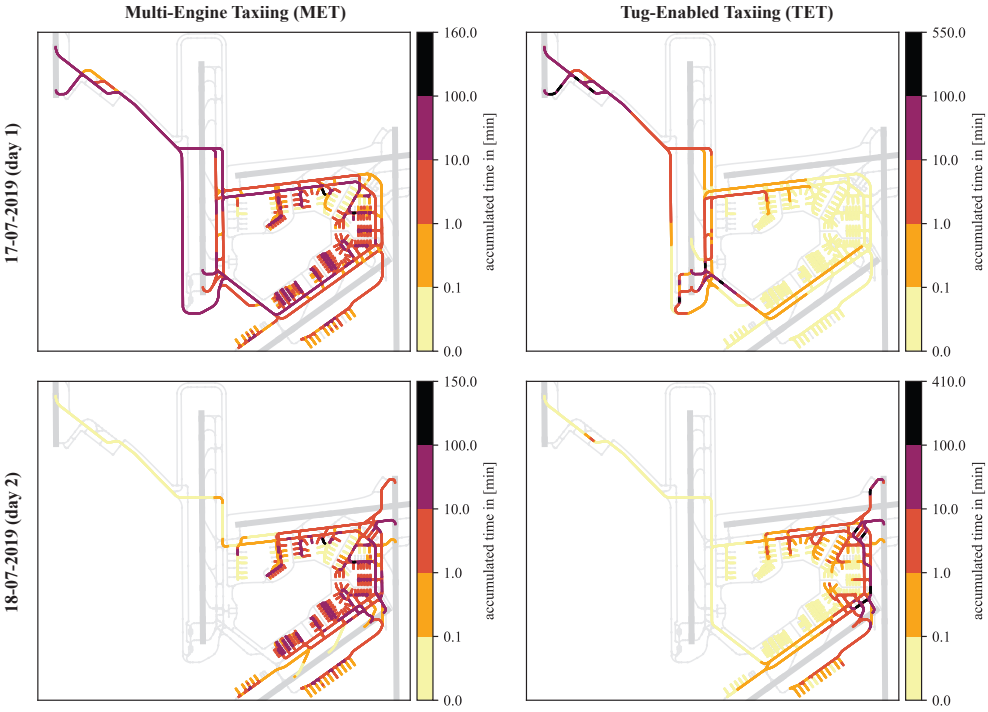


Figure 7.7: Emission hotspots depicted through accumulated engine on time per layout location for departing aircraft for each day and each taxiing mode of operation

On day 2, due to the short taxi distance to the decoupling locations in combination with the engine-start durations of 3 min for small and 6 min for large aircraft, low emissions still occur in most bay areas. Nonetheless, high emissions only arise around the decoupling locations.

7.3.5 Comparison of Engine-on Time per Runway

Figure 7.8 displays the distribution of engine-on times of departing aircraft for the different simulated cases as box-and-whisker plots per runway-strip. Note that the fixed engine-start times are excluded, since they merely add an offset dependent on the aircraft type (3 min vs. 6 min). Furthermore, the engine-on times of arriving aircraft are not shown as they are identical to the taxi times from Figure 7.5. While the engine-on times for departing flights using MET depend mostly on the distance between bay area and runway entry and thus the taxi time, the engine-on times for TET operations are similar for all runways, mostly between 1–2 min. They are also independent of the tug speed (confirmed by the case 'TET (15m/s)'), since the engines are only fully started after tug decoupling. Especially for runway 18R/36L, the engines thus only run a fraction of the times observed in the MET cases.

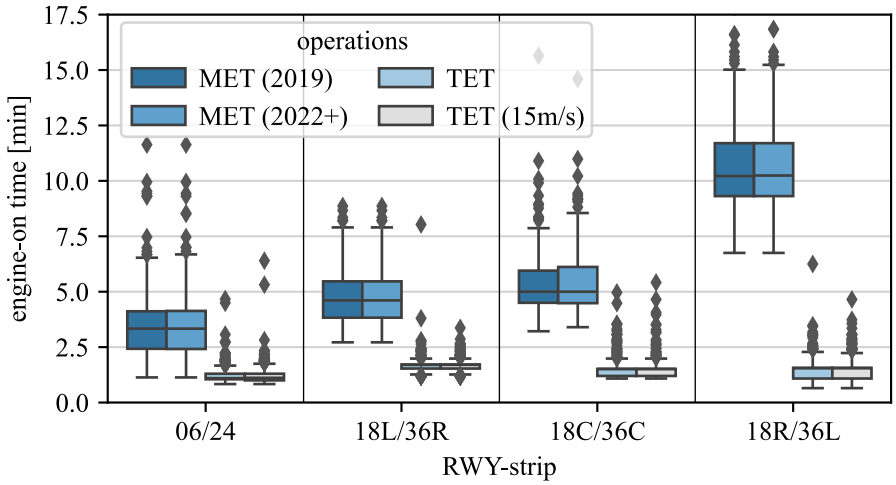


Figure 7.8: Box-and-whisker plot of engine-on times of departing flights per runway-strip

As rough estimate based on a fuel flow rate of 0.1 kg/s per engine, the total fuel consumption of departing flights decreases by 68.8 % on day 1, and 42.5 % on day 2 in comparison to MET operations. This reduces the CO_2 emissions by 250 t on day 1, and 84 t on day 2. For current jet fuel prices of 0.8 €/kg, this also lowers the total fuel costs by € 63 000 and € 21 000 for the two days, respectively. Using the median fuel consumptions per runway and number of engines, we project this onto the pre-covid yearly movements of around 250 k outbound flights at Schiphol: around 54.6 kt CO_2 emissions and € 13.8 million fuel costs could be saved per year, compared to MET operations. Since we used a rather low estimate of the fuel flow rate and did not model holding operations, the actual savings are likely higher.

7.3.6 Computational Efficiency of Routing Algorithm

Like our previous work, we used a Windows 10 laptop with 1.80 GHz Intel Core i7-10610U CPU and 16 GB RAM to run the Python-based simulations. For both MET and TET operations, the computational time increases in polynomial time with the number of agents. TET operations show slightly higher runtimes (mean: 42 s, 95th-percentile: 153 s, max: 435 s), likely because the algorithm must pick one of multiple tug decoupling locations as well as runway entry for departing flights, which increases the necessary computations.

7.4 Future Work

Based on this study, we plan to analyse different arrangements of the decoupling locations on the KPIs, especially those at runway 36C. Furthermore, we plan to study engine-off taxiing also for arriving aircraft to assess potential impact on the operations. Task assignment of tugs and directly routing them from a decoupling location to their next assignment are further directions.

Future research should also quantify different sources of uncertainty. Those with the highest impact should be included in path planning. Likewise, other ground vehicles that may especially impact the operations in the bay areas should be integrated into the model to study their impact.

7.5 Conclusion

In this chapter, we explored the operational consequences that the use of tugs for engine-off taxiing of departing aircraft pose. All movements on the airport surface were coordinated and controlled by a hierarchical multi-agent system. Its architecture with conflict-free routing and Guidance Agents to provide instructions to the Moving Agents yielded safe operations.

We showed that the runway capacity was mostly unaffected by tug-enabled taxiing. Although TET operations increase the taxi times towards runways far away from the bay areas, this is foremost due to the reduced taxiing speed of tugs. Compared to historical operations, the taxi times decreased by approx. 13 %. TET operations did not affect the taxi times of inbound flights, yielding a reduction of around 16.7 % compared to historical operations. The MAS thus efficiently coordinates the movements along the taxiway network and also around tug decoupling locations.

In comparison to multi-engine taxiing, TET operations reduced the congestion around bay areas. Furthermore, the exhaust emissions including fine particles decreased significantly in bay areas. Besides, TET operations also decreased the fuel consumption of outbound flights by 68.8 % for RMO North, and 42.5 % for RMO South. At Schiphol, this yields estimated savings of 54.6 kt CO_2 emissions and € 13.8 million fuel costs per year, based on current jet fuel prices. Therefore, TET operations offer the potential to significantly lower the environmental footprint of taxiing operations and health risks of ground personnel, while sustaining safety and efficiency levels.

The tug-enabled taxiing case study presented in this chapter addressed RQ 5. The next chapter strengthens the operational realism of the underlying modelling assumptions by incorporating feedback gathered from ATCOs and other operational experts throughout the course of this thesis. Historical aircraft tracks are analysed to identify and quantify key movement characteristics and to recalibrate the model parameters (RQ 6). In addition, the MAS model is extended to represent aircraft towing movements and to assess their interaction with regular traffic (RQ 7). Building on these modifications, multiple operational model variations are then compared and discussed.

8

Improving the Realism: Trajectory-based Calibration, Towing Operations, and Operational Variations

This chapter improves the parameter calibration of and extends the MAS model using Amsterdam Airport Schiphol as a case study, addressing RQ 6 and RQ 7. First, historical ground tracks are projected onto a detailed taxiway network, to calibrate aircraft dynamics and derive more accurate A-CDM milestones, allowing the MAS model to better reproduce the observed taxi traffic. Then, aircraft towing movements are incorporated into the MAS model, to assess their impact on the taxi performance of regular traffic and the potential to increase connected stand capacity. Finally, operational variations are analysed, revealing that both reduced takeoff separations and more progressive taxi speed limits can improve operational performance by around 10 %. Across all analyses, departure runways emerge as the dominant bottleneck at Schiphol, while realistic speed profiles and selective towing offer operationally relevant efficiency gains.

8

This chapter is based on the manuscript 'Improving the Realism of Multi-Agent Simulation Models for Airport Surface Movement Operations: Trajectory-based Calibration, Towing Operations, and Operational Variations', which will be submitted to the *Journal of Air Transport Management*.

A preprint version is available at:

M. von der Burg and A. Sharpanskykh, 'Improving the Realism of Multi-Agent Simulation Models for Airport Surface Movement Operations: Trajectory-based Calibration, Towing Operations, and Operational Variations', 2026. DOI: 10.2139/ssrn.6712278.

As a consequence of growth in air traffic, environmental constraints, infrastructure limitations, new technology, and staff shortages, airside operations at major airports are becoming increasingly complex [8], [161]. In turn, their surface congestion is expected to rise, potentially reducing the predictability of taxi times as well as creating knock-on effects across the wider network [10]. This may also lead to strong increases in workload of Air Traffic Control Officers (ATCOs), negatively affecting operational efficiency and safety [9].

Fast-time simulators have emerged as indispensable tools in this domain, enabling researchers and practitioners to analyse and refine airport surface movement operations (ASM Ops) under controlled, yet realistic, conditions. Moreover, digital twins as virtual, real-time representations of airport operations promise to support decision-making, process optimisation, and the evaluation of innovative operational concepts [162]. However, the efficacy of such simulations hinges on the use of high-fidelity models capable of realistically capturing the innate complexities of ASM Ops in detail.

Agent-based modelling and simulation is particularly suitable for ASM Ops, as system performance emerges from the interactions of many heterogeneous actors, including aircraft, ground vehicles, ATCOs, pilots, and airport resources. By representing these actors as agents with individual states, objectives, constraints, and decision logic, such models can capture local interactions, coordination mechanisms, and their system-level effects [163]. To evaluate novel concepts of operations in detail, a computational multi-agent system (MAS) model was therefore developed, as presented in Chapter 3. The generic framework has a hierarchical-distributed architecture, in line with the hierarchical sequence of ASM Ops tasks. These range from strategic planning (long-term and high-level decisions) to scheduling (tactical allocation of resources), routing (tactical planning of movements), guidance (operational planning of movements), and finally movement execution (controlling aircraft and ground vehicles). The hierarchical-distributed structure allows for effective coordination across all levels of control, owing to the stringent safety requirements in ASM Ops.

Serving as illustration of the generic framework, Section 3.2 elaborated one MAS model instance with centralised path planning as coordination strategy. In the previous chapters, this model was used to study the operational implications of different concepts of operations for fully-automated ASM Ops and engine-off taxiing Chapters 5 to 7. Additionally, the model served as basis for the decision-support systems designed as part of two SESAR 3 JU projects [38], [44].

As part of the face validation of this MAS model, elaborate discussions were carried out with ATCOs and operational experts from Amsterdam Airport Schiphol. While they highlighted the overall realism of the simulations, they also provided valuable feedback regarding the following points:

1. The modelled aircraft speeds appeared too high in turns and, for large aircraft, across the taxiway network.
2. In present-day operations, aircraft usually stop at the runway holding point while awaiting their takeoff clearance, whereas all aircraft were assigned a rolling-takeoff procedure in the MAS model.
3. The model does not yet include aircraft towing movements frequently taking place at Schiphol to free up capacity of connected gates, and move aircraft to or from hangars as well as long-term parking positions; however, these may limit routing options for regular taxiing traffic and thus affect airside efficiency.

These observations are closely related to, but not fully addressed by, the existing literature. Regarding point 1), most previous work uses fixed speed limits for both straight taxiway segments and turns across the entire taxiway network [132]. Historical surface movement trajectories, obtained from sources such as ADS-B, multilateration, or surface-movement radar, provide a valuable basis for calibrating and validating ASM Ops models. Recent data-driven approaches have used such historical data for taxi time prediction [164], [165], departure metering [86], and trajectory optimisation [166], [167]. These methods are, however, generally trained on observed aircraft behaviour and therefore aim to reproduce historical dynamics. Such trajectories are shaped not only by aircraft performance, but also by coordination with surrounding traffic, communication between ATCOs and pilots, individual pilot behaviour, and prevailing operational procedures. Therefore, while historical trajectories are essential for calibration and validation, simply setting speed limits based on averages of the observed track speeds is insufficient to define the taxiing speeds achievable under higher levels of automation.

Regarding point 2), research studying decision-support concepts such as departure metering rely on models that maintain adequate runway pressure, i.e. a sufficient queue of aircraft at the runway threshold to exploit departure opportunities while controlling taxi-out delay (e.g. [86], [168]). Runway occupancy time (ROT) is a key determinant of departure capacity, and both technical studies and operational guidance highlight rapid line-up and, where appropriate, rolling takeoffs as means to reduce ROT. However, detailed empirical analyses of how often rolling versus static takeoff techniques occur at large hubs, and how they interact with taxiway operations, remain limited in the open literature.

Regarding point 3), aircraft towing movements are usually minimised as one of the optimisation objectives in stand assignment models, typically alongside stand preferences and schedule adherence (e.g. [169]–[171]). These studies justify towing reduction mainly by referring to the additional operational burden, cost, and coordination effort associated with repositioning aircraft between stands. However, despite its wide adoption, there is little evidence on whether towing operations actually influence the operational efficiency or flow of taxiing aircraft. Existing gate-assignment formulations treat towing largely as a local penalty, whereas studies on surface movement operations generally omit towing flows altogether. Consequently, the system-level implications of towing, such as potential interactions with taxiing traffic, additional congestion, or delay propagation, remain insufficiently quantified.

Motivated by the feedback from operational experts as well as the identified research gaps, this chapter focuses on three aspects, using Amsterdam Airport Schiphol as a case study. First, the model parameters are calibrated using historical ground trajectories (Section 8.1). Based on estimated key performance indicators, including taxi times and infrastructure occupancy rates, the similarity between historical and simulated operations is then assessed. Second, the MAS model is extended with aircraft towing movements to quantify their effect on operational performance (Section 8.2). This analysis considers the impact on inbound and outbound taxi times, as well as changes in the occupancy rates of runways, taxiways, bay areas, and stands. Third, model variations are defined to evaluate the operational performance gains that could result from reduced takeoff separations and higher speed limits on taxiways (Section 8.3). Each section discusses the main findings and identifies directions for future work. Finally, Section 8.4 summarises the contributions of this chapter and their relevance for airport airside operations more broadly.

8.1 Calibration of Model Parameters using Historical Track Data

In this section, we first outline the developed methodology to project the proprietary track data provided by the Dutch air navigation service provider LVNL onto the computational graph representing the taxiway network in the MAS model. The available data match the flight schedules of the two operational days in 2019 used in Chapters 6 and 7. In Section 8.1.2, we evaluate the historical data to calibrate the modelling parameters for fast-time simulations with the MAS model. In Section 8.1.3, we compare the performance of the resulting simulated operations to the historical ones with respect to taxi times, taxi distances, and occupancy rates of different parts of the infrastructure.

8.1.1 Methodology for Evaluating Historical Track Data

The data provided by LVNL contains all flight tracks within a 3 NM radius of the aerodrome reference point (ARP), and has the following data fields: timestamp, aircraft registration, position as x- and y-coordinates relative to the ARP, movement type, speed, and heading. Although the standard recording rate is 1 Hz, there are missing entries as well as some corresponding to the same timestamp.

In the following, we outline the methodology used to project the historical track data onto the taxiway graph of the MAS model, to extract information from the data. The stepwise approach for projecting the track data is visualised in Figure 8.1.

Pre-processing and Most Plausible Path

As a pre-processing step, we retrieve radar tracks of taxiing aircraft by matching aircraft registrations to the flight schedule data. Upon inspection of the raw tracks, it becomes apparent that the reported A-CDM milestones, such as the Actual Off-Block Time (AOBT), are inconsistent for some flights, as illustrated in Figure 8.1a. Moreover, the speed and heading values recorded at low ground speeds are unreliable, as confirmed by the comparison with the projected track in Figure 8.2.

To capture pushback movements of departing flights, the location at which the aircraft reverses direction after pushback must be detected. Since the raw speed entries are unreliable, we estimate speeds as $v_{i,\text{est}} = (\Delta d/\Delta t)_{i,i+1}$, where $\Delta d_{i,i+1}$ is the Euclidean distance between successive track points i and $i+1$ after smoothing with a Savitzky–Golay filter [172], [173], and $\Delta t_{i,i+1}$ is the corresponding time difference. Contiguous sequences of points for which $v_{\text{est}} \leq 1\,\text{m/s}$ for at least five seconds are then marked as stop locations.

Next, we infer the most plausible path as sequence of graph edges that best matches the track while respecting the detected stops. For this purpose, we use an A* search [174] that minimises the cumulative Euclidean distance between each position and the candidate graph edges. Except at stops, successive edges must be consistent with the movement direction, preventing operationally infeasible sharp turns. The search therefore tracks direction switches and penalises excessive reversals (>0 for arriving and >1 for departing flights). The distance to the nearest edge is used as search heuristic. Because position noise is typically lower near runways than in bay areas [175], the search is executed from runways towards bay areas, independent of the actual flight direction. For the example in Figure 8.1, the resulting path is highlighted in green in (a).

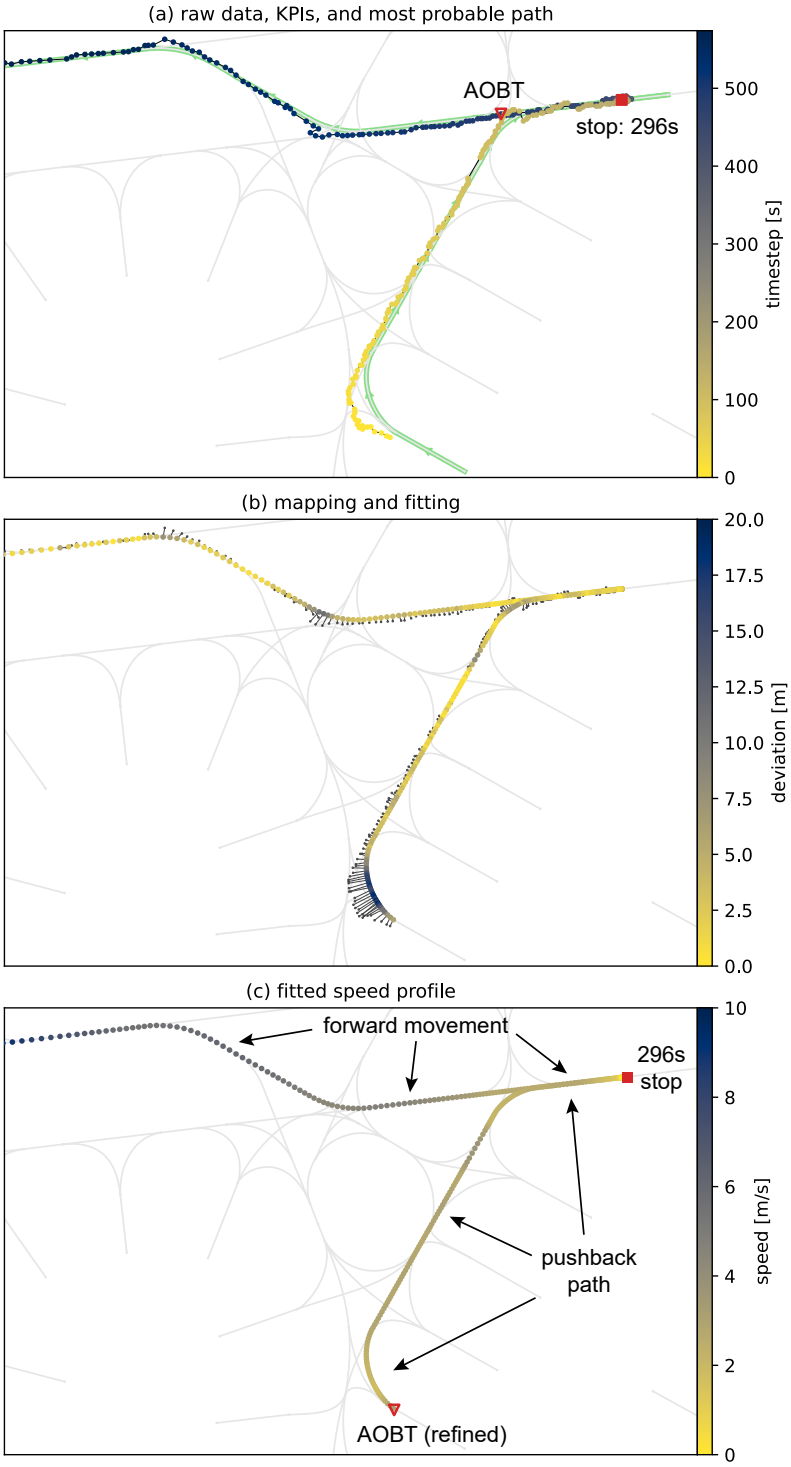


Figure 8.1: Illustration of methodology for evaluating historical track data

Mapping and Fitting

In the projection step (b), we map each track position to a point on the inferred path using a second A* search. The path is discretised into 0.1 m segments, and the search enforces temporal consistency by requiring the travelled distance along the path to increase monotonically. This results in a one-dimensional signal (travelled distance over time), enabling smoothing with a one-dimensional Savitzky–Golay filter. Due to missing entries, we apply a non-uniform implementation.

The precision of the projection is quantified as the Euclidean distance between each raw position and its projected counterpart. Overall precision is high (arriving flights: $99\,\% < 17\,\text{m}$, $75\,\% < 5\,\text{m}$; departing flights: $99\,\% < 32\,\text{m}$, $75\,\% < 6\,\text{m}$), but occasional outliers remain. Departing flights show larger errors, consistent with higher position noise in bay areas. Additional offsets arise because transponder locations (often near the aircraft nose) do not coincide with taxiway centrelines, particularly in sharp turns and during pushback. The offsets are visualised in (b) by the colour-coded projected points and their black connectors, and reported over the full trajectory in Figure 8.2 (red line). The largest offset in the example departure occurs near take-off and corresponds to a position outlier.

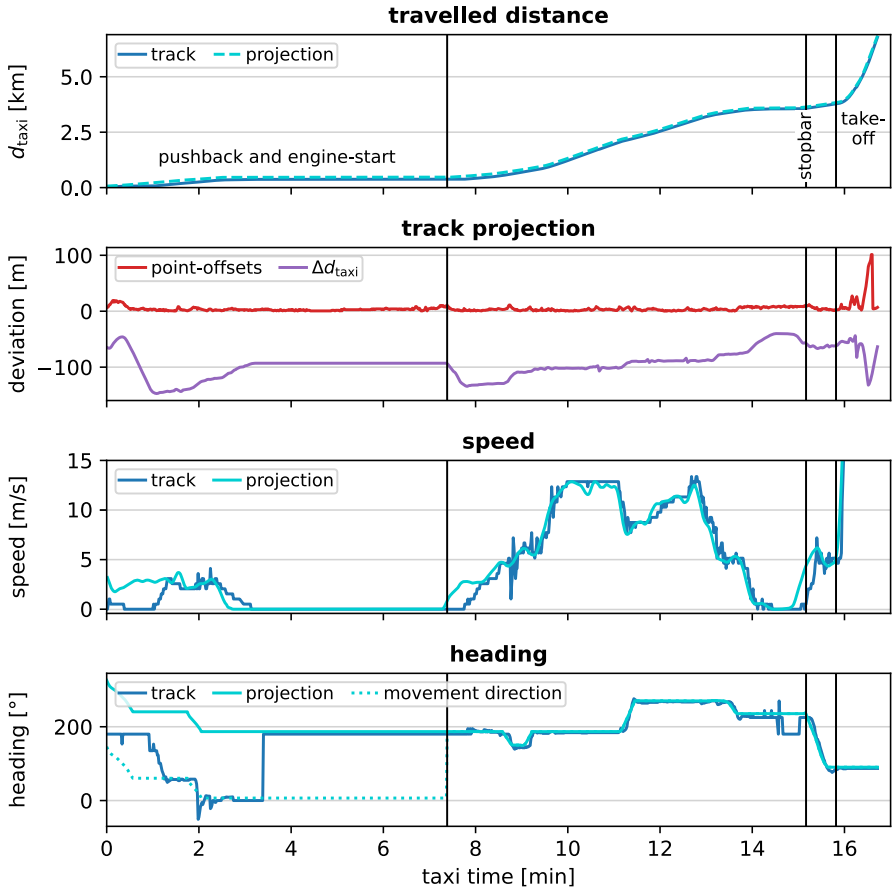


Figure 8.2: Projection KPIs of exemplary track of departing flight shown in Figure 8.1

Estimating Key Performance Indicators

For each projected trajectory, we derive key performance indicators (KPIs) for the full ground movement, selected phases, or individual graph edges. These include taxi duration, speed profiles, stop locations, and more accurate A-CDM milestones, with examples shown in Figure 8.1c. Figure 8.2 further reports time series of travelled distance, projection error (point offsets and Δd_{taxi} as the difference between raw and projected distances), speed, and heading. Overall, the projected trajectories yield more reliable measures of aircraft surface movements than the raw tracks.

The KPIs are further used in the analysis of the historical data carried out below as well as in the comparison to simulations conducted with the MAS model in the subsequent sections.

8.1.2 Evaluation of Historical Track Data

In the following, we evaluate the projected trajectories of the 48 h flight schedule to calibrate the MAS model parameters. In the following, we first assess the prevalent speed ranges in different areas of the taxiway network of Amsterdam Airport Schiphol. The respective taxiway areas are visualised in Figure 8.3. We then investigate the common aircraft behaviour for exiting and entering the runway, including the takeoff-roll and minimum time-based separations between takeoffs. Finally, we shed light on the stand use and pushback operations reflected in the trajectory data.

Historical Aircraft Speeds per Taxiway Area

To inspect the aircraft speeds per taxiway area, we determine the set of edges that fall within the respective area. Per movement along an edge for which the mapping precision remains $\leq$ 20 m, we calculate one average speed value. Figure 8.4 shows boxplots of all speed values per flight direction (arriving vs. departing flights) and per ICAO-group, for which we aggregate the values of aircraft types ICAO-B and ICAO-C, ICAO-D and ICAO-E, as well as ICAO-F, respectively.

As seen in the figure, aircraft of all types taxi with reduced speed along taxiways in bay areas ('bay TWYs') and those used primarily by cargo aircraft ('cargo-only TWYs'), matching the feedback from ATCOs. Some flights of the types ICAO-B/C taxi faster on the inner, midfield, and outer taxiways than the previously assumed general speed limit of 15 m/s. While some ICAO-D/E aircraft reach speeds exceeding 15 m/s on outer taxiways, they stay below the general speed limit of 15 m/s on the midfield and inner taxiways. ICAO-F aircraft show reduced taxiing speeds in general.

As consequence of these insights, speed limits should be set based on aircraft type and taxiway area in simulation models intending to realistically represent the maximum speeds observed in actual operations. Thus, for conducting simulations with the MAS model, we define speed limits per aircraft type on the edges corresponding to each taxiway area as listed in Table 8.1. To this end, we choose values based on upper percentiles in relation to the count of speed observations, as provided in the table for reference. Moreover, Table 8.1 also lists the historical speed medians $\tilde{v}_{hist}$, which are further used when comparing the taxi times of historical and simulated operations in Section 8.1.3.

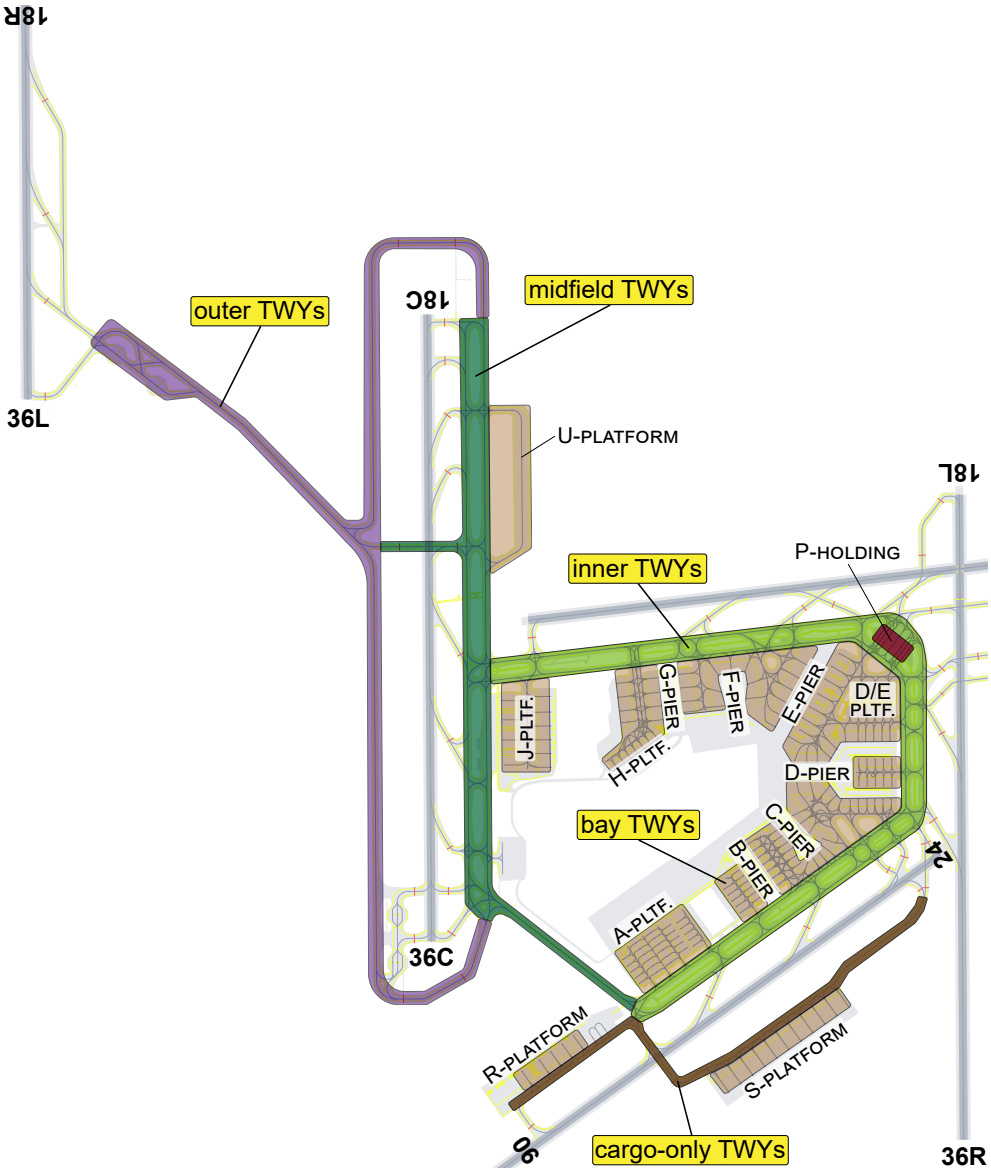


Figure 8.3: Overview of taxiway network of Amsterdam Airport Schiphol with shaded taxiway areas

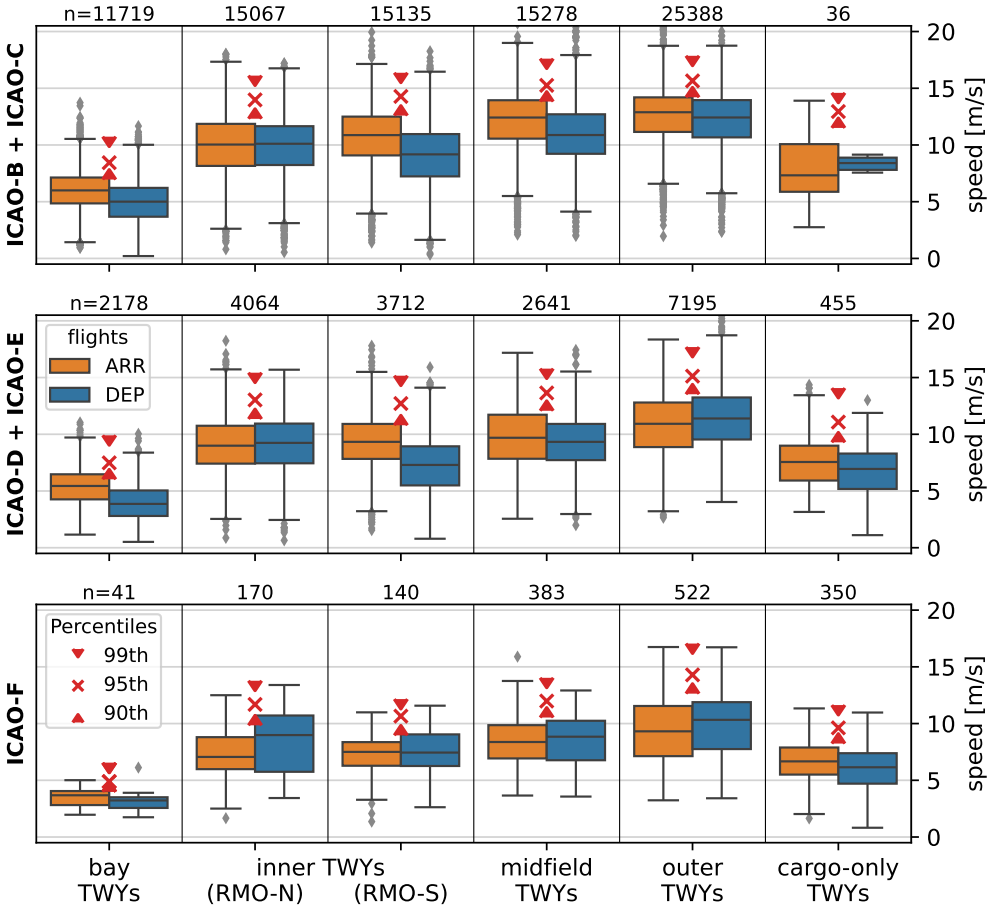


Figure 8.4: Historical speed data per aircraft type for both arriving (ARR) and departing (DEP) flights across the taxiway areas visualised in Figure 8.3, with 'TWY turns' denoting the turns in between taxiways A and B as part of the area 'inner TWYs'

Table 8.1: Maximum speeds $v_{\max}$ in [m/s] based on reference percentiles P_{ref} and historical speed medians $\tilde{v}_{\text{hist}}$ per aircraft type and different taxiway areas

		bay TWYs	inner TWYs	midfield TWYs	outer TWYs	cargo TWYs
ICAO-B, ICAO-C	P_{ref}	99th	99th	99th	99th	90th
	$v_{\max}$	9.5	15.0	16.5	17.0	12.0
	$\tilde{v}_{\text{hist}}$	5.5	10.0	11.7	12.7	8.0
ICAO-D, ICAO-E	P_{ref}	95th	95th	95th	95th	95th
	$v_{\max}$	7.5	12.5	13.5	15.0	10.5
	$\tilde{v}_{\text{hist}}$	4.5	9.0	9.5	11.2	7.3
ICAO-F	P_{ref}	90th	90th	95th	95th	95th
	$v_{\max}$	4.5	10.0	12.0	14.0	9.5
	$\tilde{v}_{\text{hist}}$	3.5	7.5	8.5	10.0	6.4

To determine speed limits on curved taxiway segments, the approach of von der Burg and Sharpanskykh [132] is followed by using the equation:

$$v_{\text{lim},\kappa,\epsilon} = \sqrt{\bar{r}_\epsilon \cdot a_{\text{lat},\kappa}} \tag{8.1}$$

with the average radius of curvature $\bar{r}_\epsilon$ of each curved taxiway edge $\epsilon \in \mathcal{E}$ and a fixed lateral acceleration $a_{\text{lat},\kappa}$ per aircraft type κ. To define suitable values for the lateral accelerations, we draw on the respective historical distributions, calculated with $a_{\text{hist}} = \bar{v}_\epsilon^2 / \bar{r}_\epsilon$, where $\bar{v}_\epsilon$ denotes the average speed over an edge ϵ which is a curved taxiway segment on or between taxiways A, B, C, D. The historical distributions are shown in Figure 8.5 for both arriving and departing flights for each of the three aggregated aircraft types. Table 8.2 lists the maximum lateral acceleration $a_{\text{lat,lim}}$ based on the 90th percentiles as well as the historical medians $\tilde{a}_{\text{lat,hist}}$.

