Multi-Agent Planning for Autonomous Airport Surface Movement Operations

Explorative Case Study at Amsterdam Airport Schiphol

Malte von der Burg & Alexei Sharpanskykh
Air Transport Operations, Faculty of Aerospace Engineering
Delft University of Technology, The Netherlands
M.F.vonderBurg@tudelft.nl, O.A.Sharpanskykh@tudelft.nl

***Abstract*—Both EASA and SESAR JU define a vision and roadmap towards an autonomous air traffic management system. Furthermore, past and ongoing SESAR JU projects investigate how to increase the efficiency and predictability of current operations by means of automation. In this paper, we explore the operational implications that may result from fully-automated airport surface movement operations. In our model, a hierarchical multi-agent system coordinates and controls all movements on the airport surface. It comprises the Airport Operations Agent to handle the flight schedule and runway configuration, the Routing Agent to compute conflict-free trajectories, and the Guidance Agents to instruct and monitor the Aircraft Agents while these execute the planned routes. To compute conflict-free trajectories for all agents, we tailored state-of-the-art multi-agent motion planning algorithms to the requirements of taxiing operations: the two-level routing algorithm combines Priority-Based Search (PBS) with Safe Interval Path Planning (SIPP). It accounts for the different taxiing processes such as pushback, engine-start, or wake turbulence separation for takeoffs by defining an activity sequence for each agent. Furthermore, we include the kinematics and different sizes of the aircraft as well as a minimal safety distance between them. 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 autonomous taxiing system with respect to the historic operations. For the considered simulation conditions, we show that the MAS yields 15% lower taxi times for both arriving and departing flights, discuss how reliable these results are, and point out directions for future work.**

***Keywords*—multi-agent system; multi-agent motion planning; autonomous airport operations; airport surface movement operations; automation; air traffic control**

I. Introduction

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 [2]. 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 [3]. Moreover, when Air Traffic Control Officers (ATCOs) have to handle more and potentially less predictable traffic, their workload is amplified [4]. 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 [5], [6]. Other projects examined more specifically how to reduce emissions by integrating engine-off taxiing techniques [7], or how to enable human-automation teamwork in the operations through higher levels of automation [8]. 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 [9]. Moreover, towards 2050 and beyond, both EASA and SESAR JU define a vision and roadmap to eventually reach autonomous air traffic management, i.e. level 3 in EASA's AI roadmap [10], or level 4-5 in the roadmap of SESAR JU [11]. However, the remaining 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 paper, we explore which operational consequences may result from autonomous surface movement operations at large airports: we model such operations using a multi-agent system (MAS) that plans conflict-free routes for all aircraft on the ground and controls their execution. As pushback and engine-start operations considerably effect delays in planning systems [12], we explicitly model these processes in the MAS.

Multi-agent simulations allow for inherent modularity, flexibility, and expressiveness. Both heterogeneous agent properties and behaviour as well as randomness can be integrated into an agent-based model [13]. Thus, a MAS is well suited to model an autonomous control system for taxiing operations.

In general, to plan conflict-free routes for a set of agents, many different multi-agent path finding algorithms have been developed that model agents as a point [14]. However, since aircraft are large vehicles that need time to speed up and slow

SESAR Innovation Days
27-30 November 2023
Seville

down, the domain of multi-agent motion planning (MAMP) offers more suitable algorithmic concepts to include agent shapes and kinematics [15]. We combine and extend such state-of-the-art MAMP algorithms to form a routing algorithm that is tailored to taxiing operations. Section II-A summarises how it computes conflict-free trajectories for all aircraft.

With an implementation of the MAS in Python, we simulate the autonomous taxiing operations using the flight schedule of two of the busiest days at Amsterdam Airport Schiphol to date. Section III outlines the experimental setup. We then analyse and discuss key performance indicators in relation to their historic counterparts in Section IV, and end with concluding remarks in Section V.

II. Multi-Agent System Model

The multi-agent system (MAS) for autonomous aircraft taxiing operations comprises a distributed-hierarchical structure of both centralized and distributed agents, which is illustrated in Fig. 1. The centralized Airport Operations Agent defines and updates the flight schedule and runway configuration, the centralized 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 autonomous operations considered in this paper, 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.

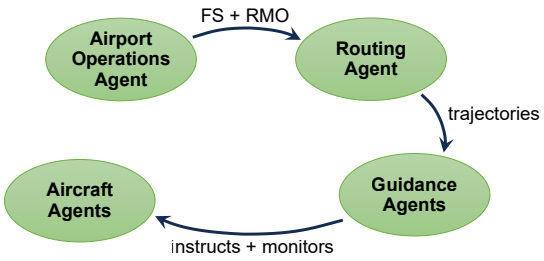

Figure 1. Overview of multi-agent system for autonomous aircraft taxiing operations

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 paper, is shown in Fig. 2. Vertices denote aircraft stands (green), taxiway intersections (black), 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 obtained from the actual locations of these taxiways at Schiphol.