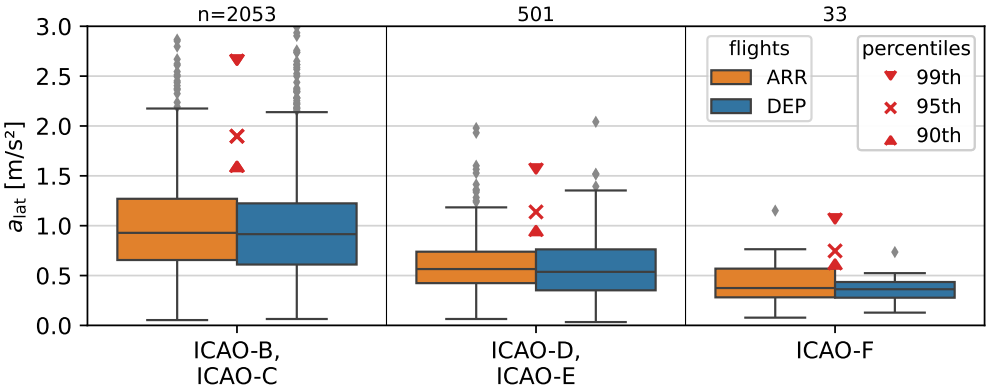


Figure 8.5: Distributions per aircraft type of historical lateral accelerations in curved taxiway segments for both arriving and departing flights

Table 8.2: Maximum lateral accelerations a_{lat} in [m/s²] based on reference percentiles P_{ref} and historical medians $\tilde{a}_{\text{lat,hist}}$ per aircraft type to calculate curve speed limits

	ICAO-B/C	ICAO-D/E	ICAO-F
P_{ref}	90th	90th	90th
$a_{\text{lat,max}}$	1.60	0.99	0.66
$\tilde{a}_{\text{lat,hist}}$	0.92	0.55	0.36

Runway Exit Use and Speeds at Exits

Figure 8.6 visualises the share per runway of exits taken by landing aircraft during the two operational days. Most flights use one of the rapid exit taxiways (RETs) to quickly vacate the respective runway. With limited RETs available at runway 36R, especially larger aircraft often use the conventional exit taxiways (CETs) that are located closer to the end of the runway.

Furthermore, the figure summarises the observed speeds at the runway exit point (RWY exit) and at the subsequent stopbars per aircraft type and exit category. As expected, aircraft

reduce their speed towards the exit stopbar, and vacate the runway with higher speeds when using one of the RETs. Nonetheless, the average speed that aircraft of the same category reach at stopbars is mostly independent of the type of exit used (CET vs. RET), despite varying strongly between flights. Large aircraft have generally slower speeds when vacating the runway, and correspondingly cross the stopbars with lower speeds.

Overall, the observed speed ranges at the stopbars fall within the values listed in Table 8.1 when accounting for the subsequently entered taxiway areas. For the MAS model, it is thus not necessary to further differentiate the speed limits on graph edges that succeed runway exits.

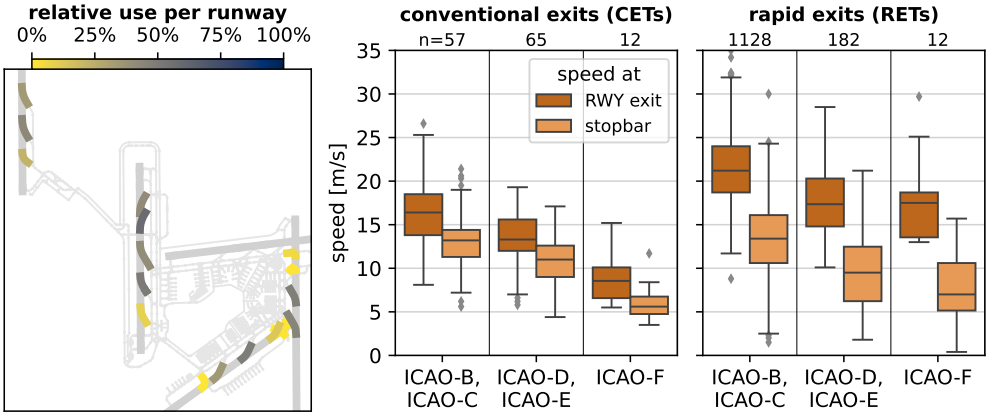


Figure 8.6: Historical landing speeds per aircraft type and type of runway exit

8

Runway Entry Use and Speeds during Takeoff-Roll

Figure 8.7 visualises the share per runway of entries used by departing flights during the two operational days. Aircraft are primarily assigned the first runway entry, with a few flights entering the runways via one of their RETs. The first entry of runway 18L requires aircraft to taxi longer in comparison to its second and third entry, which are thus sometimes assigned to smaller aircraft requiring less runway length for takeoff. To mitigate wake turbulence risks from aircraft taking off from runway 24 onto those from runway 18L, aircraft must not use the first entry of runway 24 when these two runways are in use at the same time, resulting in the larger share of the second and third entry of runway 24.

To analyse the aircraft speeds during their takeoff-roll preceding the actual takeoff, we use the minimum speed $v_{\text{min,rwy}}$ observed in the final part of the projected trajectory after passing the stopbar. As shown in Figure 8.7, we differentiate between three takeoff categories: static takeoff for $v_{\text{min,rwy}} < 0.5\,\text{m/s}$, slow rolling for $0.5\,\text{m/s} \leq v_{\text{min,rwy}} < 5\,\text{m/s}$, and fast rolling when $v_{\text{min,rwy}} \geq 5\,\text{m/s}$. Prior to takeoff, between 32 % to 55 % of all aircraft stop on the runway or one of its entries. Meanwhile, the remaining flights keep rolling with low (33 % to 50 %) or even relatively high speeds (3 % to 18 %). The extracted speed data suggests that large aircraft stop less frequently prior to takeoff, although the small sample size within the available tracks prevents to draw a definitive conclusion.

During path planning in the MAS model used in Chapters 5 to 7, the 4D trajectories of departing flights were deconflicted in a way that allows them to use a rolling takeoff.

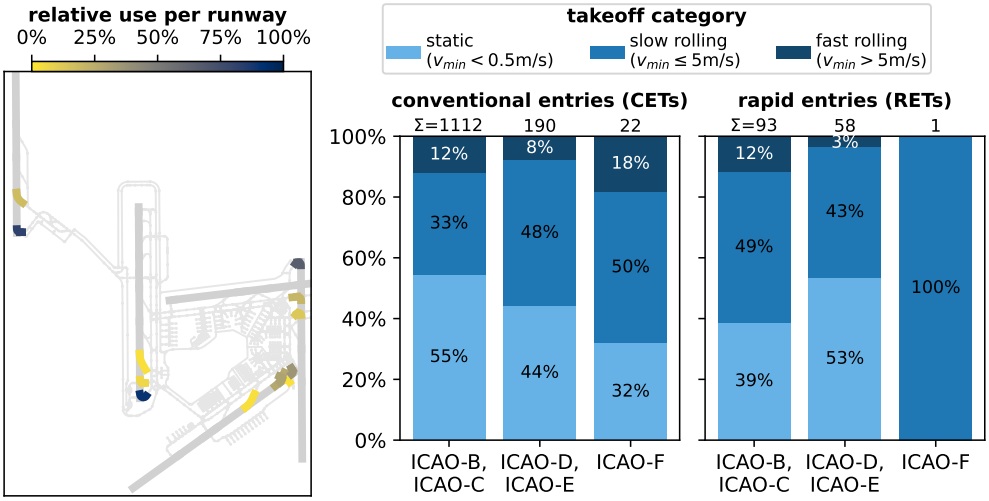


Figure 8.7: Share of takeoff categories of historical flights per aircraft type and type of runway entry

Since the conducted analysis underscores that a certain share of historical flights does also not stop prior to takeoff, changing this approach does not seem necessary. Nonetheless, the projected trajectories allow for assigning the historical entries and exits when simulating the underlying flight schedule, as done in Section 8.1.3.

8

Temporal Separation between Takeoffs

Irrespective of the speed prior to takeoff, consecutive flights taking off from the same runway must keep a minimum separation to mitigate wake turbulence effects. To this end, ICAO has specified minimum times that certain combinations of the four wake turbulence categories (WTC) light (L), medium (M), heavy (H), and super heavy (J, only the A380) must keep to each other [176]. When a minimum separation is required, another minute is added in case the subsequent aircraft uses a runway entry with less remaining takeoff length, referred to as intersection takeoff (IS-takeoff). Besides accounting for wake turbulence, aircraft must also keep sufficient separation after takeoff. Thus, when two consecutive takeoffs follow the same standard instrument departure (SID) route, ATCOs increase the temporal separation between them.

Differentiating between IS-takeoffs, diverging SID routes, and equal SID routes, Figure 8.8 shows the distributions of the separation times t_{sep} between those WTC-pairs for which sufficient data points could be extracted from the projected trajectories. From these, an ICAO time-based separation minimum only exists for the pair H → M, marked by red lines in the figure. Relevant IS-takeoffs occurred merely for pairs H → M and H → H, and do not show increased separation times on average. For equal SIDs, the historical separations of these two pairs do not indicate increased times for equal SIDs, whereas an additional 25 s on average are kept for pairs M → M and M → H. Interestingly, for the latter two pairs, the shortest observed separations are below 60 s.

Table 8.3 lists the average values for the four pairs, comprising also values based on a larger dataset provided in [177]. Overall, the extracted values are in good agreement with those found in literature.

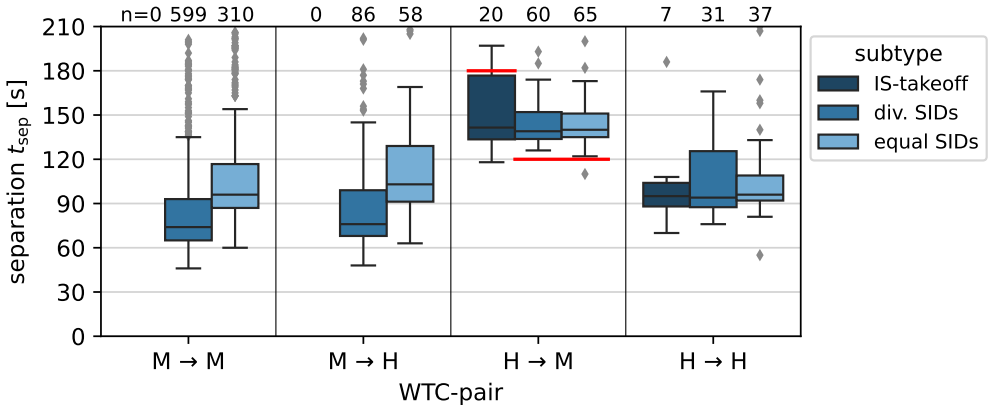


Figure 8.8: Temporal separation of historical takeoffs grouped by ICAO wake turbulence categories (WTC) [176]. Each WTC-pair is subdivided into intersection (IS) takeoffs, as well as following diverging or equal standard instrument departure (SID) routes. Red lines indicate the respective minimum separation times.

Table 8.3: Time-based separations between takeoffs in [s]

	ICAO-WTC	diverging SIDs		equal SIDs	
	[176]	[177]	Figure 8.8	[177]	Figure 8.8
M→M	n/a	72	74	91	96
M→H	n/a	78	76	96	103
H→M	120	138	139	138	140
H→H	n/a	92	94	97	96

8

Stand Use and Pushback Operations

Based on the historical flight schedule data of the two operational days, the number of aircraft parked at stands differed between piers, as visualised in Figure 8.9. Piers suited only for smaller aircraft such as the B-pier and D-pier accommodated a higher number of aircraft, whereas large stands such as those of the E-pier saw a smaller amount within the 48 h. Only a few aircraft arrived and departed from the remote parking positions at the D/E, G, and J platforms as well as the cargo piers. Especially at the remote parking positions, more aircraft might have been parked between the respective arriving and departing flights, but no tracks were available to trace any operational re-parkings of inactive aircraft.

At Amsterdam Airport Schiphol, a standard pushback path is defined per stand, with potential variations between aircraft categories, as outlined in the official pushback charts [153]. In the MAS model used in Chapters 6 and 7, the pushback operations were assumed to comply with these standard paths. In present-day operations, air traffic controllers may assign deviating pushback paths for various reasons like operational efficiency gains, safety

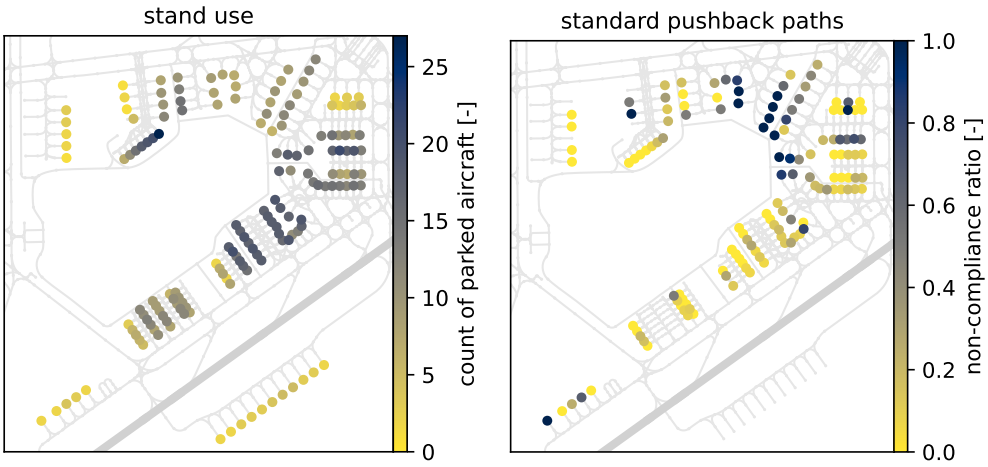


Figure 8.9: Historical stand use (left) and non-compliance rate to standard pushback path per stand (right)

reasons, or blocked taxiway segments among others. However, such reasons cannot be extracted or inferred from the historical tracks.

Based on the projected paths of the historical tracks, the amount of non-standard pushbacks differs per stand, as visualised as ratio over the two operational days in Figure 8.9. From all 1 483 historical tracks of departing flights, the pushback paths of 86.4 % could be extracted. From these, 23.3 % do not comply with the standard pushback paths, being either shorter (5.5 %), longer (6.1 %), or generally divergent (11.7 %).

8.1.3 Comparing Calibrated Model to Historical Operations

Based on the gathered insights from analysing the historical tracks, the modelling parameters of the MAS model are calibrated, as outlined in the following. We then evaluate how similar the resulting taxi times, distances, and average taxi speeds per taxiway category are between the simulated and historical operations.

Calibrated Parameters of Multi-Agent System Model

Previously, aircraft agents in the MAS model were spawned using historical A-CDM milestones. Arriving flights were assigned those runway exits for which the landing distance exceeds the minimum of either rapid-exit or conventional taxiways per modelled aircraft type (ICAO-A to ICAO-F). Likewise, potential runway entries for departing flights were assigned based on minimum takeoff distances per modelled aircraft type.

Where possible, we extract performance data per flight using the projected historical tracks. For arriving flights, we determine the time of exiting the runway as well as the respective graph edge underlying the runway exit. For departing flights, we use the time point of exiting the stand as estimated Target Off-Block Time (TOBT), and the crossed runway stopbar to determine the edge for entering the runway, with any runway entries allowing for greater takeoff distance added as alternative goal locations.

Since the historical data analysis showed that some flights use a rolling takeoff, we keep the approach of coordinating taxi routes so that departing aircraft can always use

a rolling takeoff. To ensure sufficient separation between consecutive takeoffs, we apply the separation times from [177] listed in Table 8.3 that differentiate between diverging and aligned standard instrument departure (SID) routes. An additional minute is added for flights that take off with less available runway distance than the previous flight, i.e. those that use an intersection takeoff.

As speed limits for straight taxiway segments, we use the historical median speeds $\tilde{v}_{\text{hist}}$ per aircraft type according to the respective taxiway areas as listed in Table 8.1. We calculate the speed limits on curved segments using Equation (8.1) with the historical median lateral acceleration $\tilde{a}_{\text{lat,hist}}$ per aircraft type as listed in Table 8.2. In line with Section 6.1.4, all aircraft are assumed to accelerate with $0.4\,\text{m/s}^2$ and decelerate with $-0.5\,\text{m/s}^2$.

To calculate deconflicted routes between all aircraft taxiing concurrently on the airport surface, the AS-MAMP algorithm outlined in Chapter 4 is used with a planning window of $w_{\text{p}} = 20\,\text{min}$ and a replanning period of $h_{\text{p}} = 10\,\text{min}$. Based on von der Burg and Sharpanskykh [137], $c_{\text{TWY}} = 1.2$ is set as cost-factor to conform more closely with the standard taxiway flow directions used by ATC at Schiphol. Furthermore, the model extensions for inbound and outbound holding to resolve stand conflicts and comply with CTOT-slots, respectively, are used as described in Section 6.1.2.

Hourly Movements

Providing a general overview of the simulated ground traffic at Amsterdam Airport Schiphol, Figure 8.10 shows the hourly movements over the 48 h of simulated flight schedule. The alternating peaks in arriving and departing flights closely match the corresponding runway configuration phases with inbound-peaks (orange shades), transitions (grey shades), and outbound-peaks (blue shades). In the most prominent peaks, more than 120 movements take place per hour.

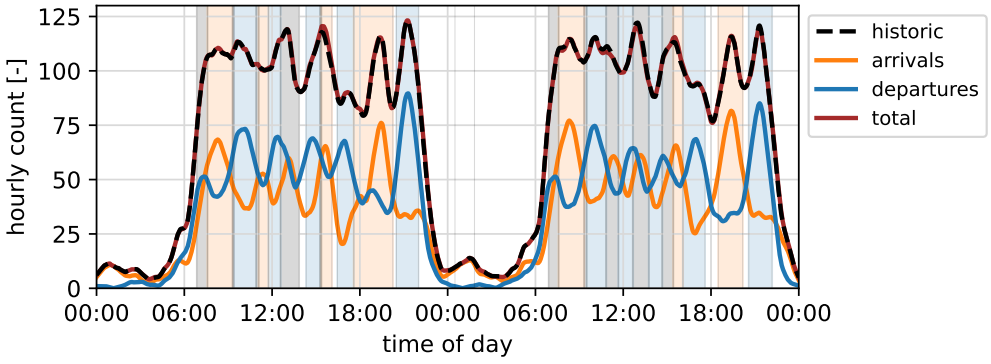


Figure 8.10: Hourly ground movements over the simulated 48 h. Shades denote the runway configuration phase: off-peak (white), arrival-peak (orange), transition (grey), and departure-peak (blue).

Taxi Times

In the following, the taxi times of both arriving and departing flights from three datasets are compared: the historical A-CDM milestones, the obtained milestones based on the projected tracks, and those from the simulated operations. From each of the three

datasets, flights were excluded whose incomplete tracks did not allow to extract the relevant KPIs. To further equalise the datasets, any flights were excluded for which either inbound or outbound holding was issued in the simulated operations, which are compared separately in Section 8.1.3. In Figure **8.11**, the resulting taxi time distributions are shown as box-and-whisker plots per runway.

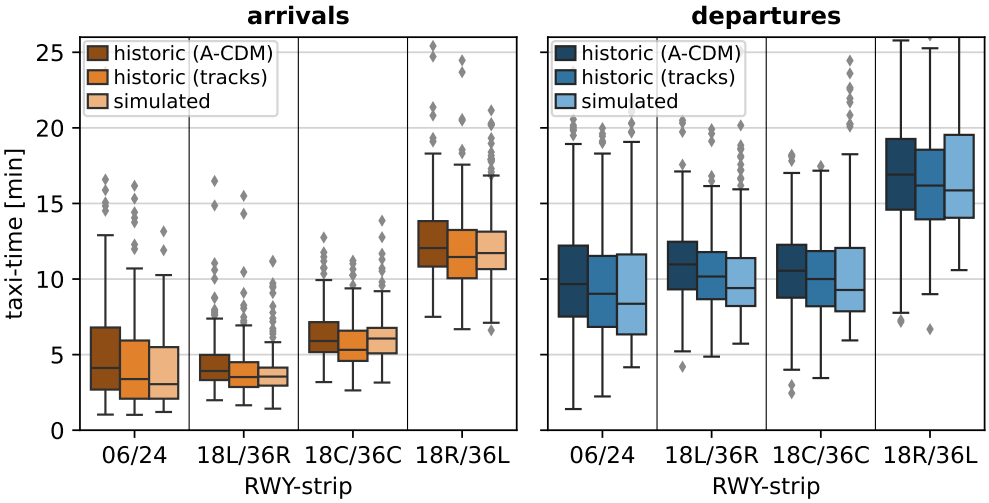


Figure 8.11: Box-and-whisker plots of taxi time distributions of historical A-CDM milestones, historical tracks, and simulated operations for arriving and departing flights per runway-strip

In general, the values from the A-CDM milestones result in longer and more spread taxi times than those based on the KPIs extracted from the historical tracks. Since the distributions of the simulated operations represent the latter more closely, we focus on the comparison between the two datasets 'historical (tracks)' and 'simulated' in the following. To assess the similarity between these two datasets, Table 8.4 lists the differences of medians as well as the Kolmogorov-Smirnov (KS) test scores for both arriving (ARR) and departing (DEP) flights per runway. Overall, the simulated operations resemble the historical taxi times well, with the KS-test scoring up to 90 %. Lower scores are reached for flights arriving at runway 18C/36C, and flights departing from 18L/36R, each showing absolute differences of 44 s between the medians of the historical and simulated taxi times.

Travelled Taxi Distances

To assess any effects due to differences in taxi distances, Figure **8.12** shows the respective distributions for the track-based historical and simulated operations, and Table 8.4 lists the corresponding similarity metrics. Although the travelled distances vary between flights from and to each of the runway-strips, the underlying distributions show a high degree of similarity, with the KS-test scoring consistently above 90 %. Therefore, any differences in travelled distances have an insignificant impact on the resulting taxi times.

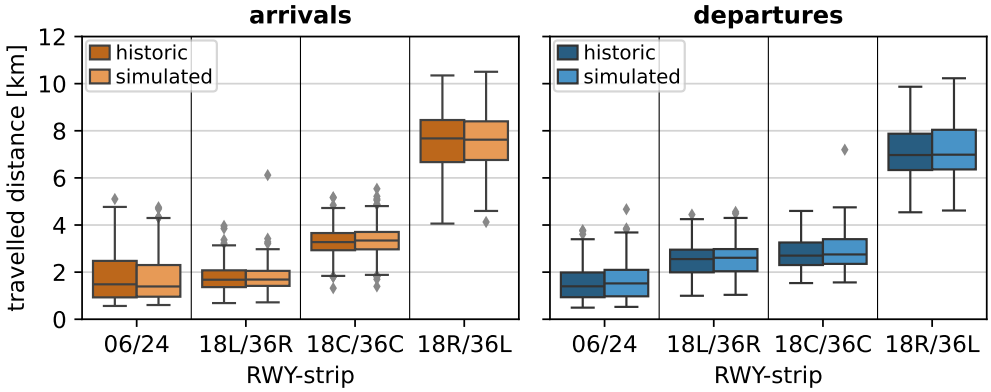


Figure 8.12: Box-and-whisker plots of taxi distance distributions of historical tracks and simulated operations for arriving and departing flights per runway

Table 8.4: Similarity between historical and simulated operations for arriving (ARR) and departing (DEP) flights, listing the difference between the median taxi times $\Delta\tilde{t}_{\text{taxi}}$ and taxi distances $\Delta\tilde{d}_{\text{taxi}}$, as well as the respective Kolmogorov-Smirnov (KS) test scores per runway-strip

		06/24	18L/36R	18C/36C	18R/36L
ARR	$\Delta\tilde{t}_{\text{taxi}}$	−19 s	0 s	44 s	14 s
	KS(t_{taxi})	91 %	89 %	78 %	87 %
	$\Delta\tilde{d}_{\text{taxi}}$	−46 m	15 m	60 m	−29 m
	KS(d_{taxi})	94 %	95 %	93 %	96 %
DEP	$\Delta\tilde{t}_{\text{taxi}}$	−38 s	−44 s	−37 s	−18 s
	KS(t_{taxi})	89 %	84 %	90 %	87 %
	$\Delta\tilde{d}_{\text{taxi}}$	129 m	60 m	66 m	14 m
	KS(d_{taxi})	93 %	95 %	92 %	97 %

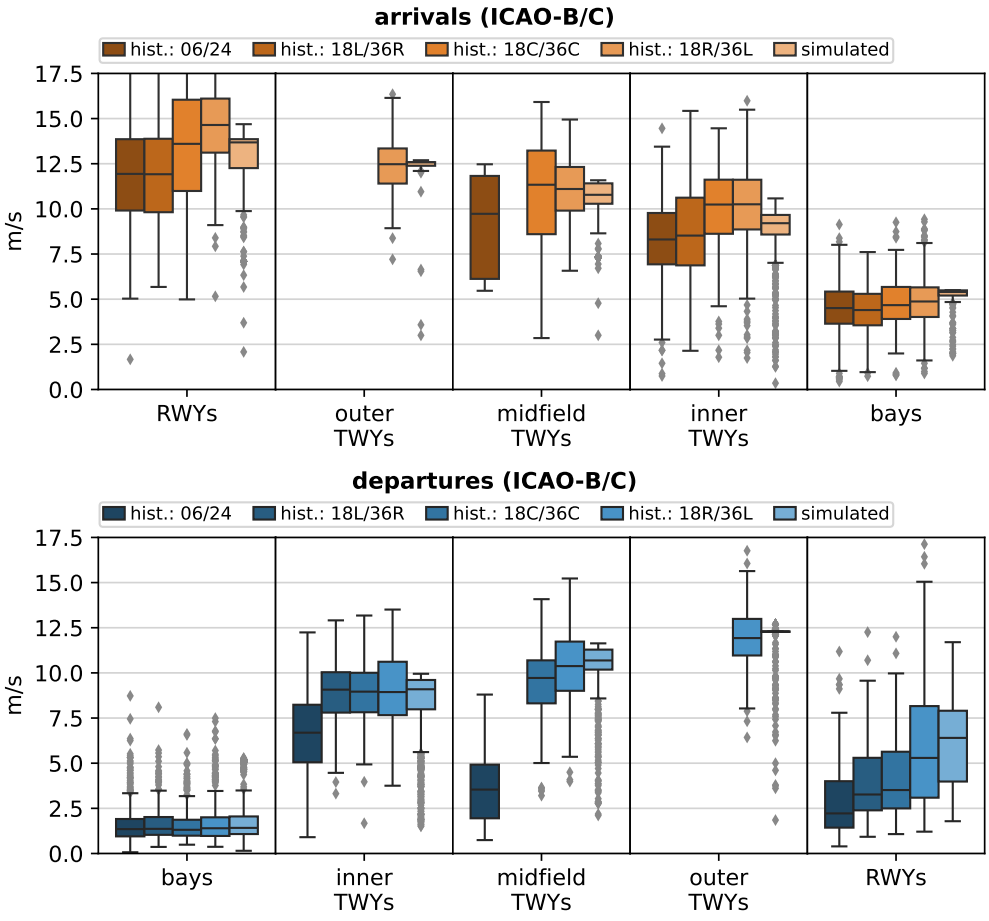


Figure 8.13: Comparison of average taxi speeds across the different taxiway areas from Figure 8.3 for both arriving (top) and departing flights (bottom)

Average Taxi Speeds across Taxiway Areas

Next, we shift to the differences in taxi speeds between historical and simulated operations. To compare the similarity of taxi speeds in each of the taxiway areas, we focus on aircraft of types ICAO-B and ICAO-C which visit Schiphol most frequently. For these, Figure 8.13 shows box-and-whisker plots of the historical speeds from and to each runway as well as the resulting taxi speed distributions of the simulated operations.

In most of the taxiway areas, the simulated operations exhibit smaller speed variances between flights than each of the respective historical speed distributions. However, flights landing on the runways 18C/36C and 18R/36L further away from the stands show significantly higher speeds along the runway exit, midfield, and inner taxiways. Along the same taxiway areas, also departing aircraft show higher average speeds when taking off from one of these two runways.

The comparison indicates that the varying degrees in taxi time similarity from Table 8.4 can be largely attributed to these historical speed variations in dependence of the assigned runway. On the other hand, the simulated operations exhibit higher speeds on runway entries than those observed in the historical tracks. This may explain the shorter taxi times of departing flights in the simulation in comparison to those observed in the historical operations. We evaluate this aspect further in Section 8.1.3.

In the bay areas, the average speeds of outbound flights are significantly lower than those of arriving flights since departing flights require additional time for pushback and engine-start. Nonetheless, the average speeds especially of outbound flights closely align between simulated and historical operations. This indicates that the times to complete the necessary procedures for outbound flights match sufficiently well, irrespective of any differences in pushback paths[1].

Inbound and Outbound Holding

In the historical operations, inbound flights stop in various places on taxiways, as visualised in Figure 8.14 (left). Most stops have a rather short duration of less than 1 min. While the longer stops usually take place on the remote holding platform (see Figure 8.14), some aircraft hold for up to 45 min on the main taxiways. Based on the conversations with ground controllers at Schiphol and confirmed through playbacks of the historical track data, such holding positions are operationally feasible due to the prevalent runway configuration that leads to unused taxiway segments, which then allow the ATCOs to let aircraft hold there. In contrast, the simulated operations in the MAS model are currently based on inbound holding taking place solely on the remote holding platform, as confirmed in Figure 8.14 (right).

During present-day taxiing, outbound flights are required to hold for various reasons: while a stop is required during a push-pull or at the end of a pushback manoeuvre, departing flights may also be required to hold for e.g. safety reasons or to wait for an assigned CTOT-slot. However, except when switching the movement direction, the reason behind stopping cannot be determined from the historical data. In contrast, trajectories in the MAS model are planned allowing aircraft only to stop when operationally required, e.g. to switch movement direction or to decouple from a tug.

[1] Recall from Figure 8.9 that some historical flights showed pushback paths deviating from the standard paths, whereas outbound flights in the MAS Model are always assigned the standard pushback paths.

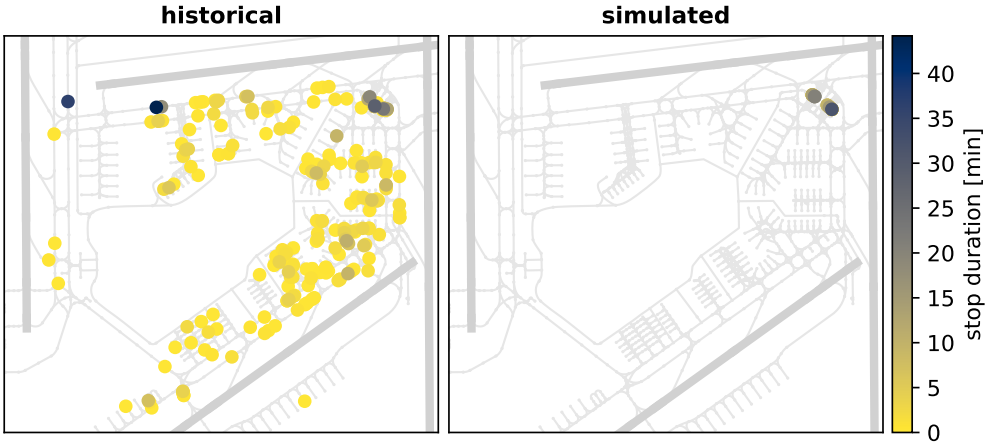


Figure 8.14: Inbound holding: stop locations and waiting durations >10 s of historical and simulated arriving flights

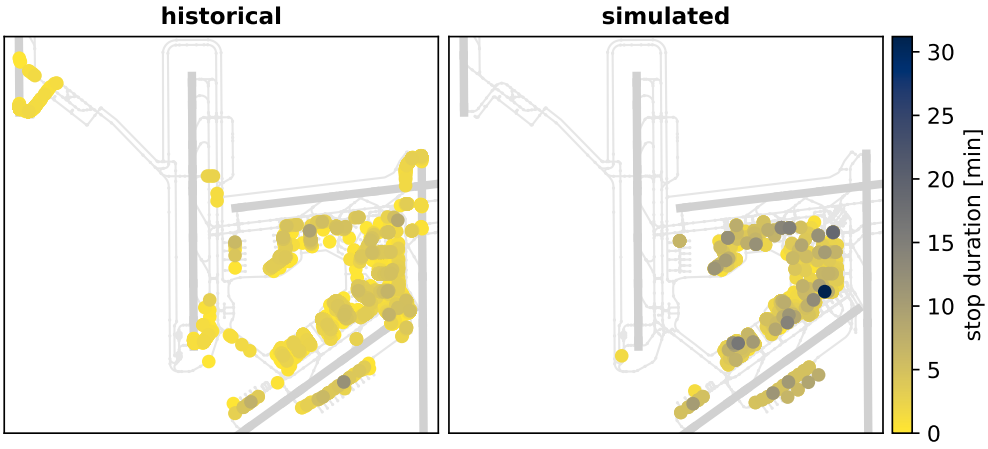


Figure 8.15: Outbound holding: stop locations and waiting durations >10 s of historical and simulated departing flights

For comparing outbound holding between the historical and simulated operations, all stopping points exceeding 10 s are visualised in Figure 8.15. While the historical tracks show that departing flights frequently hold on both the main taxiways as well as on runways and their entries, aircraft in the simulated operations only stop in the bay areas due to the pushback operations. We continue the discussion by assessing the differences in infrastructure occupancy, as outlined below.

Infrastructure Occupancy

In the following, we compare the occupancy rates on different infrastructure parts of the airport over the 48 h flight schedule data between the historical and simulated operations. Per infrastructure element, we first outline how the occupancy rates are calculated, and then draw the comparison between historical and simulated occupancy rates. To this end, Figure 8.16 visualises the occupancy rates over time for arriving and departing flights on runways, traffic on inner and bay area taxiways, as well as the occupied stands at piers, on platforms, and those used solely for cargo aircraft. All lines represent moving averages with symmetric 30 min window sizes, and are smoothed using a linear Savitzky-Golay filter.

Per time point t, the occupancy rate $occ_R(t)$ of a set of runways R is calculated as:

$$occ_{\mathrm{R}}(t) = \frac{\sum_{r \in R} \mathbf{1}_{\{\text{runway } r \text{ active and blocked at } t\}}}{\sum_{r \in R} \mathbf{1}_{\{\text{runway } r \text{ active at } t\}}} \tag{8.2}$$

where $\mathbf{1}_{\{\cdot\}}$ denotes the indicator function, returning 1 if the condition in curly brackets is true and 0 otherwise. Accordingly, only active runways contribute to the numerator and denominator, while inactive runways are excluded from the calculation. To determine whether a runway is blocked, we use the minimum separation times from Table 8.3 in combination with the general time-based separations from [176], and 60 s when no separation value is defined.

For arriving flights, the occupancy rates are identical between historical and simulated data as inbound aircraft are spawned in the MAS model at their historical landing time. Interestingly, the occupancy rate is highest during outbound peaks, i.e. when only one runway is used for landing aircraft. In comparison, runways used for departing flights show higher occupancy rates over the two operational days, reaching a maximum occupancy rate within 30 min of approximately 90 %.

The difference in occupancy rates between historical and simulated data is largest around the peaks. As observed in Figure 8.8, some historical separations are much lower than the separation values used in the simulation. Since the calculation of the runway occupancy rate is based on distinguishing binary between an occupied and a free runway, the historical rate appears to be lower. Nonetheless, this indicates that the simulated operations rather overestimate the minimum separation times. As consequence, the higher taxi speeds along runway entries of the simulated operations observed in Figure 8.13 can be attributed to the rolling takeoff procedure deployed for all departing aircraft in the MAS model. Instead of the frequent stops on both runway entries and before takeoff observed in Figure 8.15 for the historical tracks, the MAS model must lower the taxi speeds of outbound flights prior to crossing the runway stopbars to accommodate their rolling takeoff.

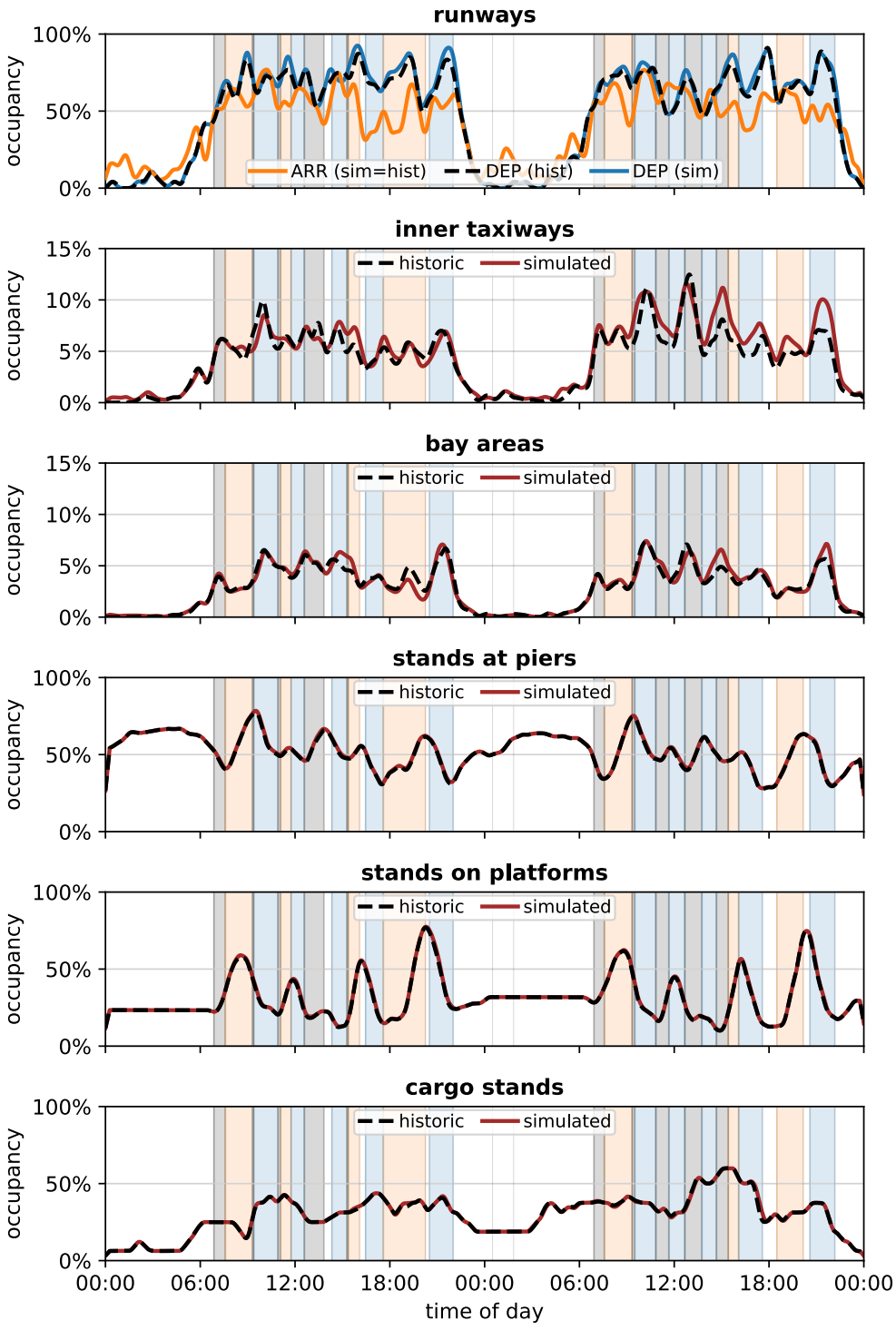


Figure 8.16: Comparison of occupancy rates in different areas of the airport between historical and simulated operations

To calculate the occupancy rate on a taxiway area over time, we approximate the relative area that is occupied by the aircraft within that area at each time point t. To this end, we define:

$$occ_{\text{twy}}(t) = \frac{l_{\text{occ}}(t)}{l_{\text{tot}}} \tag{8.3}$$

with the total length of the taxiways in the considered taxiway area l_{tot}, and the occupied part of that length $l_{\text{occ}}(t)$. To compute l_{tot}, only the segments of the core taxiway segments in the respective taxiway area are counted in. For the inner taxiways, this entails the relevant parts of TWY-A and TWY-B, without accounting for the taxiway intersections between them. For the bay areas, l_{tot} equals the sum of the taxiway segments from which stand edges branch off. We estimate the occupied part of that length $l_{\text{occ}}(t)$ per time point t using:

$$l_{\text{occ}}(t) = \sum_{a_i} 5\, r_{a_i} + 20\,\text{m}\,, \quad \forall\, a_i \text{ in taxiway area at } t \tag{8.4}$$

with the shape radius r_{a_i} of aircraft a_i. The factor 5 is used to account for the shape as well as the safety distance behind the aircraft, and 20 m are added as additional safety margin around each aircraft.

Like the runway occupancy rates, those on the inner taxiways and in the bay areas alternate similarly over time between historical and simulated operations. Most peaks fall within either one of the departure-peaks (shaded in blue) or transition periods (shaded in gray), with a clearer trend visible on the second day. On that day, the departure runways 06/24 and 18L/36R are active which are closer to the airport centre. In turn, more queuing takes place in the inner taxiway area, visible in the occupancy peaks reaching a rate $\geq 10\,\%$. While the simulation shows two additional queuing-peaks at 15:00 and 21:00, the maximum occupancy reached at approximately 13:00 remains lower than the one of the historical operations.

However, the observed occupancy difference is not reflected to the same extent between the runway occupancy rates of departing flights. Recalling that the MAS model coordinates aircraft trajectories on the basis of mitigating any stop-and-go before takeoff, the occupancy difference on the inner taxiway areas is caused by the differing deconfliction mechanisms. Instead of aircraft queues forming on the departure runways and their entries in the historical operations, the MAS model assigns slower speeds on the taxiways leading to the runways by anticipating the necessary separation time between consecutive takeoffs.

In analogy to Equation (8.2), the occupancy rate $occ_S(t)$ of a set of stands S is calculated as:

$$occ_S(t) = \frac{\sum_{s \in S} \mathbf{1}_{\{\text{stand } s \text{ accessible and occupied at } t\}}}{\sum_{s \in S} \mathbf{1}_{\{\text{stand } s \text{ accessible at } t\}}} \tag{8.5}$$

where stands inaccessible at time point t are excluded from the calculation. We define that an aircraft occupies a stand between entering the stand edge until exiting that edge, which thus includes any operational holding at the gate. If there is no historical track indicating when an aircraft departed from a stand, only the duration until standstill at the stand is used, i.e. we assume the aircraft immediately disappears, slightly underestimating the true occupancy rate.

In Figure 8.16, we distinguish three sets of stands: all cargo stands (R and S platforms), stands on all other platforms marked in Figure 8.3, and those at piers (connected stands). For each of these three sets, the occupancy rates are almost identical between simulated and historical operations. Thus, the difference in gate holding times between simulated and historical aircraft trajectories are negligible when averaged over the respective set of stands.

8.1.4 Discussion on Model Calibration and Future Work

The developed methodology to project historical track data proved effective in reconstructing realistic ground movements: most historical tracks could be projected onto the airport's taxiway graph, and many of these supported to determine more accurate A-CDM milestones. Moreover, the approach is generic and transferable to other airports: provided that a sufficiently detailed computational graph of the taxiway network exists, and the raw data meet basic quality requirements, the approach can be replicated at other airports and with alternative data sources such as ADS-B. As observed in previous research, the data quality proved poorest near terminal buildings, where the signal is occluded. Enhancing surveillance coverage and accuracy in these zones is crucial for airport operators to realise the full value of surface-movement data.

Importantly, the presented projection approach provides richer insights than conventional A-CDM event detection alone. Besides providing more precise off-block, in-block, and runway-related timestamps, it offers detailed speed distributions across different parts of the taxiway network. Such information can aid in both operational monitoring and planning processes. Furthermore, as recently explored in other research, projected trajectories can support the development of *A-CDM-lite* milestones [164], enabling airports without extensive surveillance infrastructures to derive actionable performance metrics.

Regarding the assimilation of MAS model parameters to the historical operations, the results indicate substantial potential for more realistic and type-specific speed profiles. The results also indicate that the model's reliance on standard pushback and routing procedures may underestimate the flexibility present in real operations. Allowing the MAS model to assign context-dependent pushback paths could better reflect the observed variability, particularly during taxiway or runway maintenance, as well as periods of local congestion. To this end, the availability of supplementary data, such as the precise timing of engine start-up, would allow for more accurate representations of the underlying operations. Complementing this, data-driven techniques could be used in future work to derive parameter distributions tailored to different aircraft categories within the MAS model.

Overall, the analyses in this section directly addressed feedback points 1) and 2). The close agreement between projected and simulated taxi times and distances indicates that using historical median speeds per aircraft type and taxiway area provides a suitable basis for reproducing current operations. Nevertheless, the extracted trajectories suggest that more progressive speed limits may be operationally feasible, an aspect that will be explored further in Section 8.3.

The next section turns to feedback point 3) and examines the role of aircraft towing movements, assessing how these repositioning flows interact with taxiing traffic and influence overall airside performance.

8.2 Modelling of Aircraft Towing Operations

In general, airlines are responsible for complying with the towing norms defined by the airport authority. At Schiphol, the policy document 'Regulation Aircraft Stand Allocation Schiphol' (RASAS, [178]) lays out the principles underlying the use of stands and conditions for aircraft towing. While the airport operators try to accommodate as many flights as possible at gates with connected handling, they take different actions when the scheduled demand exceeds the available capacity of connected gates. As preferred option, aircraft with long ground times are required to be towed to a remote position before towed back to a connected gate for the outbound flight. To this end, Schiphol regards long ground times as planned turn-around times exceeding 210 min for large aircraft and 170 min otherwise. As less preferred alternatives, Schiphol handles flights at remote positions, or uses outbound as well as inbound holding when necessary.

In the following, first a towing schedule is reconstructed based on the historical flight schedule. As outlined in Section 8.2.2, the MAS model is then extended with aircraft towing operations. Subsequently, the potential impact of towing movements on the surface movement operations is evaluated (Section 8.2.3). To this end, the towing volume is varied to assess its effects on hourly movements, taxiing times and distances for inbound and outbound flights, and occupancy rates.

8.2.1 Reconstructing Historical Aircraft Towing Movements

During aircraft towing movements, the transponder of aircraft is usually switched off. While that of the towing truck might be switched on, the available historical track data did not contain such tracks, rendering it impossible to extract towing movements and their timings. Nonetheless, using the aircraft registration, stand changes taking place during an aircraft's turnaround can be inferred from the historical flight schedule data. Likewise, when another aircraft used a stand in between an aircraft's arrival to and departure from the same stand, that aircraft must have been towed away in the meantime. This allows us to reconstruct the required towing movements, despite their timings and potential intermediate buffer locations remaining unknown.

In this regard, we distinguish between different tow types for towing movements from an origin stand $S_{\mathrm{o}}(\tau_{\mathrm{s}})$ to a destination stand $S_{\mathrm{d}}(\tau_{\mathrm{s}})$ with the corresponding stand types $\tau_{\mathrm{s}} \in \mathcal{T}_{\mathrm{s}} = \{\mathrm{pier}, \mathrm{platform}, \mathrm{buffer}\}$. Respectively, these mark the set of gates connected to piers, parking positions on remote (i.e. unconnected) platforms, and a buffer location subsuming any other aircraft stands such as long-term parking positions or hangars. Towing operations of aircraft moved between stands in the same bay area are likely less disruptive to the regular traffic as these do not enter the main taxiways and are relatively short. In Table 8.5, listing the reconstructed towing movements, we thus differentiate destinations of types pier and platform between being in the same or a different bay area (BA) than the origin stand.

From the 154 reconstructed towing movements, most require towing to or from a buffer location, i.e. with the actual intermediary parking location remaining unknown. Furthermore, the reconstructed dataset indicates that aircraft are usually not towed between two platforms. Overall, the reconstructed towing schedule leads to a traffic increase of approximately 5 %. However, based on conversations with ground controllers as well as a visit to Schiphol tower, it appears that more aircraft towing movements take place in present-day operations. To account for additional towing movements not captured by the reconstructed schedule, we simulate varying numbers of surplus towing movements from

Table 8.5: Aircraft towing movements reconstructed from the 48 h of historical flight schedule data

from \ to	pier		platform		buffer	∑
	same BA	diff. BA	same BA	diff. BA		
pier	4	11	8	5	48	76
platform	2	2	0	0	3	7
buffer	65		6		0	71
∑	84		19		51	154

a stand to the buffer and subsequently returning to the same stand, as further discussed in Section 8.2.3.

8.2.2 Extensions to the Multi-Agent System Model

Since the historical aircraft tracks did not show any taxiing traffic to and from the U-platform (denoted in Figure 8.3), we use it as synthetic buffer platform. Any aircraft towing movements that do not take place between two stands are towed there. The buffer platform is thus modelled to serve as both intermediate and long-term parking location, and assumed to have sufficient capacity.

In analogy to the currently defined A-CDM milestones, we use the following ones for towing movements:

- **TTDT**, Target Towing Departure Time: the planned time an aircraft should be towed away from the current stand;
- **TTAT**, Target Towing Arrival Time: the planned time a towed aircraft should arrive at its destination;
- **ATDT** and **ATAT** as the respective actual time points;
- **TTMT**, Target Towing Movement Time: as A-CDM metric with $\mathrm{TTMT} = \mathrm{TTAT} - \mathrm{TTDT}$;
- **ATMT** as the respective actual duration.

We assign the task of determining which aircraft to tow where and when to the Airport Ops Agent, and formulate two towing properties that are added to its specification defined in Section 3.2:

P5. Maintain Stand Schedule Property: the Airport Ops Agent uses the flight schedule information to monitor the time points at which flights arrive at and depart from all available stands of the airport. It updates the resulting stand schedule with the final and predicted times from the A-CDM milestones.

P6. Maintain Towing Schedule Property: based on the stand schedule of its P5, the Airport Ops Agent determines the aircraft that must be towed between stands, or to or from the buffer platform. This also includes aircraft scheduled to depart from the same stand after a long ground time which are towed to the buffer platform to free up stand capacity. Once one of such flights has arrived at its stand, the Airport Ops Agent sets the respective TTDT in accordance with the duration of the flight's turnaround processes, the time at which the stand is needed for another flight, and the estimated tow duration.

Regarding P6, we model the minimum turn-around times (MTT) of small (ICAO-B, ICAO-C) and large (ICAO-D, ICAO-E, ICAO-F) aircraft based on the characteristics of the Airbus A320 [179] and the Boeing 787 [180], respectively. However, the turn-around often takes longer than the MTT, and the underlying processes may be split into multiple parts, e.g. when the flight has a long ground time or must be towed between stands. To this end, we use the times provided in the aircraft characteristics to extract durations for splitting the overall turn-around into three sub-stages, namely the *arrival processes* such as deplaning and cargo unloading, *intermediate processes* such as refuelling, cleaning, catering, and water refilling, and *departure processes* such as boarding and cargo loading. When an aircraft is towed away after arriving at a stand, at least the arrival processes must be completed, while a tow to a stand prior to a flight's departure must leave sufficient time to complete the departure processes. Furthermore, either after arrival or before departure, sufficient time must be allocated for the intermediate processes. As a consequence, the Airport Ops Agent uses the parameters and associated minimum durations listed in Table 8.6 when executing its P6.

Table 8.6: Minimum turn-around times in [min] to determine the towing schedule

parameter	ICAO-B/C	ICAO-D/E/F
minimum turn-around time (MTT) t_{MTT}	45	60
MTT for arrival processes $t_{\text{MTT,arr}}$	10	20
MTT for intermediate processes $t_{\text{MTT,mid}}$	20	40
MTT for departure processes $t_{\text{MTT,dep}}$	15	30
buffer-time for turn-around processes t_{buf}	5	

Like the flight schedule, the Airport Ops Agent shares the towing schedule with the Routing Agent. The latter uses the schedule information to calculate and coordinate the trajectories of all agents moving concurrently over the airport surface. To this end, towing movements are modelled to follow the standard pushback path of the respective stand according to the airport provisions (see [153], disregarding pull manoeuvres).

In general, towing movements must follow the standard taxiway flow directions to the designated destination, akin to present-day towing operations. Furthermore, they must give way to taxiing aircraft, except otherwise instructed by ATC. This is reflected in the path planning conducted by the Routing Agent by assigning a priority-factor of $p_{tow} = 0.1$ to towing movements, i.e. any delay of a taxiing aircraft is equal to a tenfold delay of a towing movement, leading to a de-prioritisation of the trajectories of towing movements. Based on interviews with operational experts, a maximum tug speed of 4.5 m/s is used in general. In one experiment, the higher maximum speed of 11.8 m/s corresponding to the reference TaxiBot system [12] is used, as further outlined in the next subsection.

8.2.3 Evaluation of Aircraft Towing Operations

With the MAS model extended by aircraft towing operations, we conduct multiple simulations with varying number of towing movements to evaluate their impact on regular taxiing aircraft. In the following, we first provide an overview of the conducted towing simulations as well as the resulting hourly movements and required parking spaces at the buffer location. Subsequently, we analyse the impact on the taxi times and distances of regular traffic, and on the infrastructure occupancy rates.

Overview of Conducted Towing Simulations

An overview of the conducted simulations is provided in Table 8.7, listing the number of towing movements as well as the disregarded tows over the 48 h flight schedule. The simulation 'no towing' corresponds to the simulated operations discussed in Section 8.1.3. 'Stand changes' refers to simulating only the operationally necessary towing movements, i.e. those listed in Table 8.5. Next to the reconstructed towing schedule, aircraft with long turnaround durations could be towed to the buffer and back to the same stand to free up stand capacity in the meantime. We simulate such additional towing movements by adjusting the time limit at stands which the Airport Ops Agent uses to determine potential aircraft to be towed away, assuming that these take place only during daytime, i.e. between 06:00 h and 23:00 h.

Table 8.7: Overview of performed simulations to assess the impact of aircraft towing movements

simulation-name	max. tug speed [m/s]	towing movements	disregarded tows
no towing	n/a	0	0
stand changes	4.5	154	0
4x MTT	4.5	236	0
3x MTT	4.5	348	0
2x MTT	4.5	499	126
1.5x MTT	4.5	510	422
1.5x MTT, fast	11.8	905	226

For relatively short time limits at stands, the Airport Ops Agent might estimate that the required towing duration to move an aircraft to and from the buffer location, modelled to be their intermediate parking positions in the simulations, exceeds the estimated turn-around time. Such towing movements are thus disregarded, as listed in the respective column in Table 8.7. In the simulation '1.5x MTT', the Airport Ops Agent disregards around 45 % of towing movements due to the low yet typical speed limit of 4.5 m/s. To further increase towing traffic, a speed limit of 11.8 m/s is used in the simulation '1.5x MTT, fast', assuming that the towing movements are carried out with operations corresponding to the TaxiBot concept [12].

In general, we assume that sufficient tugs are available to carry out all planned tows. In turn, towing movements incur potential delays in relation to their TTDT only due to the necessary coordination with regular taxiing traffic.

Hourly Movements and Aircraft Parked on Buffer Platform

Selecting simulation '1.5x MTT, fast' as example, Figure 8.17 shows the hourly count of inbound, outbound, and towing movements over the two simulated days. While varying over time, the simulated towing operations result in well over 150 total hourly movements at multiple times throughout the 48 h, with up to 55 aircraft being towed hourly during daytime. The peaks of the towing movements occur when the regular traffic shifts from an arrival-peak to a departure-peak. Likewise, minimum towing movements occur in the opposite case, with least tows recorded at 18:00 on the second day due to the wider gap between the departure-peak and arrival-peak.

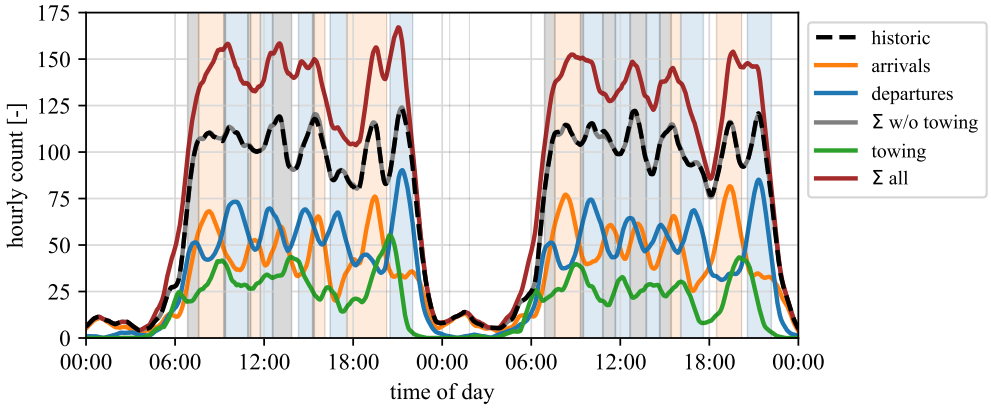


Figure 8.17: Hourly ground movements over the simulated 48 h. Shades denote the runway configuration phase: off-peak (white), arrival-peak (orange), transition (grey), and departure-peak (blue).

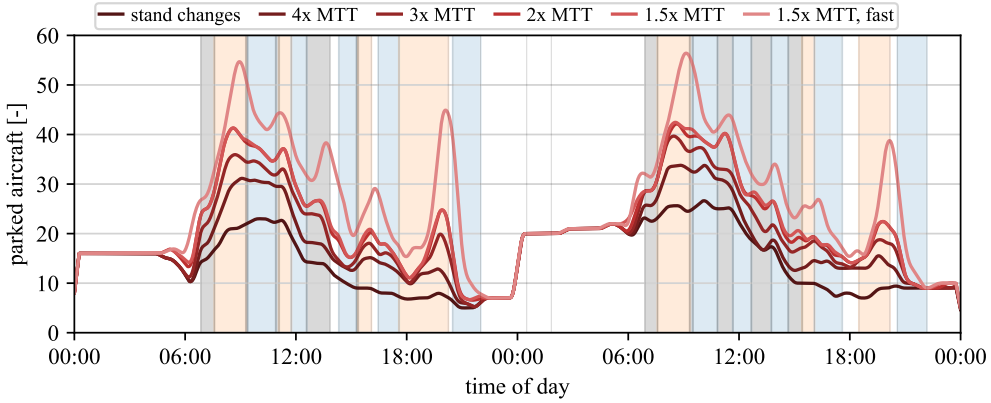


Figure 8.18: Required parking stands at buffer platform over time

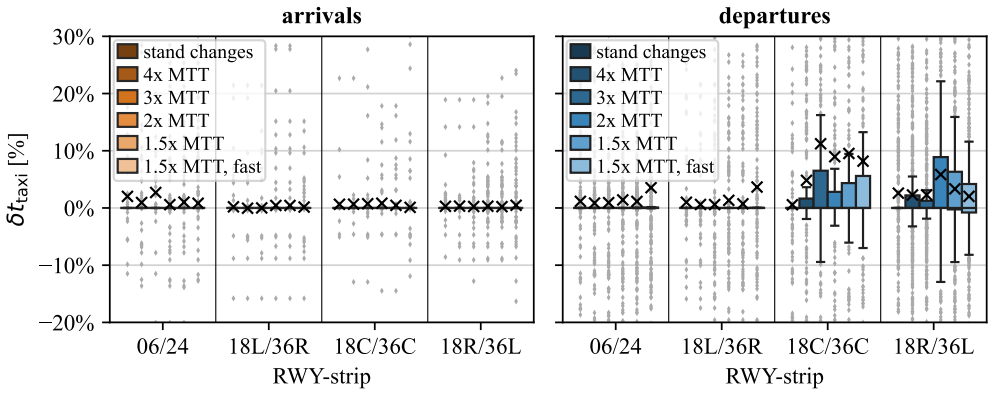


Figure 8.19: Normalised taxi time differences δt_{taxi} in relation to simulation 'no towing'; black cross denotes the mean value

Figure 8.18 shows the required parking stands at the buffer platform over time. In the simulation 'stand changes', up to 27 aircraft are parked at buffer. For lower limits of the stand occupancy times, the necessary parking spaces at the buffer platform increase, with up to 57 required in the simulation '1.5x MTT, fast'.

As already noted in Section 8.2.1, most aircraft are towed to or from the buffer location. To this end, towing movements mostly utilise the inner and bay area taxiways, with some also travelling through the midfield taxiway area. Thus, they might affect the taxi times of regular flights, which we examine in the next subsection.

Impact on Taxi Times and Distances

Figure 8.19 shows the distribution of taxi times as box-and-whisker plot per runway for both arriving and departing flights, normalised with those of from the simulation 'no towing'. The towing operations have a negligible impact on the median taxi times of both arriving and departing flights. However, the average taxi times marked by black crosses show that the trajectories of certain departing flights are affected. The strongest increase in average taxi time of up to 11 % occurs for flights departing from runway 18C/36C, while those departing from other runways remain below 6 %. The trajectories of some individual flights are strongly affected, indicated by the outliers that are marked by grey dots in Figure 8.19. In contrast, the effects on the taxi distances remained very low, with outliers showing absolute values below 3 %.

Impact on Infrastructure Occupancies

With a limited impact on the taxi times of departing aircraft, the occupancy rate of the runways remains unaffected. In contrast, as shown in Figure 8.20 for all simulations from Table 8.7, the rates on the inner taxiways, bay areas, stands at piers, and stands on platforms are affected differently.

In comparison to the simulation 'no towing', the occupancy rates increase on taxiways and bay areas for larger numbers of towing movements, reaching up to 15 % and 9 %, respectively. At the same time, the occupancy rates of stands both at piers and on platforms reduce significantly, with occupancy rates mostly remaining below 60 % during daytime. Since the maximum stand occupancy time was assumed not to apply to cargo stands, their occupancy rate did not change in comparison to Figure 8.16.

The observations apply more strongly to fast towing movements ('1.5x MTT, fast'), with stand occupancy rates mostly below 50 % during daytime. Interestingly, in contrast to 'no towing', the maximum occupancy rate of connected stands is reached during nighttime. Moreover, the occupancy rate varies less throughout the day.

8.2.4 Discussion on Towing Movements and Future Work

Overall, the simulations show that even with a substantial volume of towing movements, their influence on taxiing traffic remains limited. This effect is smaller than anticipated based on discussions with ATCOs and operational experts (feedback point 3), suggesting that towing flows at Schiphol may be less disruptive at the system level than commonly assumed.

At the same time, the results indicate clear operational benefits: towing additional aircraft to intermediate parking positions can free substantial capacity of connected stands.

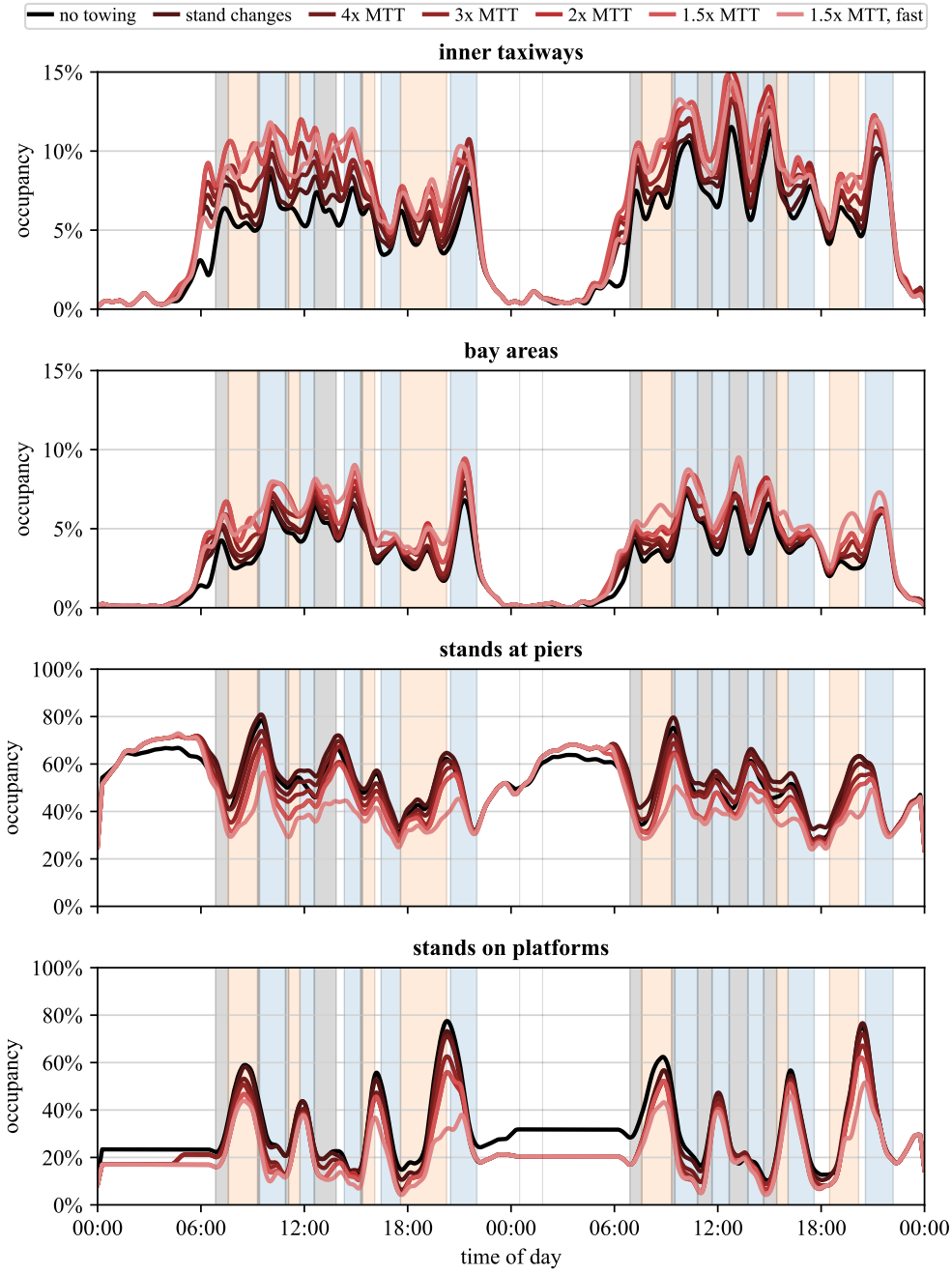


Figure 8.20: Comparison of occupancy rates in different areas of the airport across the simulated towing operations from Table 8.7

Moreover, alternative towing strategies, such as repositioning aircraft to unoccupied stands on a nearby platform, could further shorten towing durations relative to the scenarios modelled here.

Several limitations highlight the need for further research. First, the simulations assumed uninterrupted towing operations with fully available resources. How towing delays, resource shortages, or mid-movement disruptions propagate into taxiing performance remains uncertain and requires further investigation. Second, applying the analysis to other major airports, which have different taxiway structures, traffic mixes, and towing regulations, will be essential to assess how far these insights extend to other airport environments and operational constraints. Only then can the operational advantages of increased towing operations be weighed against the practical challenges encountered in real-world ASM Ops.

8.3 Evaluating Operational Model Variations

In the final part of this chapter, we evaluate the impact on performance that operational model variations pose. First, an overview of the simulated model variations is provided in Section 8.3.1. We then analyse the impact on the taxi times and distances (Section 8.3.2), and on the infrastructure occupancy rates (Section 8.3.3).

8.3.1 Overview of Simulated Model Variations

Table 8.8 lists the different model variations that are compared in the subsequent sections. Since Section 8.2 did not show a significant influence of towing operations, we use the same model as evaluated in Section 8.1.3 as baseline to which the other model variations will be compared to.

Table 8.8: Overview of simulated model variations

model name	speed limits	takeoff separation scheme
baseline	$\tilde{v}_{\text{hist}}$	ICAO-WTC
vMed+RCT	$\tilde{v}_{\text{hist}}$	RECAT-EU
vMax+WTC	v_{max}	ICAO-WTC
vMax+RCT	v_{max}	RECAT-EU

Referred to as 'vMed+RCT', we exchange the minimum takeoff separation based on ICAO-WTC with values listed in Table 8.3 with the minimum wake turbulence separations following the RECAT-EU scheme[2] [133]. Using the historical separations from the 48 h flight schedule, Figure 8.21 shows the distributions per combination of the wake turbulence categories of RECAT-EU. The red lines denote the respective minimum time separations. Interestingly, multiple consecutive takeoffs had smaller temporal separations than 60 s, indicating that ATCOs might allow shorter separation times in real-world operations than the minimum assumed here.

[2]Note that the RECAT-EU categories do not align with the ICAO aircraft types. As a rule of thumb, ICAO-B ↦ CAT-E, ICAO-C ↦ CAT-D, ICAO-D ↦ CAT-C, ICAO-E ↦ CAT-B, ICAO-F ↦ CAT-A, despite not being a one-to-one match.

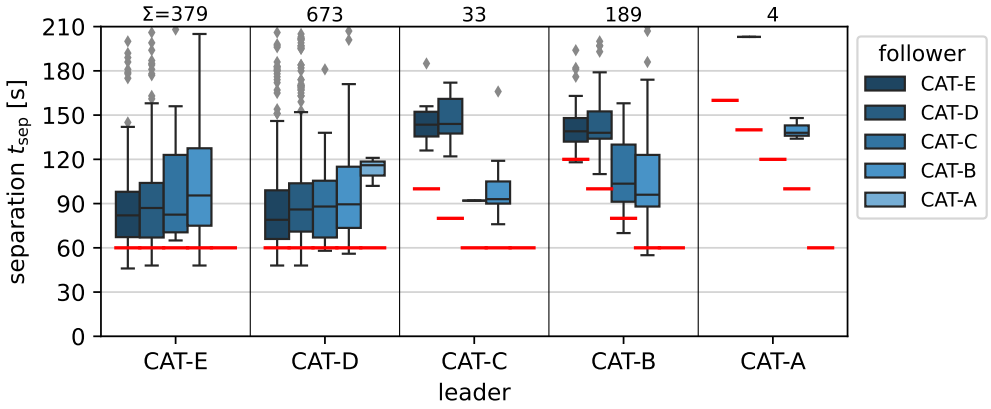


Figure 8.21: Temporal separation of consecutive takeoffs in historical data, grouped by RECAT-EU categories [133]; the minimum separation times for the respective RECAT-EU combinations of consecutive takeoffs is indicated by the red lines

For model variations using this separation scheme, we assume that the standard instrument departure (SID) routes are optimised so that aircraft taking off consecutively are always able to follow diverging SID routes. As a consequence, an additional takeoff separation is only required for intersection takeoffs, for which we use 1 min akin to the ICAO-WTC scheme.

To assess the impact of the speed limits, we use progressive limits in the model variations 'vMax+WTC' and 'vMax+RCT'. The progressive speed limits follow the maximum values defined in Table 8.1 for the different taxiway areas marked in Figure 8.3. To calculate respective speed limits along curved taxiway segments, the maximum lateral accelerations $a_{\text{lat,max}}$ from Table 8.2 are used. Note that the model 'vMax+RCT' is closest to that analysed in Chapter 6, which showed potential taxi time improvements of up to 30 %. However, that model did not use calibrated speed limits and A-CDM milestones, and did not assign the runway entries and exits based on the historical track data.

8.3.2 Impact on Taxi Times and Distances

Across all model variations, the effects on taxi distances remained insignificant. In contrast, the model variations affect the taxi times in different ways, as discussed in the following. To this end, Figure 8.22 shows the taxi time distributions per runway-strip for each of the model variants.

In comparison to the baseline model, using the RECAT-EU scheme does not have an impact on arriving flights. For departing flights, the average taxi times reduce by approximately 10 %. Furthermore, the taxi times vary notably less, as indicated by the decrease of the interquartile range marked by the boxes in Figure 8.22.

Both models with progressive speed limits showed coherent reductions in taxi times of arriving flights by approximately 25 %. When combined with ICAO-WTC (model 'vMax+WTC'), similar taxi time reductions of departing flights are achieved as the shift from ICAO-WTC to RECAT-EU under unchanged speed limits (model 'vMed+RCT'). However, the impact of increasing the speed limits is highest for runway 18R/36L due to its far distance from Schiphol's centre, with average taxi times decreasing by 18 %.

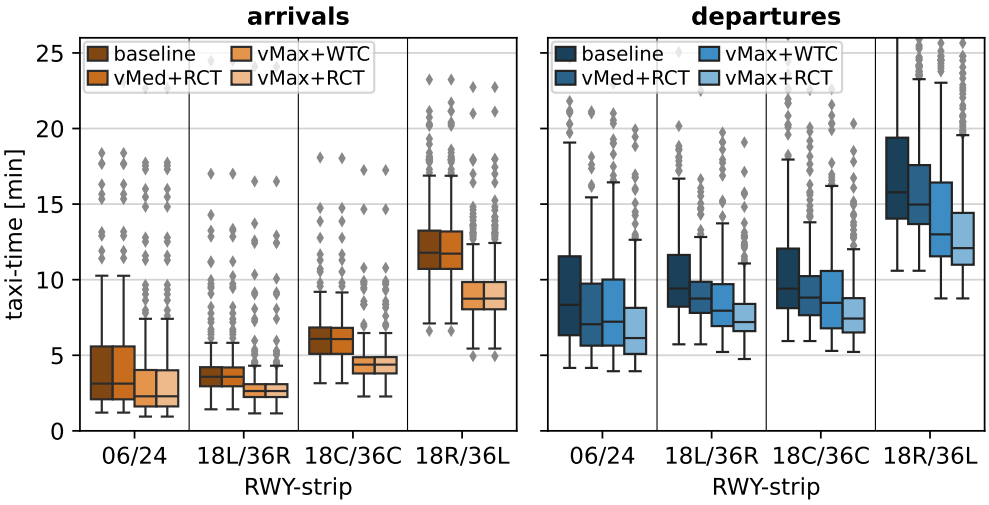


Figure 8.22: Box-and-whisker plots of taxi time distributions for arriving and departing flights per runway-strip

Combining the RECAT-EU scheme with progressive speed limits (model 'vMax+RCT') shows the largest benefits. In comparison to the baseline model, the average taxi time reduces by 20 % to 25 % across the different runways. Like 'vMed+RCT', the taxi times also vary less between flights.

8.3.3 Impact on Infrastructure Occupancies

Figure 8.23 shows the occupancy rates over time for departing flights on runways, traffic on inner and bay area taxiways, as well as the combination of all stands. Like Figure 8.16, all lines represent moving averages with symmetric 30 min window sizes, and are smoothed using a linear Savitzky-Golay filter. In the following, we discuss the impact of the model variations with respect to the baseline model, which was compared to the historical operations in Section 8.1.3.

In line with expectations, when shifting from the ICAO-WTC to the RECAT-EU separation scheme, the required minimum time separation reduces from up to 90 % to 70 %, and thus decreases the occupancy rates of the runways. In contrast, changing the speed limits has a negligible impact on the occupancy rates. The runway throughput is thus primarily determined by the necessary separation between takeoffs, while the required taxi time to arrive at runways does not play an important role.

In contrast, the largest effects on the occupancy rate of the inner taxiways is visible for the model variants with progressive speed limits. In the model 'vMax+RCT', the inner taxiways are generally less occupied throughout the 48 h simulated operations than the baseline model. Interestingly, the occupancy rate reduces especially on the second day, and remains thus closer to that of the first day, with maximum values remaining below 8 %. While switching the takeoff separation scheme has a negligible effect on the occupancy rate of the first day, especially the peaks visible during the second day are reduced in the models using the RECAT-EU scheme with respect to those that use the ICAO-WTC scheme. In comparison, the occupancy rates of the bay areas and the stands show only small and

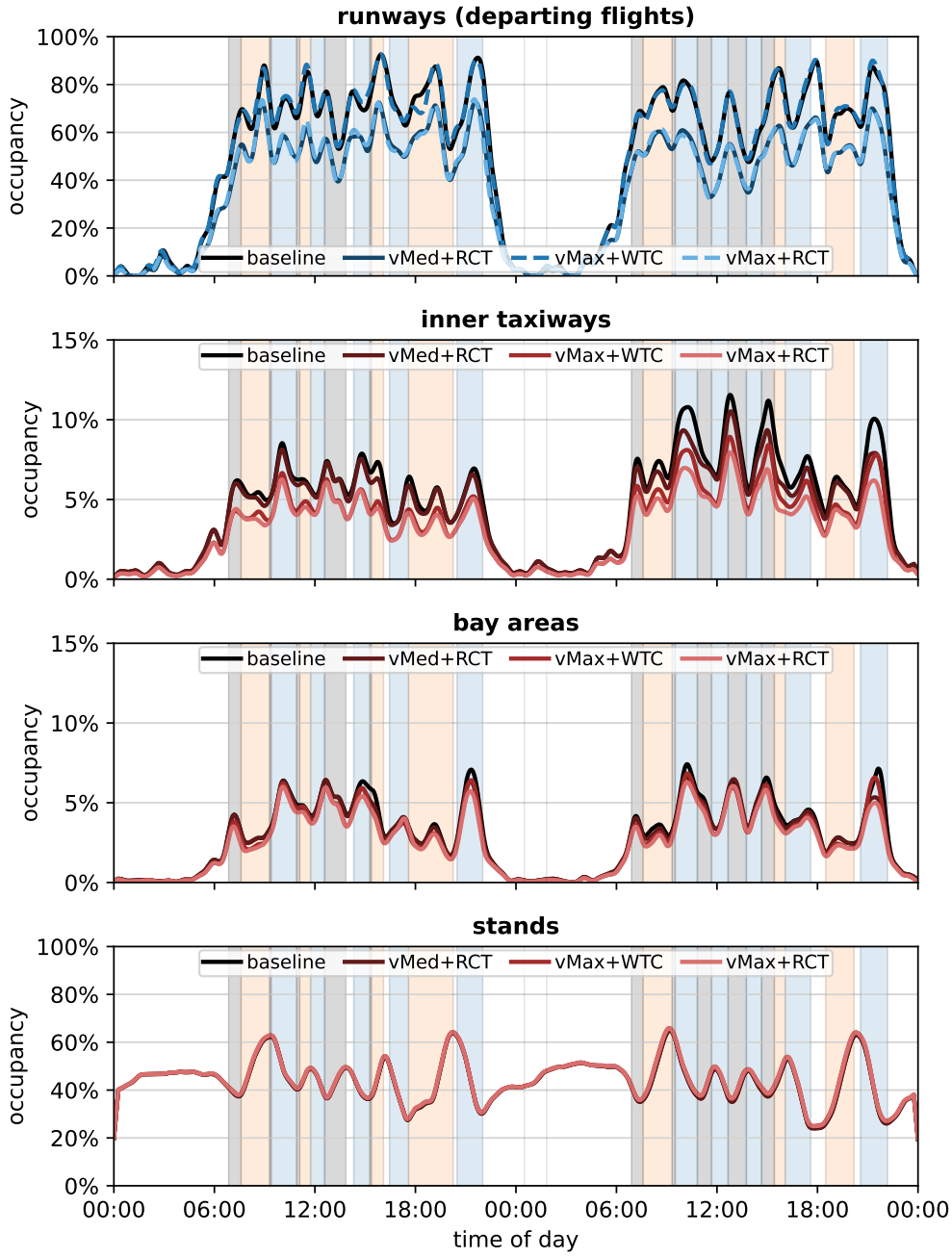


Figure 8.23: Comparison of occupancy rates in different areas of the airport across the simulated model variations

negligible changes, respectively, across the simulated model variations. This indicates that improvements to both the takeoff separations as well as the average taxi speeds lower the congestion of the main taxiways, while the use of stand holding to deconflict trajectories remains unaffected.

8.3.4 Discussion on Model Variations and Future Work

The results indicate that reducing minimum takeoff separations has a pronounced effect on runway occupancy rates and, consequently, on the upstream queues on the connected taxiways. At airports where stands are located close to the active runway, such improvements could substantially mitigate local congestion and stabilise taxi-out flows. These findings underline the strong coupling between departure-sequencing policies and surface-movement performance, suggesting that even modest adjustments to separation rules can lead to system-wide benefits.

The analysis also shows that enabling aircraft to operate closer to the feasible speed limits of different taxiway areas offers further performance gains. Supporting pilots through tailored 4D trajectory guidance or real-time speed advisories could help achieve these higher, yet operationally realistic, speeds. Nevertheless, any form of advisory or trajectory-based guidance must be designed to preserve safety margins under varying visibility, surface conditions, and traffic densities, highlighting the need for operational trials and human-in-the-loop assessments. Moreover, future research should explore how data-driven techniques could help to provide predictive digital-twin support to both pilots and ATCOs.

Taken together, the variations analysed here illustrate that substantial efficiency gains on the airport surface may be attainable through relatively targeted adjustments. However, translating these results into operational practice requires careful evaluation of safety constraints, human factors, and implementation feasibility.

8.4 Conclusion

This chapter calibrated and extended the multi-agent system model for airport surface movement operations defined in Chapter 3, addressing key realism gaps identified by operational experts. Using Amsterdam Airport Schiphol as a case study, we projected historical trajectories to calibrate aircraft dynamics, incorporated towing movements, and examined selected operational variations related to takeoff separations and taxi speed limits.

The findings highlight that runway processes remain the dominant constraint on airside operations at Amsterdam Airport Schiphol: even with reduced takeoff separations, runway occupancy rates stay high relative to those on taxiways, bay areas, and stands. In contrast, connected stand capacity can be increased effectively through additional towing. Notably, under the simulated centralised coordination scheme, even large volumes of towing movements did not degrade taxiing performance. The projected trajectories further indicate that higher taxi speeds are achievable in many areas of the airport, and both progressive speed limits and improved takeoff separations showed performance gains of around 10 % each.

Collectively, the results demonstrate how combining detailed trajectory analysis with a high-fidelity MAS model can reveal where operational bottlenecks arise, which adjustments offer meaningful efficiency gains, and where common assumptions may overstate system-level impacts. Although the study focused on Schiphol, the insights regarding runway dominance, towing effectiveness, and realistic speed profiles are relevant to other major airports seeking to enhance the efficiency and predictability of their airside operations.

> This chapter concludes the validation phase of the thesis (Chapters 6 to 8) by addressing RQ 6 through model calibration based on historical trajectory data and RQ 7 through the explicit representation of towing operations. The next chapter turns to the exploration phase by exploring human–automation interactions. Building on the validated capabilities of the MAS model developed throughout the thesis, it explores how interaction mechanisms and interface concepts can be embedded from the *technical* point of view into the MAS model to enable operator involvement. As such, these demonstrations are intended to establish technical feasibility and to outline directions for future human-in-the-loop evaluation, rather than to provide validated evidence of human performance effects.

9

Towards Human–Autonomy Teaming

EASA and SESAR JU have published complementary roadmaps for a transition towards an increasingly automated (and, where appropriate, AI-enabled) air traffic management system. Their long-term vision of advanced automation aligns with the work reported in the preceding chapters, which investigated multiple operational concepts for fully-automated airport surface movement operations (ASM Ops) using the multi-agent system model introduced in Chapter 3. Real-world deployment, however, requires such automation to be embedded in a socio-technical context in which human operators can interact with the system effectively.

In the context of the SESAR JU project 'ASTAIR – Auto-Steer Taxi at Airport', workshops and interviews with operational stakeholders were conducted to elicit operational requirements and preferred interaction patterns for automation-supported surface operations. Building on these inputs, this chapter explores how such requirements can be represented and implemented within the MAS model to enable effective human–automation interactions. In the first part, a conceptual agent-based framework is presented in which a set of interactive tools allows human operators to query, inspect, and influence the automated planning process. Selected interaction use cases – derived from the stakeholder workshops and historical data – are then demonstrated through the lens of a fictitious air traffic controller at Amsterdam Airport Schiphol. The resulting demonstrations provide an implementation-oriented basis for discussing broader considerations of human–automation interactions in next-generation ASM Ops.

This chapter is based on the following publication:

M. von der Burg and A. Sharpanskykh, 'Exploring human-automation interactions in next-generation airport surface movement operations: An agent-based modelling approach', in *SESAR Innovation Days 2024*, Rome, Italy, 2024, pp. 1–9. DOI: 10.61009/SID.2024.1.11.

The air transport industry faces the dual challenge of rising passenger demand and the requirement to reach net-zero emissions by 2050 [161]. At large airports that play a vital role in the air transport system, these challenges are exacerbated as infrastructural expansions are insufficient to facilitate the predicted growth [8]. Rising airport congestion may reduce the predictability of an aircraft's taxi time, i.e. the duration an aircraft spends travelling over the airport surface between the runway and its stand. This may impact not only the individual flight but may also cause knock-on effects on the network [10]. Moreover, the workload for Air Traffic Control Officers (ATCOs) may increase [9], which can in turn reduce the operational efficiency.

Increasing the level of automation is seen as one viable solution: towards 2050 and beyond, both EASA and SESAR JU define a vision and roadmap to eventually reach highly-automated air traffic management, i.e. Level 3 in EASA's AI roadmap [18], or Level 4–5 in the roadmap of SESAR JU [145]. Previous SESAR projects investigated how to increase the efficiency and predictability of taxiing operations through automation [139], [140]. Furthermore, aircraft manufacturers are developing technology for allowing airplanes to taxi autonomously [181].

In the preceding chapters, a detailed, computational multi-agent system (MAS) model for fully-automated airport surface movement operations (ASM Ops) was created. The model and its underlying path planning algorithm were implemented and tested in Python (Chapters 3 to 5). Furthermore, the operational consequences for different concepts of fully-automated ASM Ops were assessed (Chapters 6 to 8). However, in real-world implementations of automated systems, humans will nonetheless play a crucial role: the automation needs to be maintained and extended by developers, and human operators must oversee that it continues to function as expected and intervene if necessary. Human operators and automated systems must thus interact with each other seamlessly and effectively, motivating EASA' Level 2 which is centred on human–autonomy teaming (HAT).

HAT can be viewed as a specific class of human–automation *interaction* (HAI), comprising the observable exchanges through which operators and automated systems share information and authority. As such, HAI provide the foundational mechanisms upon which human–autonomy teaming can be built. Teaming, however, is not implied by interaction alone – it emerges only when these mechanisms support higher-order properties such as shared goals, mutual predictability, and coordinated decision-making [182], [183]. In the context of ASM Ops, this interaction layer, i.e. how controllers access, interpret, and influence automated planning, is particularly critical: effective teaming presupposes interaction mechanisms that enable transparency, predictability, and appropriate allocation of authority between human and automated agents [23], [24].

In comparison to current ASM Ops, the tasks of human operators will change: the horizon of decision-making moves towards the tactical and strategic levels, as the operational level is managed foremost by automation, like sending 4D trajectories and clearances automatically to (auto-)pilots. The interaction must be designed to reduce the cognitive load on human operators by automating routine tasks and providing decision support. However, it must also ensure that operators remain engaged and informed to prevent skill degradation and ensure quick, effective intervention when necessary [31], [184].

The SESAR JU project 'ASTAIR – Auto Steer Taxi at AIRport' aims to develop interactive tools and adaptive AI algorithms to enhance the efficiency, safety, and sustainability of

ASM Ops [35]. In the project's context, workshops and interviews with human operators from three different airports (IATA: AMS, CDG, FRA; ICAO: EHAM, LFPG, EDDF) were carried out to determine requirements and desirable interactions between humans and automated systems within ASM Ops. Despite the different characteristics of the taxiway networks of the respective airports, the experts recognised the benefits that higher levels of automation (Levels 2–3) would bring for their operations, some of which being:

- to increase the shared situation awareness [185] between multiple human operators as well as their interactions with automated systems, e.g. by providing better estimates of the engine-start times, or by keeping track of as well as managing clearances;
- to optimise e.g. the selection of a suitable holding point, the time point to cross an active runway, or the pushback procedures;
- to improve safety, e.g. by prohibiting accidents between service vehicles and aircraft at intersections.

Based on the discussions, regular occurring and characteristic situations were synthesised into use cases that are further studied in the project [63].

This chapter explores how the findings from the stakeholder workshops and interviews can be embedded into the automated multi-agent system and its path planning algorithms. Importantly, and consistent with the thesis-wide scope, the explored cases aim to illustrate their technical implementation into the MAS model, rather than to provide validated evidence from human-in-the-loop studies. Moreover, although human–automation teaming is acknowledged as a motivation and long-term goal, it constitutes a research area in its own right and therefore lies beyond the scope of the explorations presented in this chapter.

In the following, the conceptual agent-based framework to enable human–automation interactions is introduced, and the related interactive tools are outlined that allow human operators to examine and manipulate the planned paths (Section 9.1). In Section 9.2, different exemplary interactions between ATCOs and the MAS are presented, motivated by the workshop results. After introducing each case, the interaction is described in terms of how it was modelled and implemented in the multi-agent system context, followed by a discussion of its broader applicability within ASM Ops.

The cases are based on the 48 h of historical data that was analysed and used in the studies of the preceding chapters. The interactions were embedded into the MAS model and executed in scripted simulations by injecting operator inputs at predefined time steps during runtime, rather than disclosing them to the agents in advance. This enabled to verify that the MAS model can accommodate such interventions and respond effectively to ensure operational efficiency and safety. Under the assessed cases and simulation conditions, no closed-loop instabilities were observed.

Nonetheless, the presented cases remain exploratory. Many issues such as the aforementioned instabilities, appropriate scope of shared information in accordance with the urgency of a given situation, and runtime limitations of the decision-support algorithms can only be investigated in dedicated human-in-the-loop experiments. Given the breadth of open questions, only selected directions for future work are motivated at the end of this chapter.

9.1 Agent-based Model of Human–Automation Interactions

This section outlines the schematic agent-based framework of human–automation interactions in the context of ASM Ops, which is visualised in Figure 9.1. As main operator roles within ASM Ops, the **ATCO** and the **Pilot** are shown on the left-hand side in the figure. These human operators communicate with others, e.g. ground personnel and airline staff, to maintain shared situation awareness and to make time-critical decisions. While the ATCOs act as supervisors in the system, the pilots carry out the commands provided by the **MAS** from the preceding chapters of this thesis. The MAS automatically receives the position and speed of all surface movements, and sends the planned, conflict-free 4D trajectories to the flight decks. To carry out the path planning, the MAS requires data input such as the flight schedule (FS) and runway configuration (RMO) from the ATCOs, and provides informative data like the estimated taxi time to them. To separate the MAS-internal functionalities from the interactive elements, we introduce the **Interaction Agent** that handles the interaction between human operators and the MAS. The roles and tasks as well as the interactive tools are further outlined in the following.

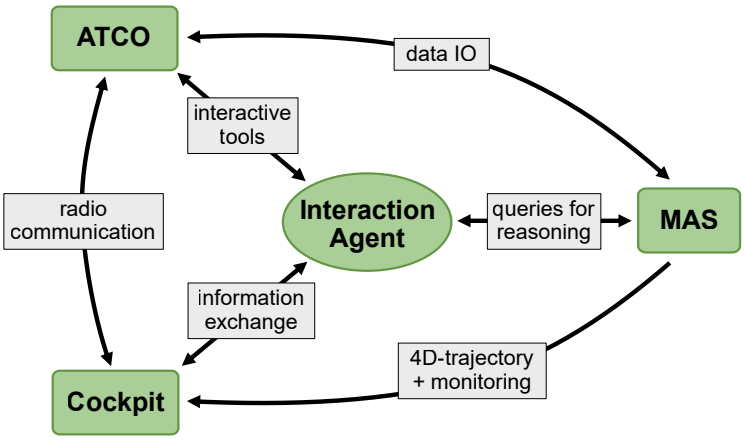


Figure 9.1: Schematic agent-based framework of human–automation interactions for airport surface movement operations

9.1.1 Role of the Human Operators

In general, *human operator* refers to any person acting within the ASM Ops, such as ATCOs, pilots, airline staff, or ground personnel. While human operators interact among each other, they can obtain relevant information from the MAS model, and interact with the Interaction Agent to get an overview of the upcoming traffic situation, update data inputs and predictions, or seek decision-support. The ATCOs interact most extensively and with varying degree of complexity with the Interaction Agent to fulfil their responsibility of ensuring safe and efficient ASM Ops. Accordingly, this chapter focuses primarily on their interactions with the MAS model.

In contrast, the (auto-)pilots primarily carry out the commands given by the MAS. While pilots will likely remain in control of the aircraft during taxiing in the mid-term, auto-pilot systems may eventually take over the nominal control of the aircraft to follow the 4D trajectories, which the MAS sends via an appropriate datalink system to the flight decks. In general, the pilots can obtain any relevant information by interacting with the Interaction Agent. Likewise, they can update aircraft-related data like the required engine-start time, which the MAS then accounts for when planning the aircraft's trajectory.

The pilots shall contact the ATCOs via radio for urgent matters, and can also pose non-critical requests. Allowing for such direct communication between ATCOs and pilots within the interaction framework does not only ensure quick response times in urgent situations, but was also seen as providing job satisfaction in interviews with operational experts. In the long term, the non-critical human-to-human communication could shift further towards human–automation interaction, provided that the deployed AI system will have gained the required additional capabilities to process such context-specific requests.

9.1.2 Role of the MAS

Based on the A-CDM milestones as well as the data provided by the ATCOs, the MAS calculates conflict-free 4D trajectories for all surface movements using the routing algorithm outlined in Chapter 4. The multi-agent motion planning algorithm sets priorities between the aircraft to deconflict their concurrent routes, and accounts for their shapes and kinematic properties. The 4D trajectories are sent to the flight decks, and are automatically updated if necessary. Based on the automatic position and speed updates of all surface movements, the MAS also monitors that the planned trajectories are executed as instructed. When deviations occur, the MAS adjusts the routes where possible to minimise the impact, and alternatively replans the routes of the affected aircraft.

In the non-nominal case that it cannot find a viable solution, it requests the ATCOs to step in and resolve the situation, for which the MAS provides any decision-support needed. Besides the regular path planning according to the flight schedule, the MAS can also provide decision-support by calculating paths between specific points in the taxiway layout of a single or multiple aircraft. As such, it can be queried to obtain alternative routing options for one aircraft, or to test the effects that pending decisions would have on the overall traffic.

9.1.3 Interaction Agent and its Interaction Tools

The Interaction Agent acts as intermediary between any human operator and the MAS by providing interfaces and interactive tools. Rather than requiring operators to select the appropriate tool for a given situation, the Interaction Agent is designed to infer the intended functionality from their inputs, with the tools ensuring ease-of-use. The Interaction Agent as well as the MAS are assumed to work reliably so that human operators have sufficient trust in them. Furthermore, human operators are assumed to have sufficient training to use the interactive tools effectively. The relevant tools used in the interaction examples in Section 9.2 are outlined in the following, for which Figure 9.2 provides an overview.

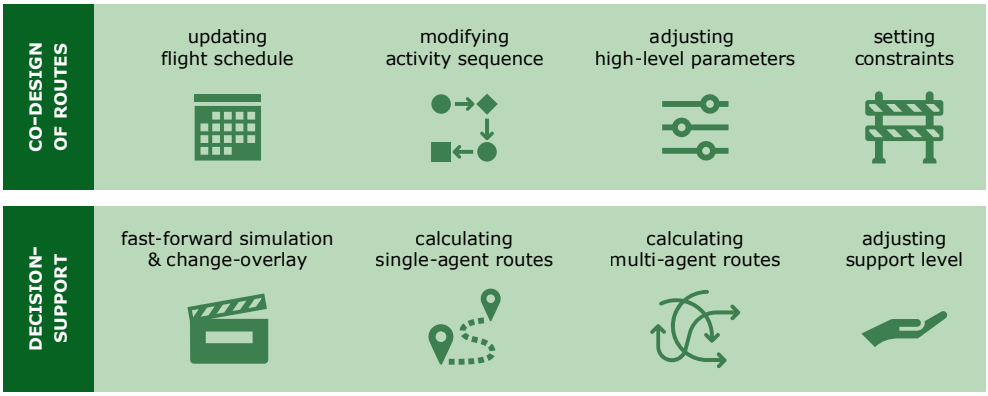


Figure 9.2: Overview of interaction tools for co-design of routes and decision-support provided by the Interaction Agent

Updating Flight Schedule

Aircraft start and goal locations, as well as start times, are obtained from the flight schedule (FS), based on the A-CDM milestones and inputs from human operators. The ATCOs can update FS entries when new schedule information becomes available.

Modifying Activity Sequence

Per aircraft, the MAS creates an activity sequence to account for the different operations, e.g. following a specific path for pushback, travelling to and holding at a holding point, etc. The ATCOs can adjust this sequence and its elements: they can for instance change the pushback path that must be followed, or the required holding duration. Likewise, data such as the estimated engine start-up duration is stored within the activity sequence and may be altered by human operators.

Adjusting High-level Parameters and Setting Constraints

The ATCOs can set and adjust various high-level parameters. These affect the path planning of all, a group of, or certain aircraft, and are valid for an extended period of time. Examples are:

- adjusting the conformance level to ATC procedures, further discussed in Section 9.2.1;
- blocking certain taxiway segments for maintenance work;
- adjusting the speed limits in general or in certain areas such as bay areas e.g. due to adverse weather conditions;
- setting general priority levels between aircraft groups, e.g. between arriving and departing aircraft.

Moreover, ATCOs can set constraints for specific aircraft that have a direct impact on their route. While some are valid for the entire taxiing, others are timed, i.e. issued for a specific duration. Examples are:

- constraining the start or goal location in case an aircraft shall leave before or arrive after a certain time point;
- selecting a certain location to be passed during taxiing, affecting the aircraft's activity sequence;
- assigning a specific priority relation between two aircraft, i.e. one has right of way over the other.

Fast-forward Simulation with/without Change-overlay

The MAS plans the 4D trajectories ahead of time, and deconflicts all routes within a pre-defined planning window. Thus, the ATCOs can inspect how the traffic will evolve in the upcoming period through a fast-forward simulation. The aircraft are categorised into one of six ICAO-types with coloured circles indicating the associated type sizes as shown in Figure 9.3 a). A cyan-coloured tug symbol indicates that a pushback tractor is coupled to the aircraft. A colour scheme is used to indicate the predicted engine status, with the aircraft colour being green when engines are switched off, orange during engine-start, and red when they are running. These symbols and colours are illustrated in Figure 9.3 b). As additional functionality, any aircraft path can be overlaid on the visualisation, with the executed segment shown in a darker shade and the remaining segment in a lighter shade of the same colour as the circle surrounding the aircraft whose route is displayed. As example, Figure 9.3 c) shows the blue and orange trajectories. Furthermore, the differences between two routing plans can be highlighted with a change-overlay: one or multiple previous/inferior route alternatives are depicted by 'shadow-aircraft' that are connected with a grey line to the new/superior solution.

Calculation of SAMP Routes

The MAS can calculate single-agent motion planning (SAMP) routes to quickly assess different routing options. These routes show the fastest possible paths that account for all relevant constraints, but are not yet coordinated with other traffic.

Calculation of MAMP Routes

Like the regular path planning, the MAS can also calculate alternative multi-agent motion planning (MAMP) routes based on altered input data, updated activity sequences, or changed high-level parameters or constraints like those mentioned above. However, to generate conflict-free trajectories for the selected set of vehicles, more computational time is needed than for calculating alternative SAMP routes.

Adjusting Support Level

The ATCOs can adjust the level of the notifications and decision-support provided by the Interaction Agent. With a high support level, the Interaction Agent carries out various analyses (see second example in Section 9.2.3) and prompts the ATCOs to take action.

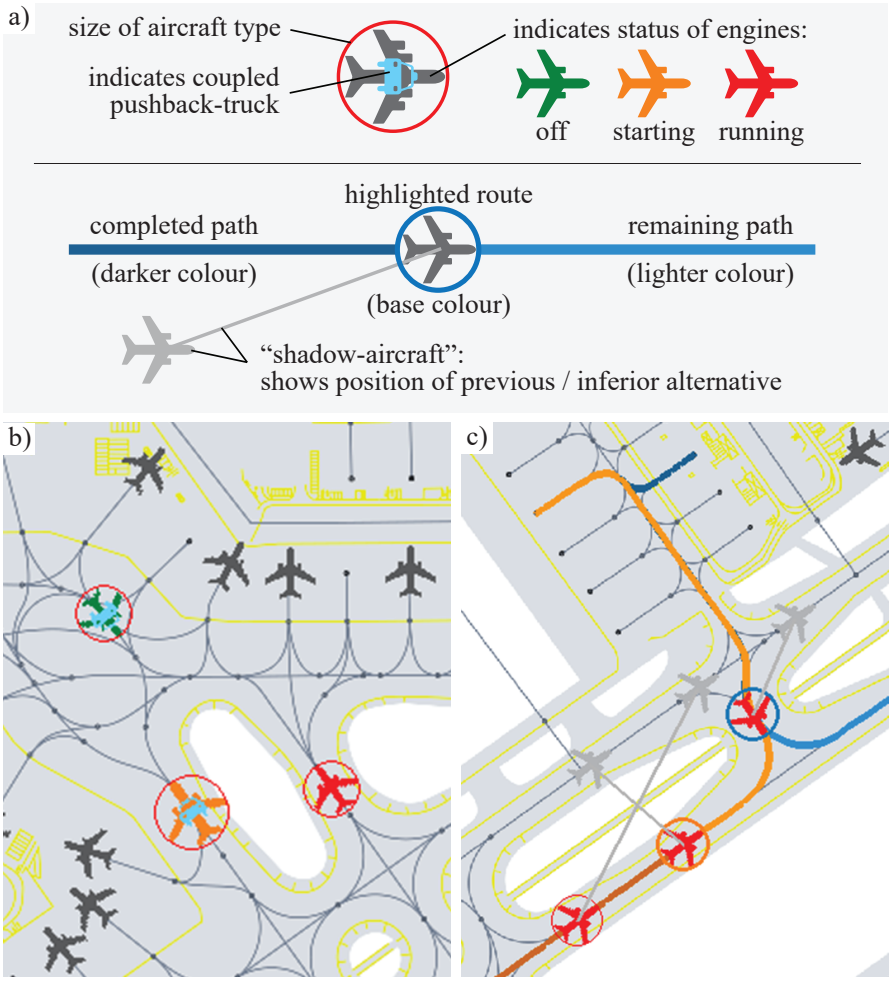


Figure 9.3: Illustration of fast-forward simulation with a) providing an overview of the symbols and colours, b) example without change-overlay, and c) example with change-overlay

9.2 Modelling and Implementation of Human–MAS Interactions

In the following, the abilities of the MAS to engage in interactions with humans using its interaction tools are demonstrated. A general motivation is provided, followed by illustrative use cases and a technical description of how such interactions can be modelled and implemented in the multi-agent system context. To make the cases more tangible, the narrative follows the fictitious air traffic controller Anne through her shifts at Amsterdam Airport Schiphol. Anne is assumed to be an experienced ATCO who has received sufficient training to use the interaction tools effectively. In addition to the technical elaboration, the broader applicability of the presented interaction mechanisms for any human operator within ASM Ops is discussed.

9.2.1 Case 'Conformance to Taxiway Procedures of ATC'

Motivation

In present-day operations, ATCOs use procedures and rules to create aircraft flows through the taxiway system. At Amsterdam Airport Schiphol, the main taxiways TWY-A, TWY-B, TWY-C, and TWY-D are used primarily in one direction as marked in Figure 9.4. However, ATCOs may deviate from these taxiway procedures on their own discretion. In historical data, most aircraft indeed follow the standard flow direction, with a few exceptions, for example to pass another aircraft that is holding on a taxiway to start its engines. For fully-automated operations, adhering to such procedures is not necessary, and paths could be optimised without such constraints, increasing the operational efficiency. However, such free-flow routes may not be comprehensible for human operators, especially when the traffic is dense. Furthermore, should the automation fail, the human operators may not be able to resolve and continue the operations. As previous studies showed, the conformance to procedures can improve ATCO acceptance, especially during the introductory phase of a new decision-support tool [186]. Accordingly, to enable meaningful routing support provided by the MAS model, the ATCOs must be able to adjust the conformance level to taxiway procedures of the path planning, as explored in the following.

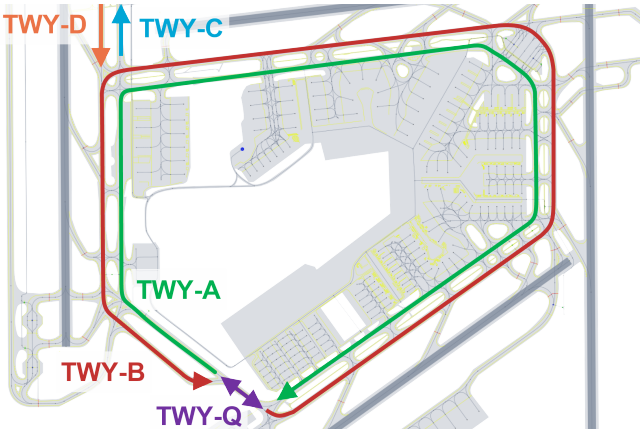


Figure 9.4: Standard taxiway flow directions as part of ATC procedures at Amsterdam Airport Schiphol

ATCO–MAS Interaction

Anne starts her shift in low traffic levels, and feels comfortable overseeing the operations with path planning having a low conformance to ATC procedures, and thus sets a low conformance level. A couple of hours later, with traffic increasing, she raises the conformance to a high level, and the MAS takes the new setting into account in the subsequent planning rounds. The trade-off between conformance level and solution quality is discussed below.

Technical Elaboration

The standard taxiway flow directions visualised in Figure 9.4 must be known to the MAS. The conformance to these taxiway procedures of ATC can then be adjusted by changing the associated cost of traversing a taxiway segment c_{seg} during path planning, by multiplying it with a cost factor $c_{\text{TWY},seg}$:

$$c_{seg} = (t_{\text{taxi}} + c_{\text{d}} \cdot d_{\text{taxi}})_{seg} \cdot c_{\text{TWY},seg} \tag{9.1}$$

with the taxi time t_{taxi} and taxi distance d_{taxi} along that segment. $c_{\text{d}} = 0.1\,\text{s/m}$ to convert the distance to unit time. For example, when $c_{\text{TWY}} = 5$, it is five times more expensive to traverse that taxiway segment in comparison to one with $c_{\text{TWY}} = 1$ for identical taxi time and distance. Since the sum-of-cost is minimised during path planning, non-standard taxiway flow directions that are assigned higher values for c_{TWY} are less likely chosen.

To analyse the impact of c_{TWY} on the conformance level, its sensitivity is analysed by varying c_{TWY} between 1 to 5 with the two days of operational data used in the preceding chapters. As shown by the red lines in Figure 9.5, the number of traversing taxiway segments in the non-standard direction sharply decreases for higher values of c_{TWY}. With $c_{\text{TWY}} > 1.2$, the path planning of the MAS adheres more to the standard taxiway flow directions than the paths taken in the historical operations (black lines). However, higher values of c_{TWY} cause the path planning to be more constrained, leading to a slight increase in taxi times as shown by the box-and-whisker plots per runway in Figure 9.6. Nonetheless, the delay patterns presented in Chapter 6 remain unaffected by changes in the adherence to the taxiway procedures.

Discussion

In the future, instead of letting the ATCOs set the conformance level manually, they could opt for letting the MAS do so dynamically, e.g. based on the number of flights to be routed in each planning round and the learned preferences of ATCOs. Moreover, other high-level routing parameters could be adjusted as well, for example:

- Setting default priorities, e.g. between arrivals and departures or between aircraft and ground vehicles: giving higher priorities to a certain group will tend to decrease their taxi times. Nonetheless, the ATCOs can deviate from these default priorities by assigning a specific priority-value to any vehicle.
- Setting the general speed limit for the entire taxiway system or certain areas like the bay areas, foremost dependent on the weather situation. Potentially, a data-driven AI subsystem could pose recommendations based on historical weather patterns.
- Setting the minimal or maximal duration that an aircraft holds at a remote holding location. This will impact the decision-support provided by the MAS, e.g. for arriving aircraft whose stand is still unavailable for a certain duration.

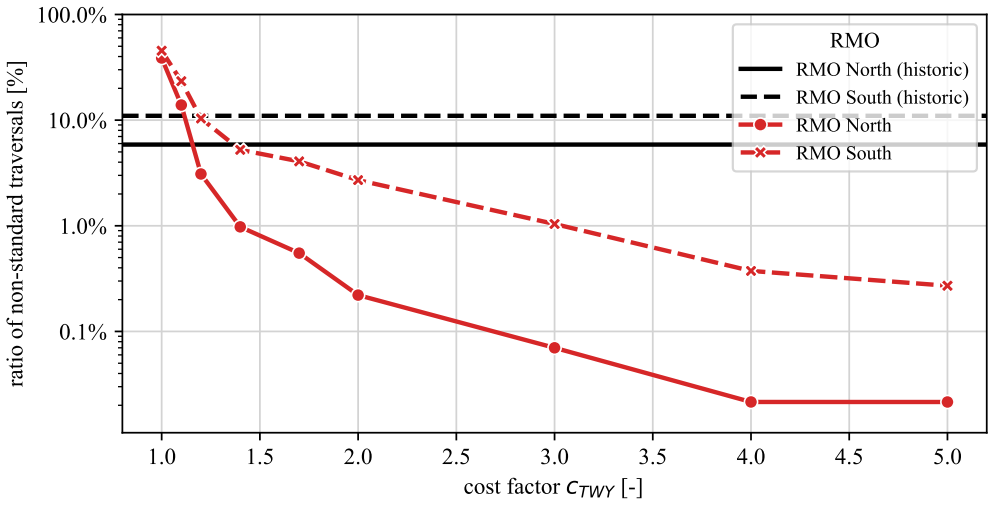


Figure 9.5: Ratio of non-standard taxiway traversals w.r.t. all traversals for two operational days under varying runway modes of operation (RMO). Black lines indicate ratios of the historical operations, red lines those of the simulated operations for varying values of the cost factor c_{TWY}.

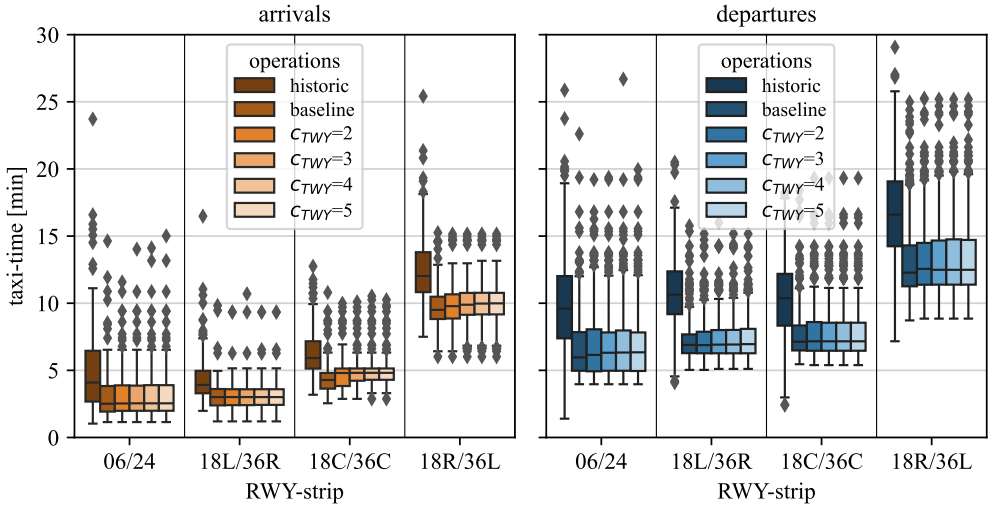


Figure 9.6: Comparison of taxi times per runway strip between historical operations and simulations with varying values of the cost factor c_{TWY}

9.2.2 Case 'Departing Aircraft is Delayed'

Motivation

In real-world airport operations, delays frequently arise out of various reasons, and must be dealt with. When planning the taxi routes of aircraft, delays occurring prior to the predicted start of the route can be counteracted by updating the prediction and replanning the route accordingly. Many delays may remain unknown to automated systems, and updating the predictions may require coordination among human operators as well as their expertise and problem-solving skills. Therefore, to achieve effective human–automation interactions, such changes and prediction updates must be steadily supplied to the MAS. In the following, one such example is explored.

ATCO–MAS Interaction

Ten minutes prior to the planned pushback time of AC-1, ATC is informed by the ground handler that multiple passengers have not boarded yet, delaying the ready-time by at least six minutes. Anne updates the **flight schedule** of AC-1 accordingly, and the MAS automatically replans all affected routes.

Technical Elaboration

All routes of aircraft are based on the flight schedule: per flight, the respective start point and time as well as the goal location are extracted from it when forming its activity sequence. The corresponding activities are updated when flight schedule entries change. The updates are then accounted for when the MAS replans the routes of all flights that are or will be taxiing within the planning window. If necessary, the replanning can also be triggered rule-based or manually. In the example, the target off-block time as start time of the route is updated.

Discussion

In a similar way, any adaptations to the flight schedule are handled, e.g.:

- Assigning a new stand to an arriving aircraft: the goal location in the activity sequence is adapted.
- Allocating deicing to a departing flight in winter conditions: intermediate activities are inserted into the sequence that demand the aircraft to taxi to one of the deicing locations at which it has to hold for a specific time to receive the deicing.
- Changes to the takeoff slot assigned by Eurocontrol (i.e. CTOT-slot): the constraining time to be at the runway is adjusted, potentially also affecting the holding time at the stand and/or remote holding location.

In general, any flight schedule change may lead to knock-on effects, e.g. trigger a stand conflict (i.e. the delayed departing flight is blocking the stand that an arriving aircraft is assigned to) that have to be resolved as well. This may entail further interactions between the MAS and the human operators. However, as such interdependent effects occur mostly after the original cause, there is likely more time for the human operators to request and act upon recommendations from the MAS or make informed decisions to resolve them.

9.2.3 Case 'Alternative Pushback Path'

Motivation

At Schiphol, the standard pushback and push-pull paths dependent on the aircraft types are defined in the airport manuals for all stands (see [153]). We integrated these paths in the airport layout and use them as part of the activity sequence for outbound flights. However, based on the historical data as well as on interviews with operational experts, it can be concluded that the ATCOs deviate from these standard pushback procedures in around 20 % to 30 % of times to further optimise the flows and taxi times of the involved aircraft. Thus, the MAS must accommodate to receive and process such informed changes based on the experience of the ATCOs. In the following, two examples are explored that showcase how such ad-hoc changes are dealt with in the MAS model.

ATCO–MAS Interaction – Part 1

The amount of traffic is low, and Anne decided to lower the **level of support** provided by the Interaction Agent. Using the **fast-forward simulation**, she checks how the traffic will likely evolve around bay area D. She notices that in around 5 min the departing aircraft AC-2 will start its engines in a location that blocks the stand entry for the arriving aircraft AC-3 for multiple minutes, with another aircraft at the neighbouring stand likely to request pushback clearance (Figure 9.7a). From her experience, she knows that a longer, non-standard pushback of AC-2 further into the bay area is unproblematic, allowing AC-3 to enter the stand faster. In turn, the bay area would be freed faster, which might also allow the other aircraft to start its pushback earlier. Thus, she is confident that the longer pushback of AC-2 will improve operational efficiency, and she manually adapts its pushback path. She then queries the MAS to compute **alternative MAMP routes** for all involved aircraft. After reviewing the pending changes to the aircraft trajectories using the **fast-forward simulation** (Figure 9.7b), she accepts the proposed new routing plans.

Technical Elaboration – Part 1

By specifying a longer pushback path, the activity sequence is automatically updated: during the pushback activity, the aircraft must follow the new path. While the changes are still pending the acceptance by the ATCO, the Interaction Agent keeps the original MAMP solution, and the original as well as the new activity sequences in cache. When queried by the ATCO, it replans the potential routes of all agents using the new activity sequence of the departing aircraft AC-2. Once the new solution is accepted by the ATCO, the MAS automatically sends the new routes to the affected aircraft.

ATCO–MAS Interaction – Part 2

With higher traffic load, Anne has increased the **level of support** provided by the Interaction Agent. She receives a notification from it to review upcoming routing plans for aircraft departing from bay area D/E as visualised in Figure 9.8: by following the standard pushback path, AC-4 (blue route) will be trailing AC-5 (green route), followed by AC-6 (orange route). This will cause delays for both AC-4 and AC-6. Since other traffic is not involved as validated with the **fast-forward simulation**, Anne changes the pushback path of AC-4 so that the aircraft can leave the bay area directly after engine start. She lets the MAS calculate the adapted **MAMP routes** (Figure 9.8b). The Interaction Agent notifies Anne that

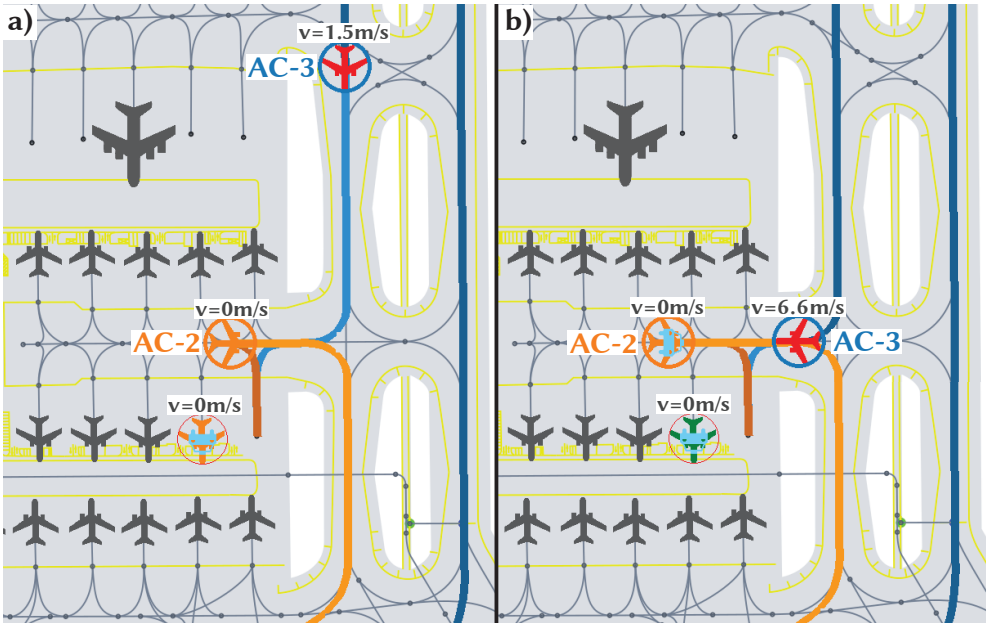


Figure 9.7: Fast-forward animations of a) the original trajectories for which the stand entry is blocked for AC-3, and b) the adapted routes when AC-2 uses a longer pushback path allowing AC-3 to enter the stand

the alternative routes result in an improved runway sequence (Figure 9.8d), reducing the taxi times of AC-4 by 2:40 and AC-6 by 1 min. Anne accepts the changed routes.

Technical Elaboration – Part 2

By increasing the provided level of support, the Interaction Agent automatically carries out various analyses of the concurrent routes to find those that likely profit from manual changes. In the case above, the two aircraft AC-4 and AC-6 are delayed in the same bay area due to having lower priority than AC-5, which hints at a possibility to improve the flow by manual changes. Akin to Part 1, the input of the ATCO leads first to an update of the activity sequence and second to a recalculation of the trajectories. The Interaction Agent then identifies the core changes as described above and lets the ATCO validate the new routes by animating the changes.

Discussion

The MAS stores the non-standard pushback paths per stand. Over time, and dependent on the chosen level of decision-support, the MAS can then automatically suggest a limited number of pushback alternatives, after having analysed that the impact on other traffic is minimal. Still, the ATCO will need to validate and accept the suggested change. Eventually, when a high level of automation is chosen, the MAS can incorporate these alternative pushback paths directly in the route optimisation, reducing the workload of the ATCO to review the suggestions.

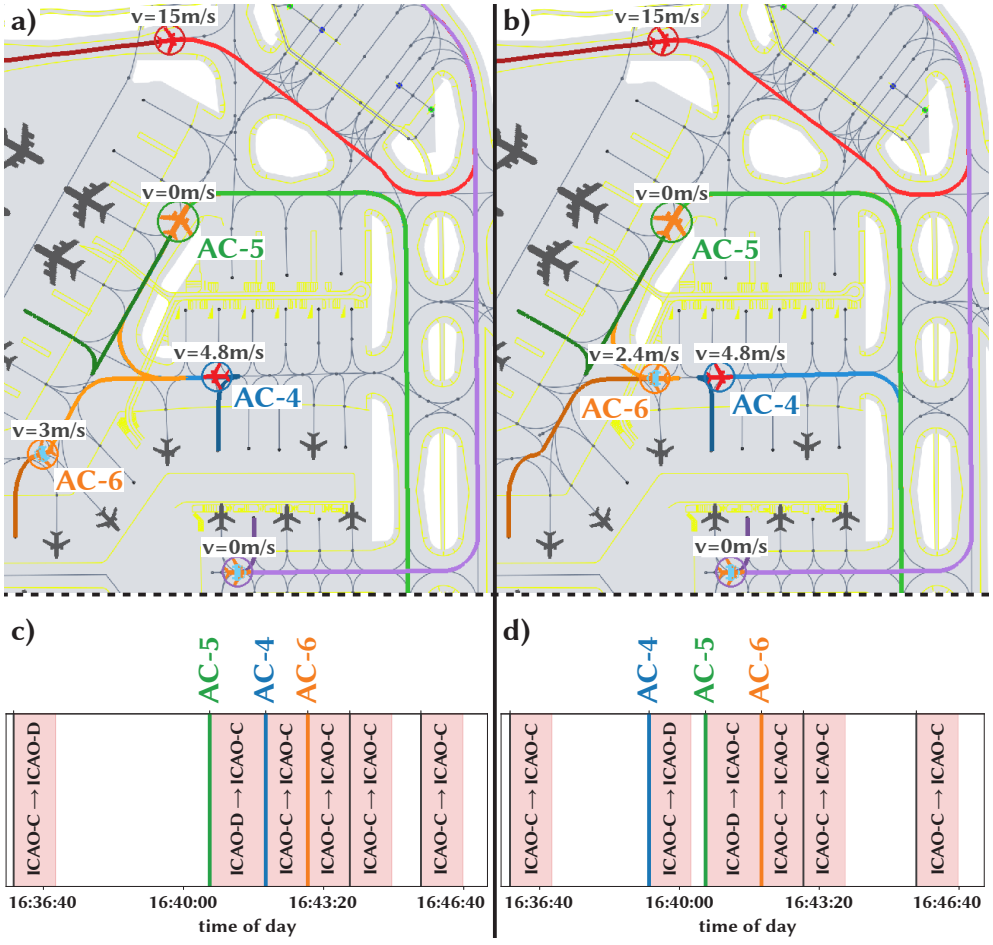


Figure 9.8: Fast-forward animations of a) the original trajectories and b) the adapted routes when AC-4 uses an alternative pushback path, with the respective runway sequences shown in c) and d)

9.2.4 Case 'Unavailability of Stand'

Motivation

The stands form one of the bottlenecks in the capacity of airports, and the aircraft stand allocation is a manifold problem of its own. Due to delays, arriving ahead of time, or allocation constraints among others, a stand may still be occupied by another aircraft at the time that an arriving aircraft could enter it. The route of the arriving aircraft must thus be adapted. Such cases are visible in the historical data, but were also raised in interviews with experts as both a common operational challenge and interesting use case for airside automation. In the following, an interactive example is presented to illustrate how stand unavailability can be resolved using the agent-based framework of human–automation interactions.

ATCO–MAS Interaction

Shortly before the flight AC-7 is landing at Schiphol's runway 06, Anne is informed by the ground handler that the chosen stand D04 is blocked / unavailable for another 20 min. Since the aircraft is landing soon, the MAS has already planned a conflict-free route. The ATCO displays the planned route (path displayed in Figure 9.9a), and notices that the aircraft would arrive approximately 15 min too early at the stand. First, Anne places a pending **goal-constraint** at the stand D04 for the blocked time period, and requests the Interaction Agent to compute initial alternatives without accounting for other traffic, i.e. **SAMP routes**. The Interaction Agent displays three initial solution strategies: (1) assign an alternative stand to the aircraft (path b), (2) let it hold at one of the remote holding points (path c), and (3) let the aircraft take a detour along TWY-B and TWY-A with minimum taxi speed (path d).

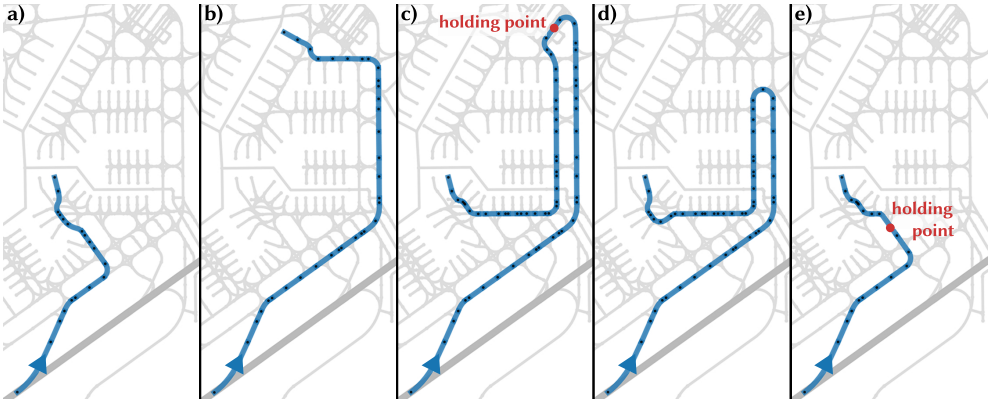


Figure 9.9: The stand of an arriving aircraft is unavailable: SAMP paths of a) original route, b) rerouting to a different stand, c) holding at a remote holding point, d) detour with minimum speed, and e) holding at a non-standard holding point

With these options in mind, Anne uses the **fast-forward simulation** to get an impression of the traffic situation in the upcoming 30 min. In turn, she disregards all three options: the alternative stand (1) is in another bay area and will likely not be acceptable for the airline, all remote holding points (2) are already occupied and would require extensive

changes to the routes of other aircraft, and she deems (3) to be impractical given the amount of traffic in the upcoming period. From her experience, Anne knows that the aircraft could alternatively hold in bay area C/D[1]. She selects the chosen point as non-standard holding location at which AC-7 shall hold for 15 min, and lets the MAS compute the **SAMP route** (path e). After checking her resolution option using again the **fast-forward simulation**, she accepts the changes to the **activity sequence** of aircraft AC-7. The MAS carries out partial replanning of the **MAMP routes** and notifies Anne that the changes have taken effect.

Technical Elaboration

Through the goal-constraint, AC-7 is not allowed to arrive at the stand prior to the end time, which is taken into account as strict requirement during path planning by the MAS. The Interaction Agent uses a set of options such as rerouting to another stand, or holding at a remote holding location among others to create appropriate activity sequences and let the MAS determine the corresponding SAMP-paths. As none of these route alternatives appear suitable to the ATCO, another activity sequence is created from the inputs provided by the ATCO: the chosen holding location is added as intermediate goal at which the aircraft must wait for the selected duration. The MAS carries out a partial replanning by deconflicting the routes of all aircraft that are affected by the new route of AC-7.

Discussion

In a similar manner, the ATCOs can add further constraints as well as additions to the activity sequence such as passing specific waypoints or waiting durations to any aircraft route. This enables a co-design of the routes that are then deconflicted by the MAS using alternative, unconstrained paths and/or speed adjustments. When such non-standard routing and resolution strategies are used more frequently, the Interaction Agent can learn to provide these alternatives automatically, and eventually use them to already adapt the original planning where possible.

9.2.5 Case ‘Emergency demands Aircraft to Return to Stand’

Motivation

Non-nominal situations may occur infrequently, but often require non-standard resolution strategies. Especially in such situations, the MAS model must provide flexibility in its provided support as well as its control decisions. In the following, a scenario is assessed in which a departing aircraft must return to its stand due to an emergency. To stress-test the path planning algorithm, an example is selected with heavy surrounding traffic near the bottleneck of TWY-Q (Figure 9.4).

ATCO–MAS Interaction

Aircraft AC-8 is following TWY-Q on its way to runway 36C, when the pilots declare an emergency via radio call to ATC. They request to be guided back to the stand as fast as possible. Anne quickly assesses the situation. She updates the **activity sequence** of the aircraft and adjusts the **priority** assigned to the aircraft. The MAS recalculates the **MAMP routes** of AC-8 and all affected aircraft accordingly.

[1] In the analysed historical data, aircraft frequently hold for multiple minutes at certain taxiway locations.

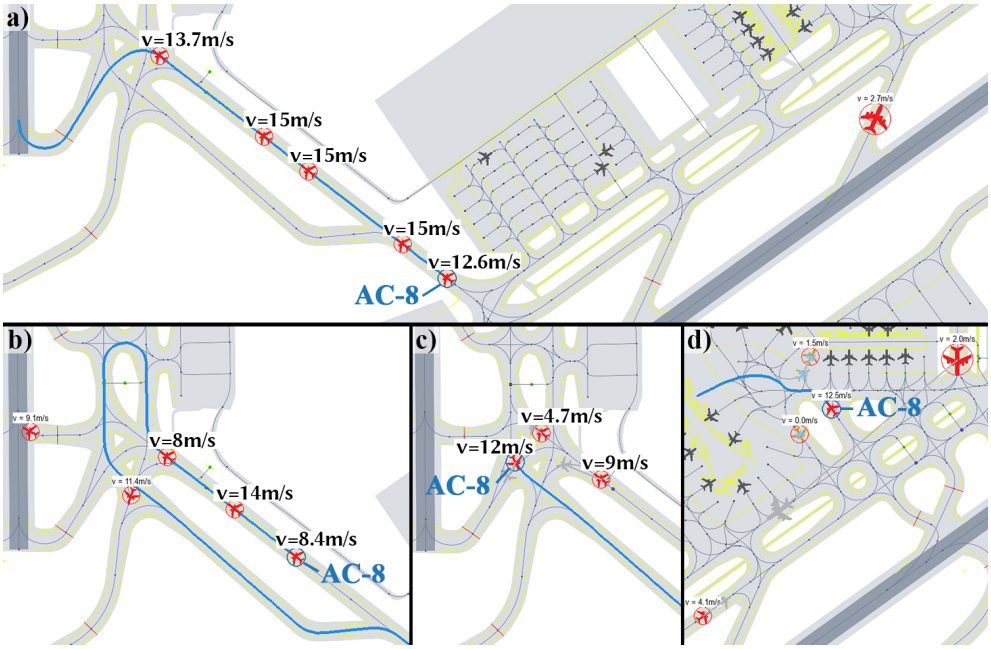


Figure 9.10: The blue-circled aircraft declares an emergency: a) shows the original route to the runway, b)-d) its route back to the stand with the effects on other traffic shown by shadow-aircraft

Technical Elaboration

The activity sequence must be redefined so that the aircraft is not routed further to the runway, but instead back to the original or alternative stand. Dependent on the emergency and general traffic situation, the ATCO can adjust the priority of the aircraft in comparison to other traffic: a high priority yields a fast return route while potentially more traffic is affected, whereas a low priority potentially lengthens its taxi time but creates minimal nuisance to the routes of other aircraft. In a range of tested emergency situations, changing the priority of the emergency-declaring aircraft did not have a significant impact on its own as well as the trajectories of affected aircraft. This suggests that the routing algorithm is able to recover to the desired level from such situations. For the default priority, Figure 9.10 shows the original route of AC-8 (a) as well as its route back to the stand (b–d). The effects on other traffic with respect to the original planning are visualised by shadow-aircraft.

Discussion

In urgent situations, ATCOs could also deviate more extensively from the regular procedures. For instance, they could let the aircraft turn immediately from TWY-A back to TWY-B which is usually not allowed since no taxiway centreline exists for this turn. Moreover, they could let the aircraft stop immediately, order emergency response teams to it, and reroute all surrounding traffic. We leave the further elaboration of such non-nominal scenarios to future work.

9.3 Discussion and Future Work

Presumably, more reoccurring cases will emerge in further discussions and continued validation with operational experts, likely leading to the identification of new interaction tools. These as well as other non-nominal situations can then further improve the decision-support provided by the Interaction Agent and the MAS. Moreover, we plan to incorporate different degradation modes into the MAS model to handle failures of the automation and conflict-free routing. A thorough analysis will be necessary to assess the system's resilience, to explore how to relay critical information to ATCOs to handle such failures effectively, and to ensure safe operations in any situation.

As emphasised in the introduction of this chapter as well as the thesis-wide scoping in Section 1.3, the explored cases aim to illustrate how the interactions between human operators and the MAS model can be embedded into the decision-support algorithms. As such, the explored interactions are intended to demonstrate conceptual feasibility rather than to provide validated evidence from human-in-the-loop experiments. Furthermore, systematically evaluating the impact on human factors such as workload, stress, situation awareness, trust, and susceptibility to potential biases constitutes a research area in its own right, and therefore falls outside the scope of this exploratory work.

Besides observability, the advanced automation concepts underlying the explored use cases can also facilitate assessment by incorporating transparency and explainability mechanisms that render the system's intent, constraints, and reasoning observable by the operator and, in turn, assessable by researchers. Such mechanisms have been widely linked to operators' situation awareness, mental model formation, and appropriately calibrated trust – particularly when designed to provide the *right* information without imposing excessive workload [187], [188]. At the same time, discussions with operational experts suggested that the time-critical nature of the operational phase in ASM Ops may leave limited cognitive bandwidth for detailed explanations in the moment. Instead, once sufficiently trained, operators may prioritise concise, actionable guidance and consult additional rationale primarily on demand (e.g. during anomalies, handovers, or when outcomes deviate from expectations). This motivates the need to establish, in the context of ASM Ops, which forms of transparency are operationally usable and which are necessary to support human–autonomy teaming in the long term, especially under non-nominal conditions.

More broadly, we acknowledge that human factors can significantly influence the decision-making in real-world settings. Thus, future research should also focus on creating a cognitive and behavioural model of the human operators within the Interaction Agent so that it can better predict and respond to the mental state of the human operators interacting with it. With such a model, the Interaction Agent could also aid in keeping the human operators better engaged and support in maintaining and honing their skills.

9.4 Conclusion

This chapter explored how human–automation interactions in airport surface movement operations can be modelled and integrated algorithmically into the MAS model on fully-automated ASM Ops used in the preceding chapters. For this purpose, a schematic agent-based framework was specified to enable human–automation interactions in ASM Ops. The interactions between human operators and the MAS model are facilitated

by the Interaction Agent providing interactive interfaces and tools. By examining use cases derived from workshops and interviews with stakeholders, this chapter demonstrated how ATCOs can oversee, influence, and co-design the conflict-free 4D trajectories computed by the MAS model, ensuring operational safety and efficiency. While the framework does not yet enable the envisioned human–autonomy teaming, it establishes the essential interaction architecture upon which such teaming concepts for ASM Ops can later be built. Nonetheless, as motivated above, substantial further research is required – particularly through real-time human-in-the-loop experiments – to advance the realisation of effective human–autonomy teaming in ASM Ops.

> Building on the validated modelling and planning capabilities developed throughout the thesis, this chapter explored how human–automation interaction mechanisms can be embedded into the MAS to enable operator involvement (addressing RQ 8), providing an outlook towards EASA Level 2 concepts. The next chapter concludes the thesis by synthesising the main findings across all research questions, reflecting on limitations and implications for future ASM Ops, and outlining directions for further research and development, including dedicated human-in-the-loop experiments of the interactions presented in this chapter.

10

Conclusions

This chapter summarises how the research questions of the thesis have been addressed, and reflects on the main findings. The discussion revisits the multi-agent system framework, the AS-MAMP algorithm, and their application to simulation studies at Amsterdam Airport Schiphol. The chapter also considers the resulting operational insights and exploratory work on human–automation interactions, before relating the findings to the overarching research objective and outlining directions for future work.

10

Airside operations at major airports are under increasing pressure as traffic demand grows, environmental constraints tighten, and expectations for safety remain uncompromised. As emphasised in the introduction (Chapter 1), this thesis concentrates on airport surface movement operations (ASM Ops). Here, increasing congestion, uncertain taxiing behaviour, and expanding operational complexity place additional demands on ATCOs and raise the risk of inefficiencies, increasing workload, and human error. At the same time, the transition towards more sustainable ground operations, such as the emergence of engine-off taxiing techniques, require a fundamental rethinking of existing concepts of operations.

Against this backdrop, advanced automation has been identified by regulators, researchers, and practitioners alike as a key enabler for future operational improvements [18], [64], [65], [189]. Yet, no existing modelling framework sufficiently captures the interrelated processes, heterogeneous actors, and fidelity requirements necessary to evaluate next-generation concepts for ASM Ops in a realistic and systematic way. The limitations of current models, which are often tailored to isolated subsystems, specific operational concepts, or based on simplifying assumptions, create a gap between innovative approaches and operationally viable solutions. To address this gap, the overarching objective of this thesis was to design models, methods, and tools to investigate next-generation concepts of operations for automated and sustainable airport surface movement operations.

With this goal in mind, the thesis first entered a development phase, in which a generalised multi-agent system architecture, a MAS model instantiation based on centralised path planning, and the dedicated Multi-Agent Motion Planning on Airport Surfaces (AS-MAMP) algorithm were designed. This phase established the conceptual and computational basis of the proposed modelling framework and evaluated its performance and sensitivities through fast-time simulations (Chapters 3 to 5).

Building on this foundation, the MAS model was applied in a series of operational studies, subsumed in the validation phase. These investigations examined the implications of fully-automated multi-engine taxiing, tug-enabled taxiing, aircraft towing movements, and used historical trajectory data to calibrate the model so that it closely replicates the operational conditions (Chapters 6 to 8).

Finally, in the exploration phase, it was investigated how the modelling framework could support future concepts for human–autonomy teaming. The MAS model was embedded into a schematic interaction framework, and selected use cases within ASM Ops were used to demonstrate how such interactions can be represented and probed in the model (Chapter 9).

The following sections present a reflection on the research questions (Section 10.1), discuss the findings in relation to the overarching research objective (Section 10.2), and outline directions for future work (Section 10.3).

10.1 Review of Findings

In this section, the research questions corresponding to the three phases of this thesis are reviewed and their main findings are discussed.

10.1.1 Research Questions in Development Phase

In the development phase, the thesis focused on designing and evaluating the underlying modelling framework (RQ 2) with embedded path planning logic (RQ 3), guided by modelling requirements for studying next-generation ConOps of ASM Ops (RQ 1).

RQ 1 Modelling Requirements

Which modelling requirements must be met when investigating next-generation concepts of operations using a high-fidelity simulation of real-world airport surface movement operations?

main findings

- The system model shall:
  - accommodate multiple actors with varying but interrelated tasks, complying with safety standards;
  - support differing airport environments, operational goals, and deal with operational uncertainties;
  - use decision-support algorithms that are effective and efficient.
- To allow design freedom, the model architecture must:
  - be modular and extensible to support ConOps across varying degrees of technological maturity, automation, and control centralisation;
  - have a hierarchical element, even within distributed system designs, to meet the strict safety standards of air transport operations;
  - support fast-time and real-time simulations, enabling both large-scale algorithmic studies and human-in-the-loop experiments.

The modelling framework must capture the full operational richness of airport surface movements to allow drawing meaningful conclusions and recommendations through model-based decision-support and design-support tools. Models that restrict themselves to simplifying assumptions or isolated subsystems lack the fidelity needed to evaluate, compare, and contrast next-generation ConOps across varying degrees of technological maturity, automation, and control centralisation. Since future ASM Ops will likely evolve through iterative design cycles, the simulation framework must be modular and adaptable. At the same time, the safety-critical nature of airport operations requires hierarchical oversight to guarantee operational integrity, even when exploring distributed decision-making structures. Combined with the need for both fast-time and real-time execution, these insights motivated to develop a modelling architecture following the multi-agent system paradigm, which led to the second research question.

RQ 2 Multi-Agent System (MAS) Model

Using the multi-agent system paradigm, how can the modelling requirements be addressed to design a realistic simulation model for studying next-generation concepts of operations?

main findings

- A multi-agent modelling architecture with a flexible, hierarchical-distributed structure provides a suitable basis for satisfying the modelling requirements and exploring next-generation ConOps. The proposed generalised architecture offers design freedom to develop more elaborate and tailored models, while allowing to reuse the overall simulation environment as well as model components.

The proposed generalised multi-agent architecture is illustrated in Figure 10.1. The four agent categories (Scheduling Agents, Routing Agents, Guidance Agents, and Moving Agents) are organised in hierarchical layers. Within each category, multiple agents may operate and engage in centralised and/or distributed control mechanisms, informed by communication and coordination within and across the hierarchical levels. Through their perception, all agents can potentially receive relevant data from environmental objects to inform their decision-making processes.

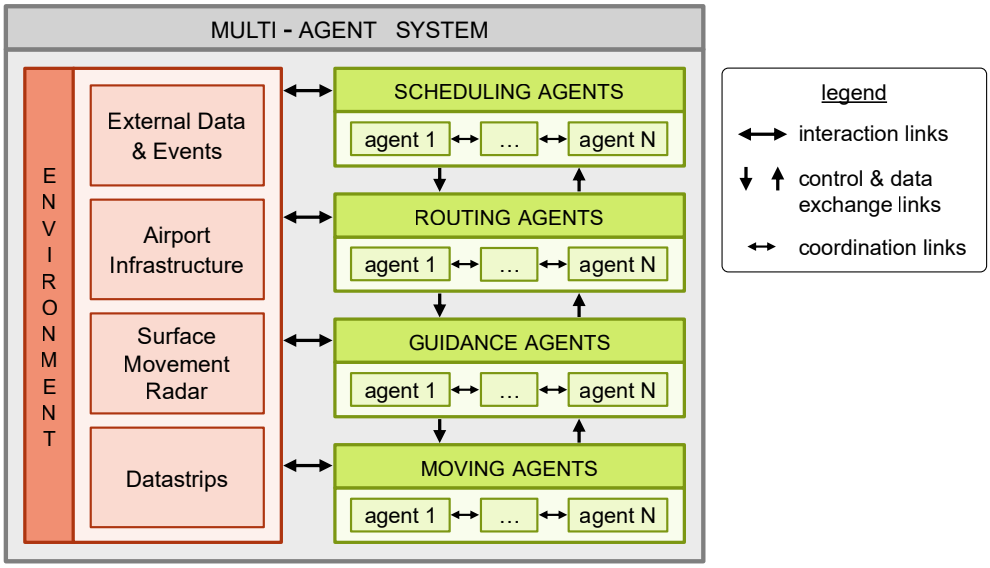


Figure 10.1: Generalised hierarchical-distributed multi-agent system modelling architecture to study diverse concepts for next-generation airport surface movement operations

Based on the generalised architecture, a large variety of MAS models can be instantiated that are tailored to a specific ConOps, while retaining the overall system structure. In turn, the system-level performance of models representing varying ConOps can be compared with each other, and iteratively refined in fast-time simulations and human-in-the-loop experiments with stakeholders. Nonetheless, significant gaps remain in demonstrating how such control system models could be adapted to the constraints of live operations, as discussed in Section 5.2. Therefore, the generalised architecture should be seen as a research tool to structure and analyse models of future operational concepts, rather than a deployable solution in its present form.

So far, several model instances were developed and used in research projects with both low [34] and high levels of automation [35]. In this thesis, one model instance based on a ConOps with centralised planning and a fully-automated control system was elaborated. The centralised Routing Agent issues conflict-free trajectories by assigning priorities between the agents that move concurrently over the airport surface. It shares the routing plans with the distributed Guidance Agents that subsequently instruct the Moving Agents which in turn execute the commands accordingly. The backbone of this MAS model is the centralised decision-logic of the Routing Agent. Since off-the-shelf multi-agent path planning solvers did not satisfy the modelling requirements, substantial algorithmic development and evaluation were required, thereby motivating the third research question.

RQ 3 Multi-Agent Motion Planning

Which algorithmic developments are necessary for multi-agent motion planning algorithms to generate safe, efficient, and computationally scalable trajectories in realistic airport environments?

main findings

- The two-level AS-MAMP solver was developed to generate conflict-free trajectories for heterogeneous agents, incorporating realistic shapes, separation requirements, and kinodynamic constraints while coordinating all concurrent surface movements efficiently. With dynamically constructed priority orders, AS-MAMP shows approximately quadratic time complexity, which reduces to almost linear complexity with predefined (fixed) orders.
- The SIMP algorithm, a SIPP-based kinodynamic single-agent solver, was introduced to handle finite accelerations and bounded speeds, enabling the computation of kinodynamically feasible trajectories in complex, graph-based airport environments. The runtime of SIMP scales almost linearly with the number of explored states.

To meet the modelling requirements, the path planning algorithm representing the centralised decision logic must accommodate heterogeneous agent shapes and kinodynamics, enforce strict spatial-temporal safety constraints, and remain computationally tractable for the traffic levels of large hub airports. This led to the development of the Multi-Agent Motion Planning on Airport Surfaces (AS-MAMP) algorithm. AS-MAMP is a two-level solver that uses windowed prioritised planning to compute conflict-free 4D agent trajectories. As ASM Ops are a safety-critical system, the AS-MAMP algorithm is designed to return a solution only when it is certainly conflict-free. Moreover, as key contribution to the MAPF-field, Chapter 4 outlined how the SIPP algorithm [60] can be enhanced to work with finite accelerations as well as upper and lower speed restrictions of agents. These algorithmic enhancements led to the novel Safe Interval Motion Planning (SIMP) algorithm. Despite being used as low-level solver in AS-MAMP, it can also serve as single-agent planner for other graph-based problem instances under kinodynamic constraints.

The AS-MAMP algorithm was benchmarked across multiple scenarios using both synthetic and real-world airport layouts, demonstrating that it meets the required performance envelope. Even in traffic scenarios representative of Amsterdam Airport Schiphol, including cases with extensive tug-enabled taxiing, as well as high volumes of aircraft towing movements, the algorithm provided safe, conflict-free trajectories. Simulations on a synthetic layout further pushed AS-MAMP to its limits, showing that it can cope with taxiway occupancy levels well above those currently experienced at busy European airports. These results provided the basis for the subsequent operational studies in the validation phase, where AS-MAMP served as the core planning component of the MAS model.

When constructing the priority order dynamically in AS-MAMP, it scaled with approximately quadratic time complexity. On standard hardware, deconflicting the trajectories of 100 agents took around 120 s. When using a fixed priority order such as FCFS, the runtime of AS-MAMP is determined largely by SIMP, whose computational complexity scales almost linearly. In turn, the required runtime to compute conflict-free trajectories for up to 100 agents remained below 30 s.

The windowed planning approach enables AS-MAMP to handle operational uncertainty by limiting replanning to a small, relevant subset of agents, thereby enhancing the algorithm's overall robustness. Moreover, it allowed to reduce the number of agents involved in each planning step to fewer than 50, thereby substantially lowering the runtime required for AS-MAMP to compute a solution. Potential runtime enhancements were discussed in Section 5.2, suggesting that sub-second runtimes may be attainable, which would constitute a major step towards real-time deployment.

10.1.2 Research Questions in Validation Phase

In the validation phase, the developed MAS model was applied to realistic case studies at Amsterdam Airport Schiphol, assessing fully-automated multi-engine taxiing (RQ 4), tug-enabled taxiing (RQ 5), calibrating the model parameters using historical ground track data (RQ 6), and aircraft towing operations (RQ 7).

RQ 4 Multi-Engine Taxiing (MET)

Using the MAS model, how do fully-automated multi-engine taxiing operations compare to the present-day, real-world operations in dense flight traffic at Amsterdam Airport Schiphol?

main findings

- The conducted simulations indicate that higher levels of automation can improve operational efficiency and predictability, enabling airports to accommodate the rising traffic demand. Though to a more limited extent, these improvements also contribute to reducing emissions associated with ASM Ops.
- The scale of the observed benefits is highly sensitive to the underlying operational parameters, particularly aircraft kinematics and takeoff separations. This sensitivity motivated to recalibrate the model parameters (addressed in RQ 6, Chapter 8), underscoring that the use of realistic parameters is essential when evaluating new ConOps.
- Higher, yet operationally feasible, taxi speeds and smoother flows become achievable when aircraft follow conflict-free 4D trajectories generated through priority-based coordination schemes. Such deconflicted trajectories also help mitigate stop-and-go movements typical of present-day operations.

The simulations of fully-automated multi-engine taxiing at Amsterdam Airport Schiphol showed – within the scope of this thesis – that higher levels of automation can substantially improve the efficiency and predictability of ASM Ops compared with current operations. Using conflict-free 4D trajectories in dense traffic, the MAS model reduced taxi time variability and enabled accurate predictions of the remaining taxi time to stand or runway. These findings are broadly consistent with earlier work on surface trajectory-based operations (STBO), that highlighted smoother flows, reduced stop-and-go behaviour, and improved predictability when conflict-free 4D trajectories are used operationally [28], [190]. While the simulations also indicated a modest reduction in associated emissions, larger emission cuts require complementary measures such as engine-off taxiing, as discussed under RQ 5.

A second key insight is that the magnitude of the simulated benefits is highly sensitive to underlying operational parameters, particularly the aircraft kinematics and takeoff separations. In Chapter 6, the calibrated kinematics led to taxi time reductions of around 3 min, equal to approximately 30 % compared with historical operations, which operational experts judged as too optimistic. This motivated to adjust the model parameters to better align with historical trajectories, as outlined in Chapter 8 and further evaluated under RQ 6. Notably, previous literature on ASM Ops frequently omitted key modelling parameters. The findings underscore that the kinematic assumptions are not a minor modelling detail but rather a central driver of reported benefits, and should thus be explicitly stated and critically assessed in operational studies.

The third main finding concerns the operational potential of coordinated 4D trajectories with priority-based deconfliction. In the MAS model, higher taxi speeds remain feasible because aircraft follow pre-planned, conflict-free 4D trajectories that are deconflicted through priority assignment, which suppresses the stop-and-go patterns typical of current ASM Ops. This is in line with NASA–DLR trajectory-based taxi concepts [29] as well as the 'Follow-the-Greens' concept [191]. Moreover, as part of its decision logic to coordinate the surface movements, the Routing Agent uses predicted taxi times to determine when holding is required, whether for inbound flights awaiting a free stand or outbound flights constrained by a CTOT-slot. When necessary, it directs aircraft to hold or detour via remote platforms. The fast-time simulations showed that this rule-based mechanism effectively resolves such situations: aircraft reach stands only when available and depart within their CTOT windows. This transparency and reliability can also enhance operator trust in automated decision-support systems, as further discussed under RQ 8.

RQ 5 Tug-Enabled Taxiing (TET) of Outbound Flights

Using the MAS model, which operational consequences may result from requiring all outbound flights to use tug-enabled taxiing under the assumption of fully-automated operations in dense flight traffic at Amsterdam Airport Schiphol?

main findings

- TET operations substantially reduce taxi-out fuel consumption and shift emission hotspots away from bay areas, lowering ground-level exposure of personnel to exhaust emissions and particulates.
- When embedded in an automated planning framework, TET operations preserve runway capacity and inbound performance, while outbound taxi times increase only moderately and mainly on longer taxi routes due to tug speed limitations.
- The operational impact of TET is highly dependent on the airport taxiway system, tug performance (e.g. maximum speed), and the underlying ConOps (e.g. decoupling locations, resource allocation).

The fast-time simulations of tug-enabled taxiing (TET) for all outbound flights at Amsterdam Airport Schiphol showed that TET can substantially reduce the environmental footprint of taxiing with only limited impact on operational performance. In the MAS model, towing aircraft from the bays to decoupling points near the runways reduced fuel consumption during taxi-out by up to 69 %. Moreover, an evaluation of the emission hotspots highlighted that the engines remain off especially in and close to the bay areas. This leads to a pronounced reduction of emission hotspots in and around the stands, implying lower exposure of ground personnel to the engine exhaust gases and ultra-fine particles. These results are consistent with other studies, pointing out that engine-off taxiing techniques such as the TaxiBot concept are the most effective options for reducing taxi fuel burn and local emissions compared to multi-engine or single-engine taxiing [11], [192], [193]. As additional benefit highlighted in a recent study, TET operations substantially reduce noise emissions and thus lower the risks of hearing impairment for ground personnel [194].

In contrast, one position paper pointed out that TET operations would significantly impact the procedures and workload of ATCOs [195]. While the posed concerns must be addressed before implemented in real-world operations, much of the risks seem to depend on the design of the control system rather than the technology enabling TET operations. The observed sensitivity of outbound taxi times to tug speed and layout-specific decoupling locations underscores that local airport characteristics and technological choices (e.g. maximum tug speed, electric vs. diesel tugs) are critical for real-world performance, and should be explicitly accounted for in future concept evaluations and implementation studies.

RQ 6 Calibration of MAS Model

How can historical aircraft ground tracks be used to calibrate the MAS model so that it closely reproduces real-world taxi traffic, and what do they reveal about suitable taxi speed limits for highly-automated operations?

main findings

- The developed trajectory projection method effectively reconstructs realistic ground movements from historical tracks and is transferable to other airports, provided a detailed taxiway graph and sufficient data quality.
- MAS simulations calibrated with parameters derived from projected tracks result in taxi times that closely match historical values, underscoring the realism and practical relevance of the calibrated model.
- The projected trajectories indicate that higher taxi speeds are achievable in many parts of the airport. Introducing progressive, area-specific speed limits led to performance gains of approximately 10 %.
- Reducing the minimum takeoff separations produced a similar performance improvement, highlighting the strong coupling between departure-sequencing policies and the performance of airside operations.

The analysis of historical aircraft ground tracks showed that the developed trajectory projection method can reliably reconstruct realistic taxi movements along the computational graph representing the airport taxiway network. This enabled to determine more accurate A-CDM milestones and more detailed performance indicators than the conventional estimates allow. Because the method mainly depends on the availability of a sufficiently detailed network graph and sufficient surveillance data quality, it is transferable to other airports and data sources such as ADS-B, in line with recent work on data-driven A-CDM and A-CDM-lite concepts that derive operational milestones directly from surveillance data [164], [166].

Using parameters obtained from the projected trajectories, the calibrated MAS model reproduced historical taxi times closely, underscoring the practical realism of the model and strengthening confidence in subsequent what-if analyses. At the same time, the results highlighted the tight coupling between runway departure-separation rules and surface performance: reducing minimum take-off separations lowered runway occupancy and upstream taxiway queues and improved the operational performance by roughly 10 %,

particularly around bay areas located close to active runways. This is consistent with prior work showing that departure management, taxi-time prediction, and surface planning are strongly interdependent [146], [196].

Finally, the projected trajectories revealed that higher taxi speeds are operationally feasible in many parts of the airport. Introducing progressive, area-specific speed limits produced performance gains of a similar order (around 10 %). Supporting pilots through tailored 4D trajectory guidance or real-time speed advisories could help achieve these higher, yet operationally realistic, speeds. Nevertheless, any form of advisory or trajectory-based guidance must be designed to preserve safety margins under varying visibility, surface conditions, and traffic densities, highlighting the need for operational trials and human-in-the-loop assessments, as discussed further in Section 10.3.

RQ 7 Aircraft Towing Movements

Using the calibrated MAS model, which impact do aircraft towing movements have on the taxi performance of regular traffic?

main findings

- Even with substantial volumes of towing movements, the simulations showed no significant system-level impact on taxi performance of regular traffic (e.g. taxi times, delays, runway throughput) under the centralised coordination scheme.
- Local effects on individual trajectories of taxiing aircraft were observed, indicating that towing movements can still influence the locally observed traffic density, and should therefore be included when discussing replays of fast-time simulations or conducting human-in-the-loop assessments with ATCOs.
- Using an appropriate coordination concept, aircraft towing operations can effectively increase connected stand capacity without degrading taxiing performance.

This research question was motivated by feedback from operational experts and ATCOs at Schiphol who stated that aircraft towing movements likely influence taxiing traffic. The calibrated MAS simulations with centralised path planning, however, indicate that the system-level impact on regular taxi performance remained limited, even under scenarios with considerable towing volumes. Average taxi times, delays, and runway throughput were only marginally affected, suggesting that towing flows at Schiphol may be less disruptive at the system level than commonly assumed.

At the same time, the results showed that towing movements do affect individual trajectories and local interactions of taxiing aircraft. For this reason, towing operations should be included whenever high-fidelity replays or human-in-the-loop experiments are conducted with ATCOs, to preserve realism and to capture potential workload effects that are not apparent in aggregated performance indicators. The analyses further suggest that towing can be used to increase connected stand capacity without degrading taxi performance, provided that it is embedded in a suitable coordination scheme. This emphasises that aircraft towing operations could serve as an important design variable for future concepts of operations aimed at more flexible stand and gate utilisation.

10.1.3 Research Questions in Exploration Phase

In the exploration phase, the thesis introduced a schematic agent-based framework for human–automation interactions in ASM Ops (RQ 8), providing a basis for exploring future concepts of human–autonomy teaming.

RQ 8 Human–Automation Interactions

How can human–automation interactions be embedded into the MAS model, methods, and tools for next-generation airport surface movement operations?

main findings

- The MAS model was extended with an interaction schema that allows ATCOs to oversee, influence, and co-design the conflict-free 4D trajectories generated by the system model, evolving the MAS model from a pure optimiser into an interactive decision-support tool.
- Rule-based mechanisms, such as the Routing Agent's logic for coordinating holding, stand availability, and CTOT compliance, sufficed to reliably resolve operational situations. As such mechanisms are in general transparent and explainable, they may support operator trust in automated decision-support systems.
- Feedback from the ASTAIR real-time validation exercise highlighted that the explored interactions are only a first step: some concepts aligned well with ATCO responsibilities, whereas others belonged to different stakeholders, underscoring the need for iterative, role-specific co-design.

The explorative cases showed that human–automation interactions can be embedded into the MAS model by giving ATCOs the ability to inspect, modify, and request alternative conflict-free 4D trajectories. In combination with the rule-based mechanisms developed earlier, such as the Routing Agent's logic for ensuring that aircraft only arrive at free stands and depart within their CTOT windows (see RQ 4), the proposed agent-based model exhibits behaviours that are transparent and easy to interpret. During the ASTAIR validation exercises [197], real-time simulations of the considered use cases were conducted. ATCOs emphasised the operational realism of the MAS model and described its behaviour as transparent and predictable. Their feedback suggests that the model's responses align well with controllers' expectations and support the development of trust in automated decision-support tools.

On the other hand, the explored interaction concepts remain exploratory rather than complete. The ASTAIR sessions made clear that some use cases were perceived as meaningful and aligned with ATCO responsibilities, whereas others were considered irrelevant or belonging to different stakeholders. The work on RQ 8 therefore provides an initial, technically viable and conceptually coherent basis for embedding human–automation interactions in ASM Ops. Building on the explorations, future research can iteratively refine interaction concepts, formalise teaming requirements, and evaluate their impact in high-fidelity human-in-the-loop experiments.

10.2 Reflection of Research Objective

Having addressed the individual research questions, this section reflects on the overarching thesis objective. It summarises how the main findings relate to the design and application of the MAS framework for studying next-generation concepts of ASM Ops, and subsequently elaborates on the points.

Research Objective

Design models, methods, and tools to investigate next-generation concepts of operations for automated and sustainable airport surface movement operations.

main findings

- The designed high-fidelity models, methods, and tools are well suited for investigating next-generation concepts of operations for automated and sustainable airport surface movement operations, producing fruitful and trustworthy results.
- The operational studies at Amsterdam Airport Schiphol show that calibrated automation, refined procedures, and sustainable taxiing concepts can yield meaningful efficiency and environmental gains without compromising safety or system stability.
- The applications in SESAR projects, together with the exploratory interaction framework developed in this thesis, demonstrate that the MAS model can underpin trajectory-based decision-support tools and constitutes a suitable starting point for future human–autonomy teaming in ASM Ops.
- The thesis showed that designing high-fidelity MAS models and efficient motion planning algorithms demands labour-intensive, iterative development, which may explain the scarcity of comparable models in current research.

To address the three challenges identified in Chapter 1, i.e. growing traffic demand, stringent environmental targets, and uncompromising safety expectations, the proposed generalised MAS architecture allows next-generation ConOps for ASM Ops to be studied in a unified framework. The results from RQ 4–RQ 7 demonstrate that the MAS model with centralised planning consistently provided trajectories that are safe, efficient, predictable, and environmentally friendly.

The conducted analyses demonstrate how combining detailed trajectory analysis with a high-fidelity MAS model can reveal where operational bottlenecks arise, which adjustments offer meaningful efficiency and sustainability gains, and where common assumptions may overstate system-level impacts. Although the study focused on Amsterdam Airport Schiphol, the insights regarding runway dominance, towing effectiveness, and realistic speed profiles are relevant to other major airports seeking to enhance the efficiency and predictability of their airside operations. The findings also suggest that substantial improvements can be achieved through relatively targeted changes to procedures and parameters. Nonetheless, translating these results into operational practice requires a careful evaluation of safety constraints, human factors, and implementation feasibility.

The practical relevance of the MAS model has been underscored by feedback from operational experts and ground controllers during face-validation sessions, who highlighted the level of detail and realism when viewing playbacks of the fast-time simulations. Stakeholders involved in several master thesis projects [198]–[201] emphasised the usefulness of the simulator for diverse applications. These ranged from analysing infrastructure changes and tug task assignment to assessing the impact of taxiway maintenance on ground-controller workload. Together, these studies illustrate the versatility of the MAS framework as both a decision-support and design-support tool.

Adapted versions of the MAS model were also used in the European SESAR JU projects AEON [34] and ASTAIR [35]. In AEON, the MAS model formed a central element of the concept of operations for sustainable taxiing [36], [38], even though it could not be employed in the final real-time simulations. The low level of automation and random human inputs in those experiments risked overloading the AS-MAMP algorithm with constraints and large deviations between planned and executed trajectories. In ASTAIR, the MAS model provided the decision-support intelligence behind the investigated ConOps for high levels of automation. In the real-time human-in-the-loop simulations, controllers appreciated that the provided suggestions adhered to standard airport rules and routing procedures. In turn, this reduced the need for continuous verification and allowed them to focus on potential conflicts. Some ATCOs reported feeling underloaded in nominal scenarios. Under higher traffic loads, however, they experienced the system more as a collaborator and trusted it to generate conflict-free trajectories, while they concentrated on non-routine events. Besides, pilot feedback from workshops indicated broad support for datalink-based clearances, provided that instructions remain clear, timely, and well integrated into cockpit systems to preserve situation awareness and compliance.

Throughout this thesis, the MAS model architecture and the AS-MAMP algorithm were refined through iterative development and verification cycles. Prior work has noted that multi-agent systems modelling is inherently labour-intensive and demands extensive testing and validation [202]. This challenge equally applies to the design and implementation of realistic yet computationally efficient path planning algorithms, as evidenced by the substantial development effort required in this work. As a likely consequence, only limited research addresses high-fidelity, operationally grounded MAS models for airport surface operations, which is mirrored in the comparatively sparse literature in this area.

> Overall, the simulation results obtained throughout this thesis demonstrate that the proposed MAS model for fully-automated operations with centralised planning enables safe, efficient, and scalable surface movement operations under realistic airport conditions. It thus provides a viable benchmark for studying diverse ConOps for next-generation ASM Ops. Accordingly, the main objective of this thesis is considered successfully achieved. At the same time, many research challenges remain. Some directions for future work are outlined in the next section.

10.3 Limitations and Research Directions

Following from the scope and assumptions listed in Section 1.3, this thesis has various limitations. Some of these are listed below, and future research directions are pointed out.

Suggested Research Direction

Quantify various sources of uncertainty in the operations and novel operational concepts, and how to deal with these in the MAS model.

Due to a lack of available historical data, not all operational parameters could be sufficiently studied, with the engine-start durations and their variability across aircraft types and turnaround times as examples of such. Moreover, a systematic quantification of the primary sources of uncertainty during the operational phase of airside operations, i.e. within the minutes leading to and during the taxiing phase, is missing, especially evaluated on the basis of the future concepts of operations investigated in this thesis.

Examples of the various open questions are: how accurate are the TOBT predictions within the 10 min leading to the actual pushback from the stand, using e.g. smart camera systems? How precise can various predictions of operational uncertainties as well as the experience and situation awareness of human operators inform the path planning of individual agent trajectories? To which extent can technical failures and their effects be forecasted?

Answers to these questions could also determine whether the use of buffer times in combination with partial or overall replanning of conflict-free trajectories as done in this work suffice to deal with the predominant uncertainties likely to occur in next-generation concepts of operations. Notwithstanding, future research should improve the real-time adaptability of the MAS model, for example through local trajectory adjustments issued by the Guidance Agents to counteract any deviations during plan execution. Furthermore, stochastic models to represent and analyse uncertainties due to sensor noise, communication delays, or operational deviations and disruptions should be incorporated into the MAS model.

Suggested Research Direction

Address the shortcomings of the AS-MAMP algorithm as centralised decision-logic for path planning.

The AS-MAMP algorithm assigns priorities between agents to resolve pending conflicts, and returns conflict-free and efficient trajectories for all agents moving over the airport surface concurrently. The conducted fast-time simulations highlighted its effectiveness. However, as centralised path-planning mechanism, the AS-MAMP algorithm has several limitations: its runtime should be optimised to allow for better performance in real-time simulations, as discussed in detail Section 5.2.1. The effectiveness of the decision-making on both local and global level should be compared to a baseline-model representing the current working procedures of ground controllers in detail.

Moreover, instead of centralised planning, other approaches could also be explored: for instance, a distributed planning method with merely high-level coordination during path planning, combined with local adjustments and decision-making during plan execution could be developed. Such an approach would mitigate the single point of failure that centralised methods pose, and minimise the computational runtime. Furthermore, short-term changes, uncertainties, and disruptions would likely be easier to counteract and deal with. Nonetheless, it needs to be assessed whether such distributed approaches can fulfil all the modelling requirements from Section 2.2, and would lead to adequately efficient and deadlock-free trajectories.

Suggested Research Direction

Showcase the ability of the MAS model to generalise to the large variability in airport layouts and sizes, operational contexts and concepts.

As noted in the list of assumptions underlying this thesis (Section 1.3), the fast-time simulations are conducted either on a synthetic layout based on potential bottlenecks in airport taxiways, or on a layout based on Amsterdam Airport Schiphol. Despite showing promising results on these layouts, the question remains whether the MAS model generalises to the large variability of taxiway networks of airports with different operational contexts and needs. Furthermore, while the MAS model proved useful in evaluating novel concepts of operations, there are many aspects of airport operations not yet considered in the model that could be explored. Following from its modular design, the MAS model should be extended with the relevant features and decision-making capabilities needed to run the respective studies.

Suggested Research Direction

Carry out an in-depth validation through human-in-the-loop experiments, and subsequently incorporate cognitive and behavioural models of human operators into the agent-based framework of human–automation interactions.

Throughout this thesis, the strong influence is acknowledged that human factors such as workload, stress, situation awareness, experience levels, and potential biases have on real-world decision-making processes. To this end, the MAS model, together with the explored agent-based framework for human–automation interactions, should be further validated through human-in-the-loop experiments. Such studies should assess the efficacy of the interactions explored in this thesis, and subsequently address follow-up questions like: which level of automation in airport surface movement operations and its specific tasks is desirable? Which allocation of tasks between humans and automated systems most effectively ensures safe and efficient operations, while retaining effective oversight, active involvement, and sustained job satisfaction of human operators?

One direction for future work is to develop cognitive and behavioural models of the human operators in ASM Ops, and to incorporate these into the agent-based framework for human–automation interactions. To illustrate this point, the conceived Interaction Agent introduced in Section 9.1 could use such models to react to the mental state of humans, and provide better and individualised assistance to them. In turn, the human operators might

remain more engaged in the operations, and their specific skills could be preserved and developed further. Such research could pave the way towards effective human-autonomy teaming in future concepts of ASM Ops.

Suggested Research Direction

Address the real-world challenges of the automated and sustainable airport surface movement operations studied in this thesis to pave the way towards deployment of such next-generation concepts of operations and decision-support tools in real-world airport operations.

The outcomes of any type of research should ideally end up being used in practice. Following from the work presented in this thesis, the MAS model could eventually serve as the basis for a control system or decision-support tools deployed in real-world airport operations. However, such a systemic (r)evolution will not happen overnight – a long transition period with many challenges to investigate and to resolve will ultimately be necessary. Human operators need to be satisfied with the solution to result in a meaningful advancement. Moreover, any liability concerns need to be resolved, certification is required, and sufficient training for trust in and ease of use of the system will be necessary.

The above list of future research directions could be extended further with many questions following from the work presented in this thesis. Therefore, the goal of this thesis is also fulfilled when researchers, industry stakeholders, and policy makers feel inspired by the results obtained and left with food for thought for next-generation concepts of automated and sustainable airport surface movement operations.

Abbreviations

AC	Aircraft
A-CDM	Airport Collaborative Decision Making
ADS-B	Automatic Dependent Surveillance –- Broadcast
AEON	SESAR 3 JU project 'Advanced Engine Off Navigation'
AI	Artificial Intelligence
ALDT	Actual Landing Time
AOBT	Actual Off-Block Time
ARR	Arriving flights
AS-MAMP	Multi-Agent Motion Planning on Airport Surfaces algorithm
A-SMGCS	Advanced Surface Movement Guidance and Control System
ASM Ops	Airport Surface Movement Operations
ASTAIR	SESAR 3 JU project 'Auto Steer Taxi at AIRport'
ATC	Air Traffic Control
ATCO	Air Traffic Control Officer
ATOT	Actual Take-Off Time
ConOps	Concept of Operations
CTOT	Calculated Take-Off Time
DEP	Departing flights
EASA	European Union Aviation Safety Agency
EHAM	Amsterdam Airport Schiphol (ICAO airport code)
FCFS	First-Come-First-Served principle
FMI	Feasible Motion Interval
FS	Flight Schedule
GPBS	Greedy Priority-Based Search algorithm
HAT	Human–Autonomy Teaming
HMI	Human–Machine Interface
IC	Induced Constraints
ICAO	International Civil Aviation Organization
IQR	Interquartile Range

KPI	Key Performance Indicator
KS	Kolmogorov-Smirnov test
MAMP	Multi-Agent Motion Planning
MAPF	Multi-Agent Path Finding
MAPP	Multi-Agent Path Planning
MAS	Multi-Agent System
MET	Multi-Engine Taxiing
ML/RL	Machine-Learning / Reinforcement-Learning
MTT	Minimum Turn-around Time
OECD	Organisation for Economic Co-operation and Development
PBS	Priority-Based Search algorithm
PBT	Pushback Tractor
RECAT-EU	European Wake Turbulence Categorisation and Separation Minima on Approach and Departure
RET	Rapid-Exit Taxiway
RMO	Runway Mode of Operation, i.e. runway configuration
RQ	Research Question
RWY	Runway
SAMP	Single-Agent Motion Planning
SAPF	Single-Agent Path Finding
SI	Safe Interval
SID	Standard Instrument Departure
SIMP	Safe Interval Motion Planning algorithm
SIPP	Safe Interval Path Planning algorithm
STBO	Surface Trajectory-Based Operations
TET	Tug-Enabled Taxiing
TOBT	Target Off-Block Time
TWY	Taxiway
USI	Unsafe Interval
WTC	Wake Turbulence Category

List of Figures

1.1 Overview of the airport environment . . . 3
1.2 Overview of different engine-off taxiing concepts . . . 4
1.3 Overview of the TaxiBot concept . . . 4
1.4 EASA's classification of AI applications . . . 5
1.5 Overview of an AI system and its elements . . . 6
1.6 Thesis outline with reading paths . . . 16

2.1 Overview of modelling requirements to study new ConOps for ASM Ops in detail 22

3.1 Generalised hierarchical-distributed multi-agent system modelling architecture to study divers ConOps for next-generation ASM Ops . . . 33
3.2 Multi-agent system model for fully-automated ASM Ops with centralised planning and distributed plan execution . . . 36
3.3 Representation of taxiway networks as graph . . . 38

4.1 Overview of AS-MAMP algorithm . . . 45
4.2 Exemplary trajectories through layout and state space . . . 48
4.3 Exemplary motions . . . 53
4.4 Definition of shapes and safety distances . . . 55
4.5 Graph reservations and exemplary conflicts . . . 57
4.6 Example of successor generation . . . 64

5.1 Synthetic airport layout with potential bottlenecks . . . 69
5.2 Simulation snapshots of test cases on synthetic layout . . . 70
5.3 Sensitivity analysis of acceleration/deceleration . . . 75
5.4 AS-MAMP computational performance on synthetic layout . . . 78
5.5 Taxi time distributions of MET operations on EHAM layout . . . 79
5.6 Taxi time distributions of TET operations on EHAM layout . . . 80
5.7 Taxi time distributions of AS-MAMP high-level variants on EHAM layout . . . 80
5.8 AS-MAMP computational performance on EHAM layout . . . 80
5.9 Comparison of FCFS to depth-first search for continuous schedules . . . 82
5.10 Simulation snapshots of continuous schedules . . . 82

6.1 Overview of multi-agent system for fully-automated airport surface movement operations . . . 93
6.2 Layout graph of Amsterdam Airport Schiphol . . . 93
6.3 Activity sequences . . . 95
6.4 Calibration of longitudinal acceleration and deceleration values with historical track data . . . 99

6.5 Calibration of curve speed with historical track dat . . . 99
6.6 Exemplary historical, un-calibrated, and calibrated speed profiles of a departing flight . . . 100
6.7 RMO North and RMO South at Schiphol . . . 102
6.8 Hourly count of flights over the two days, with shades denoting the RMO phase 105
6.9 Aircraft positions at historical A-CDM milestones for inbound and outbound flights . . . 107
6.10 Box-and-whisker plot of historical and simulated taxi times for arrivals and departures per runway-strip . . . 107
6.11 Predictability of taxi times . . . 109
6.12 Runtimes of planning rounds over the number of agents . . . 109
6.13 Additional taxi time per layout location on each of the two days . . . 110
6.14 Box-and-whisker plot of historical and simulated taxi times dependent on the hold-type of arrivals and departures per runway-strip . . . 111
6.15 Compliance to CTOT-slot of historical and simulated operations . . . 111
6.16 Comparison of exemplary takeoff order at runway 36L between historical and simulated operations . . . 112
6.17 Remaining takeoff slots for ICAO-C aircraft per day for both historical and simulated operations . . . 112

7.1 Overview of multi-agent system for automated multi-engine and tug-enabled taxiing operations . . . 118
7.2 Layout graph of Amsterdam Airport Schiphol for TET operations . . . 119
7.3 Activity sequence for arriving and departing aircraft . . . 120
7.4 Hourly count of flights over the two days, with shades denoting the RMO phase 123
7.5 Box-and-whisker plot of historical and simulated taxi times for arrivals and departures per runway-strip . . . 124
7.6 Delay hotspots depicted through additional taxi time per layout location for each day and each taxiing mode of operation . . . 126
7.7 Emission hotspots depicted through accumulated engine on time per layout location for departing aircraft for each day and each taxiing mode of operation . 127
7.8 Box-and-whisker plot of engine-on times of departing flights per runway-strip . 128

8.1 Illustration of methodology for evaluating historical track data . . . 136
8.2 Projection KPIs of exemplary track of departing flight shown in Figure **8.1** . . . 137
8.3 Overview of taxiway network of Amsterdam Airport Schiphol with shaded taxiway areas . . . 139
8.4 Historical speed data per aircraft type for both arriving and departing flights across the taxiway areas . . . 140
8.5 Distributions per aircraft type of historical lateral accelerations in curved taxiway segments for both arriving and departing flights . . . 141
8.6 Historical landing speeds per aircraft type and type of runway exit . . . 142
8.7 Share of takeoff categories of historical flights per aircraft type and type of runway entry . . . 143
8.8 Temporal separation of historical takeoffs grouped by ICAO wake turbulence categories . . . 144
8.9 Historical stand use and non-compliance rate to standard pushback path per stand 145

8.10 Hourly ground movements over the simulated 48 h, with shades denoting the RMO phase . . . 146
8.11 Box-and-whisker plots of taxi time distributions of historical A-CDM milestones, historical tracks, and simulated operations for arriving and departing flights per runway-strip . . . 147
8.12 Box-and-whisker plots of taxi distance distributions of historical tracks and simulated operations for arriving and departing flights per runway . . . 148
8.13 Comparison of average taxi speeds across the different taxiway areas for both arriving and departing flights . . . 149
8.14 Inbound holding of historical and simulated arriving flights . . . 151
8.15 Outbound holding of historical and simulated departing flights . . . 151
8.16 Comparison of occupancy rates in different areas of the airport between historical and simulated operations . . . 153
8.17 Hourly ground movements over the simulated 48 h, with shades denoting the RMO phase . . . 160
8.18 Required parking stands at buffer platform over time . . . 160
8.19 Normalised taxi time differences δt_{taxi} in relation to simulation 'no towing' . . . 160
8.20 Comparison of occupancy rates in different areas of the airport across the simulated towing operations from Table 8.7 . . . 162
8.21 Temporal separation of consecutive takeoffs in historical data, grouped by RECAT-EU categories . . . 164
8.22 Box-and-whisker plots of taxi time distributions for arriving and departing flights per runway-strip . . . 165
8.23 Comparison of occupancy rates in different areas of the airport across the simulated model variations . . . 166

9.1 Schematic agent-based framework of human–automation interactions for airport surface movement operations . . . 173
9.2 Overview of interaction tools for co-design of routes and decision-support provided by the Interaction Agent . . . 175
9.3 Illustration of fast-forward simulation . . . 177
9.4 Standard taxiway flow directions as part of ATC procedures at Amsterdam Airport Schiphol . . . 178
9.5 Ratio of non-standard taxiway traversals w.r.t. all traversals for two operational days under varying runway configurations . . . 180
9.6 Comparison of taxi times per runway strip between historical operations and simulations with varying values of the cost factor c_{TWY} . . . 180
9.7 Fast-forward animations of case 'Alternative Pushback Path – Part 1' . . . 183
9.8 Fast-forward animations of case 'Alternative Pushback Path – Part 2' . . . 184
9.9 SAMP routes to deal with unavailable stand of an arriving aircraft . . . 185
9.10 Rerouting of aircraft that declared an emergency . . . 187

10.1 Generalised hierarchical-distributed multi-agent system modelling architecture to study divers ConOps for next-generation ASM Ops . . . 195

List of Tables

2.1 Algorithmic properties of route optimisation techniques as backbone for a high-fidelity system model of ASM Ops . . . 29

3.1 Categorisation of aircraft sizes into ICAO-types . . . 41
3.2 Minimum wake turbulence separation times . . . 42

5.1 Flight schedules for sensitivity analyses on synthetic layout . . . 70
5.2 Default values for the key parameters of AS-MAMP . . . 71
5.3 AS-MAMP high-level benchmark . . . 72
5.4 AS-MAMP low-level benchmark . . . 74
5.5 Extended flight schedules . . . 76
5.6 Maximal runway throughputs and occupancy rates . . . 83

6.1 Kinematic and algorithm parameters that are used in the routing algorithm . . . 99
6.2 Overview of (A) traffic data and (B) parameters and daily counts of ICAO-types 102
6.3 Comparison of historical and simulated operations with $w_p = 20\,\text{min}$, $h_p = 10\,\text{min}$ . . . 105

7.1 Overview of operational data . . . 122
7.2 Vehicle types: name, shape diameter, wake turbulence category (WTC) for aircraft, and count in operational data . . . 122
7.3 Kinematic properties of Moving Agents and algorithm parameters that are used in the routing algorithm . . . 122
7.4 KPIs of historical, MET, and TET operations . . . 124

8.1 Maximum speeds v_{max} in [m/s] based on reference percentiles P_{ref} and historical speed medians $\tilde{v}_{hist}$ per aircraft type and different taxiway areas . . . 140
8.2 Maximum lateral accelerations a_{lat} in [m/s²] based on reference percentiles P_{ref} and historical medians $\tilde{a}_{lat,hist}$ per aircraft type to calculate curve speed limits . . . 141
8.3 Time-based separations between takeoffs in [s] . . . 144
8.4 Similarity between historical and simulated operations for arriving and departing flights . . . 148
8.5 Aircraft towing movements reconstructed from the 48 h of historical flight schedule data . . . 157
8.6 Minimum turn-around times in [min] to determine the towing schedule . . . 158
8.7 Overview of performed simulations to assess the impact of aircraft towing movements . . . 159
8.8 Overview of simulated model variations . . . 163

References

[1] IATA. 'Net-zero carbon emissions by 2050', Press Release No. 66. (2021), [Online]. Available: https://www.iata.org/en/pressroom/pressroom-archive/2021-releases/2021-10-04-03/ (visited on 04/04/2023).

[2] UNFCCC, 'Paris Agreement', 2015. [Online]. Available: https://unfccc.int/sites/default/files/english_paris_agreement.pdf (visited on 17/05/2023).

[3] UNFCCC, 'Global mutirão: Uniting humanity in a global mobilization against climate change', Draft decisions FCCC/PA/CMA/2025/L.24, 2025.

[4] SKYbrary. 'Ground operations: Ground collision'. (2024), [Online]. Available: https://skybrary.aero/articles/ground-collision (visited on 23/06/2024).

[5] SKYbrary. 'Accident and serious incident reports: Runway incursion'. (2024), [Online]. Available: https://skybrary.aero/articles/accident-and-serious-incident-reports-ri (visited on 23/06/2024).

[6] R. Morris, C. S. Pasareanu, K. Luckow *et al.*, 'Planning, scheduling and monitoring for airport surface operations', *AAAI-16 Workshop on Planning for Hybrid Systems*, 2016.

[7] D. Teodorović and M. Janić, 'Capacity and Level of Service', in *Transportation Engineering*, Elsevier, 2017, pp. 197–292, ISBN: 978-0-323-90813-9. DOI: 10.1016/B978-0-323-90813-9.00005-9.

[8] EUROCONTROL, *European aviation in 2040 - challenges of growth*, Statistics and Forecast Service, 2018. [Online]. Available: https://www.eurocontrol.int/sites/default/files/2019-07/challenges-of-growth-2018-annex1_0.pdf (visited on 14/12/2021).

[9] Z. Chua, M. Cousy, M. Causse and F. Lancelot, 'Initial assessment of the impact of modern taxiing techniques on airport ground control', in *Proceedings of HCI-Aero 2016*, Paris France: ACM, 2017, pp. 1–8, ISBN: 978-1-4503-4406-7. DOI: 10.1145/2950112.2964589.

[10] EUROCONTROL, *Data snapshot #22 on lower summer taxi-out times*, 2021. [Online]. Available: https://www.eurocontrol.int/sites/default/files/2021-11/eurocontrol-data-snapshot-22-taxi-time-affect-ops.pdf (visited on 30/11/2021).

[11] D. Fordham, M. Stephens, A. Chymiy *et al.*, *Deriving Benefits from Alternative Aircraft-Taxi Systems*. Washington, D.C.: Transportation Research Board, 2016, p. 23 618, ISBN: 978-0-309-44845-1. DOI: 10.17226/23618.

[12] 'TaxiBot concept', TaxiBot International. (2023), [Online]. Available: https://taxibot-international.com/concept/ (visited on 07/02/2024).

[13] EUROCONTROL, *Data snapshot #40 on taxi times*, 2023. [Online]. Available: https://www.eurocontrol.int/sites/default/files/2023-03/eurocontrol-data-snapshot-40.pdf (visited on 04/04/2023).

[14] B. Rowland. 'Which part of a flight uses the most fuel?', Aviation Sustainability. (2022), [Online]. Available: https://www.oag.com/blog/which-part-flight-uses-most-fuel (visited on 05/04/2023).

[15] M. Lukic, P. Giangrande, A. Hebala, S. Nuzzo and M. Galea, 'Review, challenges, and future developments of electric taxiing systems', *IEEE Transactions on Transportation Electrification*, vol. 5, no. 4, pp. 1441–1457, 2019, ISSN: 2332-7782. DOI: 10.1109/TTE.2019.2956862.

[16] M. Zoutendijk, M. Mitici and J. M. Hoekstra, 'An investigation of operational management solutions and challenges for electric taxiing of aircraft', *Research in Transportation Business & Management*, vol. 49, p. 101 019, 2023, ISSN: 2210-5395. DOI: 10.1016/j.rtbm.2023.101019.

[17] ACAAF-TF, *Concept of operations of battery and hydrogen-powered aircraft at aerodromes*, Airport Compatibility of Alternative Aviation Fuels Task Force (ACAAF-TF), 2025. [Online]. Available: https://www.iata.org/globalassets/iata/publications/sustainability/concept-of-operations-of-battery-and-hydrogen-powered-aircraft-at-aerodromes.pdf.

[18] EASA, *AI roadmap 2.0: Human-centric approach to AI in aviation*, 2023. [Online]. Available: easa.europa.eu/ai.

[19] OECD, *Artificial Intelligence in Society*. OECD Publishing, 2019. DOI: 10.1787/eedfee77-en.

[20] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach* (Pearson Series in Artificial Intelligence), Fourth edition, global edition. Harlow: Pearson, 2021, 1166 pp., ISBN: 978-1-292-40113-3.

[21] L. Bainbridge, 'Ironies of automation', *Automatica*, vol. 19, no. 6, pp. 775–779, 1983, ISSN: 0005-1098. DOI: 10.1016/0005-1098(83)90046-8.

[22] M. R. Endsley, 'Ironies of artificial intelligence', *Ergonomics*, vol. 66, no. 11, pp. 1656–1668, 2023, ISSN: 0014-0139. DOI: 10.1080/00140139.2023.2243404.

[23] J. B. Lyons, K. Sycara, M. Lewis and A. Capiola, 'Human–Autonomy Teaming: Definitions, Debates, and Directions', *Frontiers in Psychology*, vol. 12, 2021, ISSN: 1664-1078. DOI: 10.3389/fpsyg.2021.589585.

[24] T. O'Neill, N. McNeese, A. Barron and B. Schelble, 'Human–Autonomy Teaming: A Review and Analysis of the Empirical Literature', *Human Factors*, vol. 64, no. 5, pp. 904–938, 2022, ISSN: 0018-7208. DOI: 10.1177/0018720820960865.

[25] R. Parasuraman and V. Riley, 'Humans and Automation: Use, Misuse, Disuse, Abuse', *Human Factors*, vol. 39, no. 2, pp. 230–253, 1997, ISSN: 0018-7208. DOI: 10.1518/001872097778543886.

[26] T. B. Sheridan and R. Parasuraman, 'Human-Automation Interaction', *Reviews of Human Factors and Ergonomics*, vol. 1, no. 1, pp. 89–129, 2005, ISSN: 1557-234X. DOI: 10.1518/155723405783703082.

[27] B. L. Hooey, V. H. L. Cheng and D. C. Foyle, 'A concept of operations for far-term surface trajectory-based operations (STBO)', NASA/TM-2014-218354, 2014. [Online]. Available: https://ntrs.nasa.gov/citations/20190002687 (visited on 13/05/2024).

[28] N. Okuniek, I. Gerdes, J. Jakobi *et al.*, 'A concept of operations for trajectory-based taxi operations', in *16th AIAA Aviation Technology, Integration, and Operations Conference*, Washington, D.C.: American Institute of Aeronautics and Astronautics, 2016, pp. 1–18, ISBN: 978-1-62410-440-4. DOI: 10.2514/6.2016-3753.

[29] N. Okuniek, Z. Zhu, Y. C. Jung, S. Gridnev, I. Gerdes and H. Lee, 'Performance evaluation of conflict-free trajectory taxiing in airport ramp area using fast-time simulations', in *2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC)*, London: IEEE, 2018, pp. 1–10, ISBN: 978-1-5386-4112-5. DOI: 10.1109/DASC.2018.8569228.

[30] T. Bernatzky, 'Automation concept for cockpit crew integration into trajectory-based dispatch towing', Ph.D. dissertation, TU Darmstadt, Germany, 2019. [Online]. Available: https://doi.org/10.26083/tuprints-00008751.

[31] M. Rieth and V. Hagemann, 'Automation as an equal team player for humans? A view into the field and implications for research and practice', *Applied Ergonomics*, vol. 98, no. 4, p. 103 552, 2022, ISSN: 00036870. DOI: 10.1016/j.apergo.2021.103552.

[32] Á. Câmara, D. C. Silva, E. Oliveira and P. H. Abreu, 'Comparing centralized and decentralized multi-agent approaches to air traffic control', in *Proceedings of the 28th European Simulation and Modelling Conference (ESM-2014)*, 2014, pp. 189–193.

[33] H. Udluft, 'Decentralization in air transportation', Ph.D. dissertation, TU Delft, 2017. [Online]. Available: https://doi.org/10.4233/uuid:2af25aa3-be84-4d37-9332-fe2af319db1e.

[34] AEON. 'AEON - Advanced Engine Off Navigation'. (2021), [Online]. Available: https://www.aeon-project.eu/ (visited on 25/09/2023).

[35] ASTAIR. 'ASTAIR - Auto Steer Taxi at AIRport'. (2023), [Online]. Available: https://research.dblue.it/astair/ (visited on 20/02/2025).

[36] P. Lanzi, E. Spiller, S. Gottofredi *et al.*, 'D1.2 concept of operations final version', AEON, project deliverable D1.2, 2022.

[37] S. van Oosterom and M. Mitici, 'D2.2 model for optimal allocation of towing vehicles', AEON, project deliverable D2.2, 2022.

[38] M. von der Burg, A. Sharpanskykh and J. Kamphof, 'D2.1 multi-agent system for routing', AEON, project deliverable D2.1, 2022.

[39] J. Garcia, D.-B. Vo and V. Peyruqueou, 'D3.2 supervision HMI description', AEON, project deliverable D3.2, 2022.

[40] AEON, 'D5.1 solution assessment plan', AEON, project deliverable D5.1, 2022.

[41] S. Gottofredi, P. Lanzi and E. Spiller, 'D5.2 human performance assessment report', AEON, project deliverable D5.2, 2022.

[42] A. Sharpanskykh, 'D5.3 safety assessment report', AEON, project deliverable D5.3, 2022.

[43] P. Roling and G. Ermis, 'D5.4 cost-benefit assessment report', AEON, project deliverable D5.4, 2022.

[44] ASTAIR, 'D1.3 initial concept outline', project deliverable D1.3, 2024.

[45] ASTAIR, 'D5.1 exploratory research plan (ERP)', project deliverable D5.1, 2024.

[46] D. Helbing, 'Agent-based modeling', in *Social Self-Organization: Agent-Based Simulations and Experiments to Study Emergent Social Behavior*, D. Helbing, Ed., Berlin, Heidelberg: Springer, 2012, pp. 25–70, ISBN: 978-3-642-24004-1. DOI: 10.1007/978-3-642-24004-1_2.

[47] C. G. Cena, P. F. Cardenas, R. S. Pazmino, L. Puglisi and R. A. Santonja, 'A cooperative multi-agent robotics system: Design and modelling', *Expert Systems with Applications*, vol. 40, no. 12, pp. 4737–4748, 2013, ISSN: 0957-4174. DOI: 10.1016/j.eswa.2013.01.048.

[48] J. Ota, 'Multi-agent robot systems as distributed autonomous systems', *Advanced Engineering Informatics*, vol. 20, no. 1, pp. 59–70, 2006, ISSN: 1474-0346. DOI: 10.1016/j.aei.2005.06.002.

[49] P. Iñigo-Blasco, F. Diaz-del-Rio, M. C. Romero-Ternero, D. Cagigas-Muñiz and S. Vicente-Diaz, 'Robotics software frameworks for multi-agent robotic systems development', *Robotics and Autonomous Systems*, vol. 60, no. 6, pp. 803–821, 2012, ISSN: 0921-8890. DOI: 10.1016/j.robot.2012.02.004.

[50] R. Domínguez, S. Cannella and J. M. Framinan, 'SCOPE: A multi-agent system tool for supply chain network analysis', in *IEEE EUROCON 2015 - International Conference on Computer as a Tool (EUROCON)*, 2015, pp. 1–5. DOI: 10.1109/EUROCON.2015.7313688.

[51] K. Hager, J. Rauh and W. Rid, 'Agent-based modeling of traffic behavior in growing metropolitan areas', *Transp. Res. Procedia*, 18th Euro Working Group on Transportation, EWGT 2015, 14-16 July 2015, Delft, The Netherlands, vol. 10, pp. 306–315, 2015, ISSN: 2352-1465. DOI: 10.1016/j.trpro.2015.09.080.

[52] O. Wangapisit, E. Taniguchi, J. S. E. Teo and A. G. Qureshi, 'Multi-agent systems modelling for evaluating joint delivery systems', *Procedia - Social and Behavioral Sciences*, Eighth International Conference on City Logistics 17-19 June 2013, Bali, Indonesia, vol. 125, pp. 472–483, 2014, ISSN: 1877-0428. DOI: 10.1016/j.sbspro.2014.01.1489.

[53] G. H. Merabet, M. Essaaidi, H. Talei *et al.*, 'Applications of multi-agent systems in smart grids: A survey', in *2014 International Conference on Multimedia Computing and Systems (ICMCS)*, 2014, pp. 1088–1094. DOI: 10.1109/ICMCS.2014.6911384.

[54] M. Gatti, P. Cavalin, S. B. Neto *et al.*, 'Large-scale multi-agent-based modeling and simulation of microblogging-based online social network', in *Multi-Agent-Based Simulation XIV*, S. J. Alam and H. V. D. Parunak, Eds., Berlin, Heidelberg: Springer, 2014, pp. 17–33, ISBN: 978-3-642-54783-6. DOI: 10.1007/978-3-642-54783-6_2.

[55] L. Ma and Y. Zhang, 'Hierarchical social network analysis using multi-agent systems: A school system case', in *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2014, pp. 1412–1419. DOI: 10.1109/SMC.2014.6974113.

[56] K. M. Sim, 'Agent-based cloud computing', *IEEE Transactions on Services Computing*, vol. 5, no. 4, pp. 564–577, 2012, ISSN: 1939-1374. DOI: 10.1109/TSC.2011.52.

[57] S. Shamshirband, N. B. Anuar, M. L. M. Kiah and A. Patel, 'An appraisal and design of a multi-agent system based cooperative wireless intrusion detection computational intelligence technique', *Engineering Applications of Artificial Intelligence*, vol. 26, no. 9, pp. 2105–2127, 2013, ISSN: 0952-1976. DOI: 10.1016/j.engappai.2013.04.010.

[58] M. J. Wooldridge, *An Introduction to Multiagent Systems*, 2nd ed. Chichester, U.K: John Wiley & Sons, 2009, 461 pp., ISBN: 978-0-470-51946-2.

[59] A. Dorri, S. S. Kanhere and R. Jurdak, 'Multi-agent systems: A survey', *IEEE Access*, vol. 6, pp. 28 573–28 593, 2018, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2018.2831228.

[60] M. Phillips and M. Likhachev, 'SIPP: Safe Interval Path Planning for dynamic environments', in *Proceedings of ICRA-11*, 2011, pp. 5628–5635. DOI: 10.1109/ICRA.2011.5980306.

[61] H. Ma, D. Harabor, P. J. Stuckey, J. Li and S. Koenig, 'Searching with consistent prioritization for multi-agent path finding', in *Proceedings of AAAI-19*, vol. 33, 2019, pp. 7643–7650.

[62] S.-H. Chan, R. Stern, A. Felner and S. Koenig, 'Greedy Priority-Based Search for suboptimal multi-agent path finding', *Proceedings of SoCS-23*, vol. 16, no. 1, pp. 11–19, 2023, ISSN: 2832-9163, 2832-9171. DOI: 10.1609/socs.v16i1.27278.

[63] ASTAIR, 'D1.2 workshops report', project deliverable D1.2, 2024. [Online]. Available: https://research.dblue.it/astair/deliverables/.

[64] J. Conde, A. Munoz-Arcentales, M. Romero *et al.*, 'Applying digital twins for the management of information in turnaround event operations in commercial airports', *Advanced Engineering Informatics*, vol. 54, p. 101 723, 2022, ISSN: 14740346. DOI: 10.1016/j.aei.2022.101723.

[65] S. Okwir, K. Amouzgar and A. H. Ng, 'Exploring prediction accuracy for optimal taxi times in airport operations using various machine learning models', *Journal of Air Transport Management*, vol. 122, p. 102 684, 2025, ISSN: 09696997. DOI: 10.1016/j.jairtraman.2024.102684.

[66] Schiphol. 'Deep turnaround'. (2026), [Online]. Available: https://www.schiphol.nl/en/aviation-solutions/deep-turnaround/ (visited on 08/01/2026).

[67] WiMAX Forum. 'Aeronautical Mobile Airport Communication System (AeroMACS)'. (2001), [Online]. Available: http://wimaxforum.org/Page/AeroMACS (visited on 08/01/2026).

[68] R. Lane, S. Dubuisson, M. Ellejmi and E. Cerasi, *EUROCONTROL specification for A-SMGCS services*, Eurocontrol Specification, version 2.0, 2020, Released Issue.

[69] ICAO, 'Aerodrome design and operations', in *Annex 14 to the Convention on International Civil Aviation*, ser. Aerodromes, 7th ed., vol. I, 2016.

[70] H. Lee, 'Airport surface traffic optimization and simulation in the presence of uncertainties', Ph.D. dissertation, Massachusetts Institute of Technology, 2014. [Online]. Available: http://hdl.handle.net/1721.1/87480 (visited on 06/05/2024).

[71] J. A. D. Atkin, E. K. Burke and S. Ravizza, 'The airport ground movement problem: Past and current research and future directions', in *4th International Conference on Research in Air Transportation (ICRAT)*, 2010, pp. 131–138.

[72] J. A. Bennell, M. Mesgarpour and C. N. Potts, 'Airport runway scheduling', *4OR*, vol. 9, no. 2, pp. 115–138, 2011, ISSN: 1614-2411. DOI: 10.1007/s10288-011-0172-x.

[73] A. Lieder, D. Briskorn and R. Stolletz, 'A dynamic programming approach for the aircraft landing problem with aircraft classes', *European Journal of Operational Research*, vol. 243, no. 1, pp. 61–69, 2015, ISSN: 0377-2217. DOI: 10.1016/j.ejor.2014.11.027.

[74] J. Guépet, O. Briant, J.-P. Gayon and R. Acuna-Agost, 'Integration of aircraft ground movements and runway operations', *Transp. Res. Part E: Logistics and Transportation Review*, vol. 104, pp. 131–149, 2017, ISSN: 13665545. DOI: 10.1016/j.tre.2017.05.002.

[75] H. Balakrishnan and Y. Jung, 'A framework for coordinated surface operations planning at Dallas-Fort Worth International Airport', in *AIAA Guidance, Navigation and Control Conference and Exhibit*, Hilton Head, South Carolina: American Institute of Aeronautics and Astronautics, 2007, p. 6553, ISBN: 978-1-62410-015-4. DOI: 10.2514/6.2007-6553.

[76] S. Ravizza, J. A. D. Atkin and E. K. Burke, 'A more realistic approach for airport ground movement optimisation with stand holding', *Journal of Scheduling*, vol. 17, no. 5, pp. 507–520, 2014, ISSN: 1099-1425. DOI: 10.1007/s10951-013-0323-3.

[77] J. Guépet, O. Briant, J. Gayon and R. Acuna-Agost, 'The aircraft ground routing problem: Analysis of industry punctuality indicators in a sustainable perspective', *European Journal of Operational Research*, vol. 248, no. 3, pp. 827–839, 2016, ISSN: 03772217. DOI: 10.1016/j.ejor.2015.08.041.

[78] A. Ashok, H. Balakrishnan and S. R. H. Barrett, 'Reducing the air quality and CO2 climate impacts of taxi and takeoff operations at airports', *Transp. Res. Part D: Transport and Environment*, vol. 54, pp. 287–303, 2017, ISSN: 1361-9209. DOI: 10.1016/j.trd.2017.05.013.

[79] J. Yin, M. Zhang, Y. Ma, W. Wu, H. Li and P. Chen, 'Prediction and analysis of airport surface taxi time: Classification, features, and methodology', *Applied Sciences*, vol. 14, no. 3, p. 1306, 3 2024, ISSN: 2076-3417. DOI: 10.3390/app14031306.

[80] J. Li, M. Gong, Z. Liang *et al.*, 'Scheduling and airport taxiway path planning under uncertainty', in *AIAA Aviation 2019 Forum*, ser. AIAA AVIATION Forum, 2019, pp. 1–8. DOI: 10.2514/6.2019-2930.

[81] X. Wang, A. E. I. Brownlee, M. Weiszer, J. R. Woodward, M. Mahfouf and J. Chen, 'A chance-constrained programming model for airport ground movement optimisation with taxi time uncertainties', *Transp. Res. Part C: Emerging Technologies*, vol. 132, p. 103 382, 2021, ISSN: 0968-090X. DOI: 10.1016/j.trc.2021.103382.

[82] A. E. I. Brownlee, M. Weiszer, J. Chen, S. Ravizza, J. R. Woodward and E. K. Burke, 'A fuzzy approach to addressing uncertainty in airport ground movement optimisation', *Transp. Res. Part C: Emerging Technologies*, vol. 92, pp. 150–175, 2018, ISSN: 0968-090X. DOI: 10.1016/j.trc.2018.04.020.

[83] Z. Wang, M. Liang and D. Delahaye, 'Automated data-driven prediction on aircraft estimated time of arrival', in *SESAR Innovation Days 2018*, 2018, pp. 1–8. DOI: 10.1016/j.jairtraman.2020.101840.

[84] I. Dhief, Z. Wang, M. Liang, S. Alam, M. Schultz and D. Delahaye, 'Predicting aircraft landing time in extended-TMA using machine learning methods', in *Proceedings of ICRAT-20*, 2020, hal–02 907 597.

[85] C. J. Woo, S. K. Goh, S. Alam, M. Ferdaus and M. Ellejmi, 'A runway exit prediction model with visually explainable machine decisions', in *Proceedings of ICRAT-22*, 2022, pp. 1–9.

[86] H. Ali, D.-T. Pham, S. Alam and M. Schultz, 'A deep reinforcement learning approach for airport departure metering under spatial–temporal airside interactions', *IEEE Transactions on Intel. Transp. Sys.*, vol. 23, no. 12, pp. 23 933–23 950, 2022, ISSN: 1524-9050, 1558-0016. DOI: 10.1109/TITS.2022.3209397.

[87] Y. Jung, T. Hoang, J. Montoya *et al.*, 'Performance evaluation of a surface traffic management tool for Dallas/Fort Worth International Airport', in *Proceedings of ATM2011*, 2011.

[88] M. Hayashi, T. Hoang, Y. C. Jung, W. Malik, H. Lee and V. L. Dulchinos, 'Evaluation of pushback decision-support tool concept for Charlotte Douglas International Airport ramp operations', in *Proceedings of ATM2015*, Lisbon, 2015. [Online]. Available: https://ntrs.nasa.gov/citations/20160008912 (visited on 18/06/2024).

[89] D. L. Bakowski, B. L. Hooey, D. C. Foyle and C. A. Wolter, 'NextGen surface trajectory-based operations (STBO): Evaluating conformance to a four-dimensional trajectory (4DT)', *Procedia Manufacturing*, vol. 3, pp. 2458–2465, 2015, ISSN: 23519789. DOI: 10.1016/j.promfg.2015.07.506.

[90] D. L. Bakowski, B. L. Hooey and D. C. Foyle, 'Flight deck surface trajectory-based operations (STBO): A four-dimensional trajectory (4DT) simulation', in *2017 IEEE/AIAA 36th Digital Avionics Systems Conference (DASC)*, 2017, pp. 1–10. DOI: 10.1109/DASC.2017.8102008.

[91] EUROCONTROL, *A-CDM impact assessment: Final report*, 2016. [Online]. Available: https : / / www . eurocontrol.int/sites/default/files/2019-04/a-cdm-impact-assessment-2016.pdf.

[92] ICAO, *Advanced Surface Guidance and Control Systems (A-SMGCS) manual*, Doc 9830, version 1, 2004, published.

[93] I. Gerdes and M. Schaper, 'Management of time based taxi trajectories coupling departure and surface management systems', in *Proceedings of ATM2015*, 2015.

[94] H. Ma, S. Koenig, N. Ayanian *et al.*, 'Overview: Generalizations of multi-agent path finding to real-world scenarios', in *IJCAI-16 Workshop on Multi-Agent Path Finding (WoMAPF)*, 2016, pp. 1–5.

[95] L. Cohen, T. Uras, T. K. Kumar and S. Koenig, 'Optimal and bounded-suboptimal multi-agent motion planning', in *Proceedings of SoCS-19*, 2019, pp. 44–51.

[96] T. T. Walker, 'Multi-agent pathfinding in mixed discrete-continuous time and space', Ph.D. dissertation, University of Denver, 2022.

[97] X. Zou, P. Cheng, W. Liu, N. Cheng and J. Zhang, 'A two-stage taxi scheduling strategy at airports with multiple independent runways', *Transp. Res. Part C: Emerging Technologies*, vol. 95, pp. 165–184, 2018, ISSN: 0968090X. DOI: 10.1016/j.trc.2018.07.005.

[98] Y. Jiang, Z. Liu, Z. Hu and H. Zhang, 'A priority-based conflict resolution strategy for airport surface traffic considering suboptimal alternative paths', *IEEE Access*, vol. 9, pp. 606–617, 2021, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.3045586.

[99] S. Zaninotto, J. Gauci and B. Zammit, 'A testbed for performance analysis of algorithms for engineless taxiing with autonomous tow trucks', in *2021 IEEE/AIAA 40th Digital Avionics Systems Conference (DASC)*, 2021, pp. 1–9. DOI: 10.1109/DASC52595.2021.9594411.

[100] R. Stern, 'Multi-agent path finding – an overview', in *Artificial Intelligence: 5th RAAI Summer School*, ser. Lecture Notes in Computer Science, G. Osipov, A. Panov and K. Yakovlev, Eds., Cham, 2019, pp. 96–115, ISBN: 978-3-030-33274-7. DOI: 10.1007/978-3-030-33274-7_6.

[101] H. Ma, T. K. Kumar and S. Koenig, 'Multi-agent path finding with delay probabilities', in *Proceedings of AAAI-17*, vol. 31, 2017, pp. 3605–3612.

[102] R. Stern and N. R. Sturtevant, 'Multi-agent pathfinding: Definitions, variants, and benchmarks', in *Proceedings of SoCS-19*, 2019, pp. 151–158.

[103] A. Felner, R. Stern, S. E. Shimony *et al.*, 'Search-based optimal solvers for the multi-agent pathfinding problem: Summary and challenges', in *Proceedings of SoCS-17*, 2017, pp. 29–37.

[104] J. Gao, Y. Li, X. Li, K. Yan, K. Lin and X. Wu, 'A review of graph-based multi-agent pathfinding solvers: From classical to beyond classical', *Knowledge-Based Systems*, vol. 283, p. 111 121, 2024, ISSN: 09507051. DOI: 10.1016/j.knosys.2023.111121.

[105] G. Wagner and H. Choset, 'M*: A complete multirobot path planning algorithm with performance bounds', in *Proceedings of IROS-11*, 2011, pp. 3260–3267.

[106] G. Sharon, R. Stern, M. Goldenberg and A. Felner, 'The increasing cost tree search for optimal multi-agent pathfinding', *Artificial Intelligence*, vol. 195, pp. 470–495, 2013, ISSN: 0004-3702. DOI: 10.1016/j.artint.2012.11.006.

[107] G. Sharon, R. Stern, A. Felner and N. R. Sturtevant, 'Conflict-Based Search for optimal multi-agent path finding', in *Proceedings of AAAI-12*, vol. 26, 2012, pp. 563–569.

[108] M. Barer, G. Sharon, R. Stern and A. Felner, 'Suboptimal variants of the Conflict-Based Search algorithm for the multi-agent pathfinding problem', in *Proceedings of SoCS-14*, 2014, pp. 19–27.

[109] L. Cohen, T. Uras, T. K. Kumar, H. Xu, N. Ayanian and S. Koenig, 'Improved solvers for bounded-suboptimal multi-agent path finding.', in *Proceedings of IJCAI-16*, 2016, pp. 3067–3074.

[110] J. Li, W. Ruml and S. Koenig, 'EECBS: A bounded-suboptimal search for multi-agent path finding', *Proceedings of AAAI-2021*, vol. 35, no. 14, pp. 12 353–12 362, 2021, ISSN: 2374-3468, 2159-5399. DOI: 10.1609/aaai.v35i14.17466.

[111] J. Li, P. Surynek, A. Felner, H. Ma, T. K. Kumar and S. Koenig, 'Multi-agent path finding for large agents', in *Proceedings of AAAI-19*, vol. 33, 2019, pp. 7627–7634. DOI: 10.1609/aaai.v33i01.33017627.

[112] T. T. Walker, N. R. Sturtevant and A. Felner, 'Generalized and sub-optimal bipartite constraints for Conflict-Based Search', *Proceedings of AAAI-2020*, vol. 34, no. 05, pp. 7277–7284, 2020, ISSN: 2374-3468, 2159-5399. DOI: 10.1609/aaai.v34i05.6219.

[113] J. Li, T. A. Hoang, E. Lin, H. L. Vu and S. Koenig, 'Intersection coordination with Priority-Based Search for autonomous vehicles', in *Proceedings of AAAI-23*, vol. 37, 2023, pp. 11 578–11 585. DOI: 10.1609/aaai.v37i10.26368.

[114] Y. Zheng, H. Ma, S. Koenig, E. Kline and T. K. Kumar, 'Priority-Based Search for the virtual network embedding problem', in *Proceedings of ICAPS-23*, vol. 33, 2023, pp. 472–480. DOI: 10.1609/icaps.v33i1.27227.

[115] D. Silver, 'Cooperative pathfinding', in *Proceedings of the Artificial Intelligence and Interactive Digital Entertainment Conference (AIIDE-2005)*, 2005, pp. 117–122.

[116] J. Li, A. Tinka, S. Kiesel, J. W. Durham, T. K. Kumar and S. Koenig, 'Lifelong multi-agent path finding in large-scale warehouses', in *Proceedings of AAAI-21*, vol. 35, 2021, pp. 11 272–11 281.

[117] J. Švancara, M. Vlk, R. Stern, D. Atzmon and R. Barták, 'Online multi-agent pathfinding', in *Proceedings of AAAI-19*, vol. 33, 2019, pp. 7732–7739. DOI: 10.1609/aaai.v33i01.33017732.

[118] T. T. Walker, N. R. Sturtevant and A. Felner, 'Extended Increasing Cost Tree Search for non-unit cost domains', in *Proceedings of IJCAI-18*, Stockholm, Sweden: International Joint Conferences on Artificial Intelligence Organization, 2018, pp. 534–540, ISBN: 978-0-9992411-2-7. DOI: 10.24963/ijcai.2018/74.

[119] H. Ma, G. Wagner, A. Felner, J. Li, T. K. Kumar and S. Koenig, 'Multi-agent path finding with deadlines', in *Proceedings of IJCAI-18*, Stockholm, Sweden: International Joint Conferences on Artificial Intelligence Organization, 2018, pp. 417–423, ISBN: 978-0-9992411-2-7. DOI: 10.24963/ijcai.2018/58.

[120] K. Yakovlev, A. Andreychuk and V. Vorobyev, 'Prioritized multi-agent path finding for differential drive robots', in *2019 European Conference on Mobile Robots (ECMR)*, Prague, Czech Republic: IEEE, 2019, pp. 1–6, ISBN: 978-1-7281-3605-9. DOI: 10.1109/ECMR.2019.8870957.

[121] W. Hoenig, T. K. Kumar, L. Cohen *et al.*, 'Multi-agent path finding with kinematic constraints', in *Proceedings of ICAPS-16*, 2016, pp. 477–485.

[122] D. González, J. Pérez, V. Milanés and F. Nashashibi, 'A review of motion planning techniques for automated vehicles', *IEEE Transactions on Intel. Transp. Sys.*, vol. 17, no. 4, pp. 1135–1145, 2016, ISSN: 1558-0016. DOI: 10.1109/TITS.2015.2498841.

[123] K. Yakovlev and A. Andreychuk, 'Any-angle pathfinding for multiple agents based on SIPP algorithm', in *Proceedings of ICAPS-17*, vol. 27, 2017, pp. 586–594.

[124] K. Yakovlev and A. Andreychuk, 'Towards time-optimal any-angle path planning with dynamic obstacles', *Proceedings of ICAPS 2021*, vol. 31, pp. 405–414, 2021, ISSN: 2334-0843, 2334-0835. DOI: 10.1609/icaps.v31i1.15986.

[125] Y. Zou and C. Borst, 'Zeta*-SIPP: Improved time-optimal any-angle Safe-Interval Path Planning', in *Proceedings of IJCAI-24*, vol. 7, 2024, pp. 6823–6830. DOI: 10.24963/ijcai.2024/754.

[126] A. Andreychuk, K. Yakovlev, D. Atzmon and R. Stern, 'Multi-agent pathfinding (MAPF) with continuous time', in *Proceedings of IJCAI-19*, 2019, pp. 39–45.

[127] A. Andreychuk, K. Yakovlev, E. Boyarski and R. Stern, 'Improving Continuous-time Conflict Based Search', *Proceedings of AAAI-2021*, vol. 35, no. 13, pp. 11 220–11 227, 2021, ISSN: 2374-3468. DOI: 10.1609/aaai.v35i13.17338.

[128] H. Ma, W. Hönig, T. K. Kumar, N. Ayanian and S. Koenig, 'Lifelong path planning with kinematic constraints for multi-agent pickup and delivery', in *Proceedings of AAAI-19*, vol. 33, 2019, pp. 7651–7658.

[129] J. Li, Z. Chen, D. Harabor, P. J. Stuckey and S. Koenig, 'MAPF-LNS2: Fast repairing for multi-agent path finding via large neighborhood search', *Proceedings of AAAI-2022*, vol. 36, no. 9, pp. 10 256–10 265, 2022, ISSN: 2374-3468, 2159-5399. DOI: 10.1609/aaai.v36i9.21266.

[130] Z. A. Ali and K. Yakovlev, 'Safe Interval Path Planning with kinodynamic constraints', *Proceedings of AAAI-23*, 10th ser., vol. 37, no. 10, pp. 12 330–12 337, 2023. DOI: 10.1609/aaai.v37i10.26453.

[131] M. von der Burg and A. Sharpanskykh, 'Studying the operational consequences of automated engine-off taxiing using multi-agent planning: Explorative case study at Amsterdam Airport Schiphol', in *Proceedings of ICRAT-24*, Singapore, 2024, pp. 1–8.

[132] M. von der Burg and A. Sharpanskykh, 'Modelling and analysis of autonomous airport surface movement operations based on multi-agent planning', *European Journal of Transport and Infrastructure Research*, vol. 25, no. 1, pp. 1–23, 1 2025, ISSN: 1567-7141. DOI: 10.59490/ejtir.2025.25.1.7459.

[133] F. Rooseleer and V. Treve, *RECAT-EU: European wake turbulence categorisation and separation minima on approach and departure*, version 1.2, 2018. [Online]. Available: https://www.eurocontrol.int/sites/default/files/2021-07/recat-eu-released-september-2018.pdf (visited on 05/04/2023), Released Issue.

[134] T. T. Walker and N. R. Sturtevant. 'Collision detection for agents in multi-agent pathfinding'. (2019), pre-published.

[135] EUROCONTROL, *Airport capacity imbalance*, Performance Review Commission, 2020. [Online]. Available: https://www.eurocontrol.int/sites/default/files/2020-12/eurocontrol-prc-technical-note-airport-capacity-imbalance-11122020.pdf (visited on 19/05/2023).

[136] J.-M. Alkazzi and K. Okumura, 'A comprehensive review on leveraging machine learning for multi-agent path finding', *IEEE Access*, vol. 12, pp. 57 390–57 409, 2024, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2024.3392305.

[137] M. von der Burg and A. Sharpanskykh, 'Exploring human-automation interactions in next-generation airport surface movement operations: An agent-based modelling approach', in *SESAR Innovation Days 2024*, Rome, Italy, 2024, pp. 1–9. DOI: 10.61009/SID.2024.1.11.

[138] H. Ali, D.-T. Pham, S. Alam *et al.*, 'Human-AI hybrids in safety-critical systems: Concept, definition and perspectives from air traffic management', *Adv. Eng. Informatics*, vol. 65, p. 103 256, 2025, ISSN: 1474-0346. DOI: 10.1016/j.aei.2025.103256.

[139] 'SESAR Joint Undertaking | MOTO - the Embodied Remote Tower'. (2016), [Online]. Available: https://www.sesarju.eu/projects/moto (visited on 25/09/2023).

[140] 'SESAR Joint Undertaking | TaCo - Take Control'. (2017), [Online]. Available: https://www.sesarju.eu/projects/taco (visited on 25/09/2023).

[141] P. C. Roling and H. G. Visser, 'Optimal airport surface traffic planning using mixed-integer linear programming', *International Journal of Aerospace Engineering*, vol. 2008, e732828, 2008, ISSN: 1687-5966. DOI: 10.1155/2008/732828.

[142] M. Weiszer, E. K. Burke and J. Chen, 'Multi-objective routing and scheduling for airport ground movement', *Transp. Res. Part C: Emerging Technologies*, vol. 119, p. 102 734, 2020, ISSN: 0968090X. DOI: 10.1016/j.trc.2020.102734.

[143] M. Zhang, Huang, Liu and Li, 'Multi-objective optimization of aircraft taxiing on the airport surface with consideration to taxiing conflicts and the airport environment', *Sustainability*, vol. 11, p. 6728, 2019. DOI: 10.3390/su11236728.

[144] Y. Liu, M. Hansen, G. Gupta, W. Malik and Y. Jung, 'Predictability impacts of airport surface automation', *Transp. Res. Part C: Emerging Technologies*, vol. 44, pp. 128–145, 2014, ISSN: 0968090X. DOI: 10.1016/j.trc.2014.03.010.

[145] SESAR JU, *Automation in air traffic management: Long-term vision and initial research roadmap*, 2020.

[146] C. Stergianos, J. Atkin, P. Schittekat, T. Nordlander, C. Gerada and H. Morvan, 'The importance of considering pushback time and arrivals when routing departures on the ground at airports', in *Proceedings of ICAOR-16*, vol. 8, 2016, pp. 41–46.

[147] X-Plane, *WorldEditor (WED)*, version WED 2.5, 2024. [Online]. Available: https://developer.x-plane.com/tools/worldeditor/ (visited on 02/08/2024).

[148] J. Chen, M. Weiszer, P. Stewart and M. Shabani, 'Toward a more realistic, cost-effective, and greener ground movement through active routing—part I: Optimal speed profile generation', *IEEE Transactions on Intel. Transp. Sys.*, vol. 17, pp. 1–14, 2015. DOI: 10.1109/TITS.2015.2477350.

[149] I. Bae, J. Moon, J. Jhung *et al.* 'Self-driving like a human driver instead of a robocar: Personalized comfortable driving experience for autonomous vehicles'. (2022), [Online]. Available: http://arxiv.org/abs/2001.03908 (visited on 08/03/2024), pre-published.

[150] K. N. De Winkel, T. Irmak, R. Happee and B. Shyrokau, 'Standards for passenger comfort in automated vehicles: Acceleration and jerk', *Applied Ergonomics*, vol. 106, p. 103 881, 2023, ISSN: 00036870. DOI: 10.1016/j.apergo.2022.103881.

[151] R. G. Sargent, 'Verification and validation of simulation models', in *Proceedings of WSC-10*, Baltimore, MD, USA: IEEE, 2011, pp. 166–183, ISBN: 978-1-4244-9866-6. DOI: 10.1109/WSC.2010.5679166.

[152] M. von der Burg and A. Sharpanskykh, 'Multi-agent planning for autonomous airport surface movement operations', in *SESAR Innovation Days 2023*, 2023, pp. 1–9. DOI: 10.61009/SID.2023.1.27.

[153] Schiphol. 'Standaard pushback per positie', Schiphol. (), [Online]. Available: https://www.schiphol.nl/en/operations/page/sleep-en-pushbackbewegingen/ (visited on 30/08/2022).

[154] NielsB. 'Schiphol (Amsterdam Airport) overview map', Wikimedia Commons. (2007), [Online]. Available: https://commons.wikimedia.org/wiki/File:Schiphol-overview.png (visited on 28/09/2023).

[155] D. Bresser and S. Prent. 'The benefits of sustainable taxiing', Schiphol Innovation. (2020), [Online]. Available: https://www.schiphol.nl/en/innovation/page/the-benefits-of-sustainable-taxiing/ (visited on 05/02/2024).

[156] D. Bresser and S. Prent. 'Is sustainable taxiing possible at a busy Schiphol?', Schiphol Innovation. (2021), [Online]. Available: https://www.schiphol.nl/en/innovation/blog/is-sustainable-taxiing-possible-at-a-busy-schiphol/ (visited on 05/02/2024).

[157] 'Response by schiphol to labour authority decision', Schiphol Newsroom. (2024), [Online]. Available: https://news.schiphol.com/response-by-schiphol-to-labour-authority-decision/ (visited on 06/02/2024).

[158] H. Khadilkar and H. Balakrishnan, 'Estimation of aircraft taxi fuel burn using flight data recorder archives', *Transp. Res. Part D: Transport and Environment*, vol. 17, no. 7, pp. 532–537, 2012, ISSN: 13619209. DOI: 10.1016/j.trd.2012.06.005.

[159] J. Overton, *The growth in greenhouse gas emissions from commercial aviation*, Environmental and Energy Study Institute, 2022.

[160] IATA. 'Jet Fuel Price Monitor', Jet Fuel Price Monitor. (), [Online]. Available: https://www.iata.org/en/publications/economics/fuel-monitor/ (visited on 01/02/2024).

[161] IATA, *Net zero 2050: Operational and infrastructure improvements*, 2022.

[162] B. Ghosh, Y. Lim, T. Tay, S. Zhang and J. Shankar, 'A Physical Aircraft-Following Model for Aerodrome Modelling and Simulation', in *Proceedings of ICRAT-24*, 2024.

[163] G. Weiss, V. Conitzer, M. Wooldridge *et al.*, *Multiagent Systems*. Cambridge, UNITED STATES: MIT Press, 2013, ISBN: 978-0-262-31355-1. [Online]. Available: http://ebookcentral.proquest.com/lib/delft/detail.action?docID=3339590 (visited on 29/12/2021).

[164] M. Schultz, J. Rosenow and X. Olive, 'Data-driven airport management enabled by operational milestones derived from ADS-B messages', *Journal of Air Transport Management*, vol. 99, p. 102 164, 2022, ISSN: 09696997. DOI: 10.1016/j.jairtraman.2021.102164.

[165] I. Navarro, P. Ortega-Kral, J. Patrikar *et al.*, 'Amelia: A large model and dataset for airport surface movement forecasting', in *AIAA AVIATION FORUM AND ASCEND 2024*, 2024, pp. 1–24. DOI: 10.2514/6.2024-4251.

[166] J. Ma, J. Zhou, M. Liang and D. Delahaye, 'Data-driven trajectory-based analysis and optimization of airport surface movement', *Transp. Res. Part C: Emerging Technologies*, vol. 145, p. 103 902, 2022, ISSN: 0968090X. DOI: 10.1016/j.trc.2022.103902.

[167] J. Bao, J. Kang, J. Zhang, Z. Zhang and J. Han, 'A dynamic control method for airport ground movement optimization considering adaptive traffic situation and data-driven conflict priority', *Journal of Air Transport Management*, vol. 124, p. 102 753, 2025, ISSN: 09696997. DOI: 10.1016/j.jairtraman.2025.102753.

[168] S. Badrinath, H. Balakrishnan, J. Ma and D. Delahaye, 'A comparative analysis of departure metering at Paris (CDG) and Charlotte (CLT) airports', in *Proceedings of ATM2019*, 2019. DOI: https://hdl.handle.net/1721.1/145251.

[169] U. Dorndorf, F. Jaehn and E. Pesch, 'Flight gate scheduling with respect to a reference schedule', *Annals of Operations Research*, vol. 194, no. 1, pp. 177–187, 2012, ISSN: 1572-9338. DOI: 10.1007/s10479-010-0809-8.

[170] U. M. Neuman and J. A. D. Atkin, 'Airport Gate Assignment Considering Ground Movement', in *Computational Logistics*, D. Pacino, S. Voß and R. M. Jensen, Eds., Berlin, Heidelberg: Springer, 2013, pp. 184–198, ISBN: 978-3-642-41019-2. DOI: 10.1007/978-3-642-41019-2_14.

[171] J. Du, M. Hu, J. Yin and W. Zhang, 'Multi-Objective Gate Allocation Problem Based on Multi-Commodity Network Flow Model', *Applied Sciences*, vol. 12, no. 19, p. 9849, 2022, ISSN: 2076-3417. DOI: 10.3390/app12199849.

[172] Abraham. Savitzky and M. J. E. Golay, 'Smoothing and Differentiation of Data by Simplified Least Squares Procedures.', *Analytical Chemistry*, vol. 36, no. 8, pp. 1627–1639, 1964, ISSN: 0003-2700. DOI: 10.1021/ac60214a047.

[173] P. A. Gorry, 'General least-squares smoothing and differentiation by the convolution (Savitzky-Golay) method', *Analytical Chemistry*, vol. 62, no. 6, pp. 570–573, 1990, ISSN: 0003-2700. DOI: 10.1021/ac00205a007.

[174] P. E. Hart, N. J. Nilsson and B. Raphael, 'A formal basis for the heuristic determination of minimum cost paths', *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968, ISSN: 2168-2887. DOI: 10.1109/TSSC.1968.300136.

[175] M. Schlosser, H. Braßel and H. Fricke, 'Analysis of Aircraft Ground Trajectories: Map-Matching with Open Source Data for Modeling Safety-Driven Applications', in *Proceedings of ICRAT-24*, 2024, pp. 1–8.

[176] ICAO, *Doc 4444: Air traffic management*, version 15th, 2007.

[177] A. Okina, F. Dijkstra and N. Jester, 'RECAT-EU for departures at Schiphol', FWY-2020-01, 2020.

[178] Schiphol, *RASAS: Regulation aircraft stand allocation Schiphol*, 2024.

[179] Airbus, *A320 aircraft characteristics - airport and maintenance planning*, 2024.

[180] Boeing, *787 airplane characteristics for airport planning*, 2024.

[181] L. Blain. 'Autonomous Airbus aces autopilot taxi, takeoff and landing tests', New Atlas. (2020), [Online]. Available: https://newatlas.com/aircraft/airbus-attol-autonomous-airliner/ (visited on 07/05/2024).

[182] K. Christoffersen and D. D. Woods, 'How to make automated systems team players', in *Advances in Human Performance and Cognitive Engineering Research*, vol. 2, Emerald Group Publishing Limited, 2002, ISBN: 978-0-7623-0864-4. DOI: 10.1016/S1479-3601(02)02003-9.

[183] G. Klien, D. Woods, J. Bradshaw, R. Hoffman and P. Feltovich, 'Ten challenges for making automation a "team player" in joint human-agent activity', *IEEE Intelligent Systems*, vol. 19, no. 6, pp. 91–95, 2004, ISSN: 1941-1294. DOI: 10.1109/MIS.2004.74.

[184] M. R. Endsley and E. O. Kiris, 'The Out-of-the-Loop Performance Problem and Level of Control in Automation', *Human Factors*, vol. 37, no. 2, pp. 381–394, 1995, ISSN: 0018-7208. DOI: 10.1518/001872095779064555.

[185] M. R. Endsley, 'Toward a Theory of Situation Awareness in Dynamic Systems', *Human Factors*, vol. 37, no. 1, pp. 32–64, 1995, ISSN: 0018-7208. DOI: 10.1518/001872095779049543.

[186] C. Westin, C. Borst and B. Hilburn, 'Strategic Conformance: Overcoming Acceptance Issues of Decision Aiding Automation?', *IEEE Transactions on Human-Machine Systems*, vol. 46, no. 1, pp. 41–52, 2016, ISSN: 2168-2291, 2168-2305. DOI: 10.1109/THMS.2015.2482480.

[187] M. R. Endsley, 'Supporting Human-AI Teams:Transparency, explainability, and situation awareness', *Computers in Human Behavior*, vol. 140, p. 107 574, 2023, ISSN: 0747-5632. DOI: 10.1016/j.chb.2022.107574.

[188] K. van de Merwe, S. Mallam and S. Nazir, 'Agent Transparency, Situation Awareness, Mental Workload, and Operator Performance: A Systematic Literature Review', *Human Factors*, vol. 66, no. 1, pp. 180–208, 2024, ISSN: 0018-7208. DOI: 10.1177/00187208221077804.

[189] L. Peters. 'Amsterdam Schiphol develops autonomous GPU robot', AGN. (2025), [Online]. Available: https://aerospaceglobalnews.com/news/amsterdam-schiphol-airport-robotic-arc-gpu-connection/ (visited on 27/11/2025).

[190] SESAR JU, *SESAR Solutions Catalogue*, 4th ed. LU: Publications Office, 2021. [Online]. Available: https://data.europa.eu/doi/10.2829/998701 (visited on 08/12/2025).

[191] T. Zhang, M. Weiszer and J. Chen, 'The feasibility of Follow-the-Greens for 4-dimensional trajectory based airport ground movements', *Transp. Res. Part C: Emerging Technologies*, vol. 116, p. 102 632, 2020, ISSN: 0968090X. DOI: 10.1016/j.trc.2020.102632.

[192] R. Guo, Y. Zhang and Q. Wang, 'Comparison of emerging ground propulsion systems for electrified aircraft taxi operations', *Transp. Res. Part C: Emerging Technologies*, vol. 44, pp. 98–109, 2014, ISSN: 0968090X. DOI: 10.1016/j.trc.2014.03.006.

[193] S. T. Taskforce, *Sustainable taxi operations: Concept of operations & industry guidance*, EUROCONTROL, 2024.

[194] B. von den Hoff, 'Noise emissions and annoyance of sustainable aviation systems: Identifying noise sources and validating noise prediction models', Ph.D. dissertation, TU Delft, 2025, ISBN: 9789463848008. DOI: 10.4233/uuid:e97e1d99-11f0-46f3-9be9-2f8eeddde3bc.

[195] IFATCA, *Sustainable ground movement and its operational impact*, 2025. [Online]. Available: https://ifatca.org/wp-content/uploads/WP-2025-94-1.pdf (visited on 08/12/2025).

[196] A. Srivastava, 'Improving departure taxi time predictions using ASDE-X surveillance data', in *2011 IEEE/AIAA 30th Digital Avionics Systems Conference*, 2011, 2B5-1-2B5–14. DOI: 10.1109/DASC.2011.6095989.

[197] ASTAIR, *D5.2 exploratory research report*, 2025.

[198] J. Soomers, 'A delay analysis in airport engine-off towing operations using hierarchical multi-agent path planning', M.S. thesis, TU Delft, 2022.

[199] J. Kamphof, 'Design and analysis of tug-enabled engine-off taxiing operation using hierarchical multi-agent planning', M.S. thesis, TU Delft, 2022.

[200] S. Poelstra, 'Evaluation of the Impact of Large- Scale Taxiway Maintenance on the Task Load of Ground Controllers', M.S. thesis, TU Delft, 2025. [Online]. Available: https://resolver.tudelft.nl/uuid:8d608b28-e08e-4de8-893a-48cf7678b3a3.

[201] T. van Rens, 'Integrated Task Assignment and Path Planning for Electric Towing Vehicle Operations at a Major Hub Airport', M.S. thesis, TU Delft, 2025. [Online]. Available: https://resolver.tudelft.nl/uuid:095e77c5-2924-477b-ac6f-3926afc4ce78.

[202] F. Klügl, C. Oechslein, F. Puppe and A. Dornhaus, 'Multi-agent modelling in comparison to standard modelling', p. 6, 2002.

Curriculum Vitæ



Malte Frederic von der Burg was born on 17 December 1991 in Münster, Germany. He started a Bachelor in Mechanical Engineering at the Karlsruhe Institute for Technology (KIT) in Germany. He completed his BSc by conducting research on the failure analysis of a rotor blade spar from wind turbines at the German Aerospace Centre (DLR) in Braunschweig, Germany. After graduation, he did a six-month internship in the department for technology development of fibre-reinforced plastics at AUDI AG in Neckarsulm, Germany.

Malte returned to KIT for his Master in Mechanical Engineering, with a focus on lightweight construction and fluid mechanics. In 2017, he participated in a three-week measurement campaign for turbulence research in Grenoble, France, in which a multi-camera Particle Image Velocimetry (PIV) setup was placed in a large rotating tank filled with over 130 t of water. In his subsequent Master thesis at the Institute of Fluid Mechanics (ISTM) at KIT, he characterised the boundary layer instabilities forming at the curved tank walls, for which he had to evaluate the 10 TB of captured image data. He finished his MSc in Mechanical Engineering cum laude in September 2017.

After a three-months travel through South America, Malte started working as test process developer for the series production of electrical drives at Robert Bosch GmbH in Bühl, Germany. During his three years in that position, he developed two new test processes and completed six test bench projects. For this purpose, he developed multiple algorithms in Python to analyse and visualise the data acquired from optical micrometres, LVDT sensors, and thermographic cameras. During a Lean management initiative, he represented the team's views and assessed areas of improvement.

With the desire to return to academia, move to the field of aviation, and live in the Netherlands, Malte started as PhD candidate in Operations and Environment (O&E) at the Faculty of Aerospace Engineering (AE) at Delft University of Technology in March 2021. His research was funded by the European Union's SESAR JU project 'AEON – Advanced Engine Off Navigation'. The thesis focused on designing multi-agent system models and path planning algorithms for more sustainable airport surface movement operations. Aside from his research, he was part of the AE-PhD council, initiating the much debated Triangle Tuning Program at the faculty.

List of Publications

7. M. von der Burg and A. Sharpanskykh, 'Improving the Realism of Multi-Agent Simulation Models for Airport Surface Movement Operations: Trajectory-based Calibration, Towing Operations, and Operational Variations', 2026. DOI: 10.2139/ssrn.6712278
(preprint, submitted to journal)

6. M. von der Burg, J. Kamphof, J. Soomers and A. Sharpanskykh, 'Towards next-generation airport surface movement operations using hierarchical-distributed multi-agent planning', *Transportation Research Part C: Emerging Technologies*, vol. 186, p. 105 583, 2026, ISSN: 0968-090X. DOI: 10.1016/j.trc.2026.105583

5. M. von der Burg and A. Sharpanskykh, 'Modelling and analysis of autonomous airport surface movement operations based on multi-agent planning', *European Journal of Transport and Infrastructure Research*, vol. 25, no. 1, pp. 1–23, 1 2025, ISSN: 1567-7141. DOI: 10.59490/ejtir.2025.25.1.7459

4. M. von der Burg and A. Sharpanskykh, 'Exploring human-automation interactions in next-generation airport surface movement operations: An agent-based modelling approach', in *SESAR Innovation Days 2024*, Rome, Italy, 2024, pp. 1–9. DOI: 10.61009/SID.2024.1.11

3. M. von der Burg and A. Sharpanskykh, 'Studying the operational consequences of automated engine-off taxiing using multi-agent planning: Explorative case study at Amsterdam Airport Schiphol', in *Proceedings of ICRAT-24*, Singapore, 2024, pp. 1–8

2. M. von der Burg and A. Sharpanskykh, 'Multi-agent planning for autonomous airport surface movement operations', in *SESAR Innovation Days 2023*, 2023, pp. 1–9. DOI: 10.61009/SID.2023.1.27

1. M. von der Burg, A. Sharpanskykh and J. Kamphof, 'D2.1 multi-agent system for routing', AEON, project deliverable D2.1, 2022



Coordinating the movements of aircraft along the surface of busy airports is a complex task that requires multiple stakeholders to work together to ensure safe and efficient operations. In response to the **interrelated challenges** of ever-increasing air traffic demand, emission reduction, and sustaining safety levels, technological advances are emerging which, in turn, necessitate **novel concepts of operations**. A range of such concepts, differing in technological features, level of automation, and degree of control centralisation, may offer promising ways forward. However, before these innovative concepts are matured to be deployable at real-world airports, extensive evaluation based on computational modelling is necessary.

To enable the detailed modelling and analysis of the large variety of concepts for next-generation **airport surface movement operations**, this thesis proposes models, methods, and tools based on the **multi-agent systems** paradigm. In algorithmic and operational studies, these are evaluated on both a synthetic airport layout and the **real-world layout** of Amsterdam Airport Schiphol. The results indicate that calibrated automation, refined procedures, and sustainable taxiing concepts can yield meaningful efficiency and environmental gains without compromising safety or system stability. As a step towards real-world deployment, selected **interaction use cases** are explored to help pave the way for future human–autonomy teaming.