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

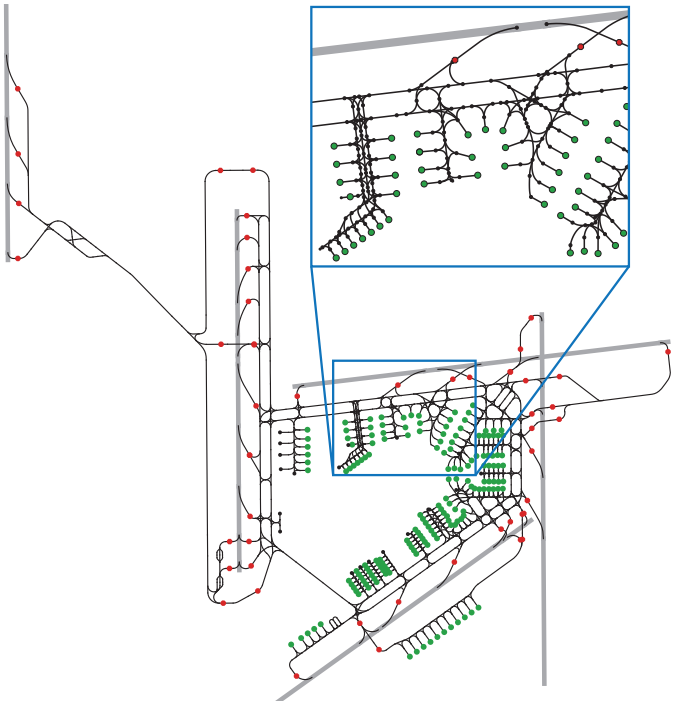

Figure 2. Graph of layout of Amsterdam Airport Schiphol with edges for runways (grey) or taxiways (black), and vertices for taxiway intersections (black), stopbars (red), or gates (green)

mechanism is also applicable for taxiway segments that are temporarily unavailable. Furthermore, the Airport Operations Agent schedules all flights, and updates them whenever new predictions of the underlying A-CDM milestones are available.

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_{plng}. It re-computes the routing plans when it receives updates from the Airport Operations Agent, or latest after the re-planning period h_{plng} 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) [16] with an augmented version of the Safe Interval Path Planning (SIPP) algorithm [17]. This routing algorithm is presented in Section II-A.

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 minimize 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 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:

- Go-to 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 go-to 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-truck 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 Fig. 3.

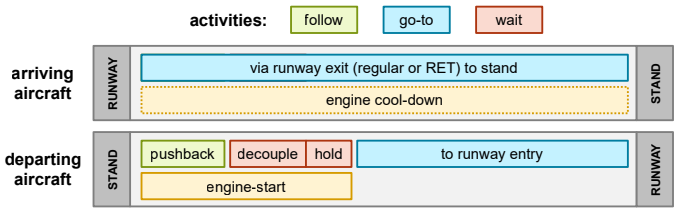

Figure 3. Activity sequence for arriving and departing aircraft. While engine-start (orange box) of departing aircraft is accounted for, engine cool-down (dotted orange box) is neglected.

In the sequence, the warmup and cooldown of the engines represent special cases. The routing algorithm takes the warmup-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-truck, additional waiting in form of holding is added to the route. We do not model engine cooldown, as it does not have an influence on the routing regarding the kinematics, since the engines are switched off after standstill at the gate.

To account for the different sizes of aircraft, all flights are categorized as one of the 6 aircraft types from the ICAO aerodrome reference codes [18]. They are assumed to have a circular shape with a pre-defined radius according to the type. Table II in Section III 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 the same runway must have a minimal separation to mitigate the wake turbulence of the preceding aircraft. We use the time-based separation minima from RECAT-EU for that [19].

A. 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, we extended the Safe Interval Path Planning (SIPP) algorithm [17], and adapted the Priority-Based Search (PBS) algorithm [16] to serve as high-level solver.

PBS constructs a priority order between agents to deconflict their space-time trajectories. In its priority tree, each parent-node has up to two child-nodes. Thus, a priority-relation between a conflicting pair of agents is established. In each child-node, one additional priority-pair is added with which one of the two agents that were previously in conflict must give way to the other agent along its entire route. Then, PBS checks the child node that has the lowest sum-of-cost of all agent trajectories for conflicts between those agents that do not yet form a priority-relation with each other. We define the cost of a trajectory as sum of the taxiing duration and travelled distance. Once a child-node is expanded without any collisions, PBS returns the resulting conflict-free trajectories.

In the low-level search, the route of a deprioritized 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, velocity 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 velocity in the current state. A motion that is part of the follow-activity for push-back is for example constrained by a lower maximum speed than regular taxiing in a go-to activity. In addition, vehicles that have maximum velocity 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 velocity, braking distance, and reservations or velocity restrictions within the braking distance.

B. Verification and Validation

Verification and validation of the simulation model were performed in accordance with validation techniques and tests as described by Sargent [20]. 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.

III. Experimental Setup

In this section, we present the experimental setup to simulate the flight schedules of two consecutive days of the 17th and 18th July 2019 at Amsterdam Airport Schiphol. To this end, 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 historic Actual Landing Time (ALDT); their velocity at rapid-exit taxiways $v_{RET} = v_{max}$, or at regular exits $v_{exit} = v_{turn}$
- departing aircraft are spawned at their historic 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 [21])
- no departure sequence at runways; any Computed Take-Off Times issued by Eurocontrol (CTOT-slots) are neglected
- potential conflicts at the same stand are neglected between an arriving aircraft and a departing aircraft that is not yet spawned
- 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 standard taxiway 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. CTOT-slots as well as potential stand conflicts are neglected as we did not yet implement the necessary resolution strategies to cope with these special cases.

At Schiphol, two main runway mode of operations (RMOs) exist: **RMO North** (active on 17th July 2019), and **RMO South** (active on 18th July 2019). During each day, different runway combinations are set. These RMO phases are visualized in Fig. 4.

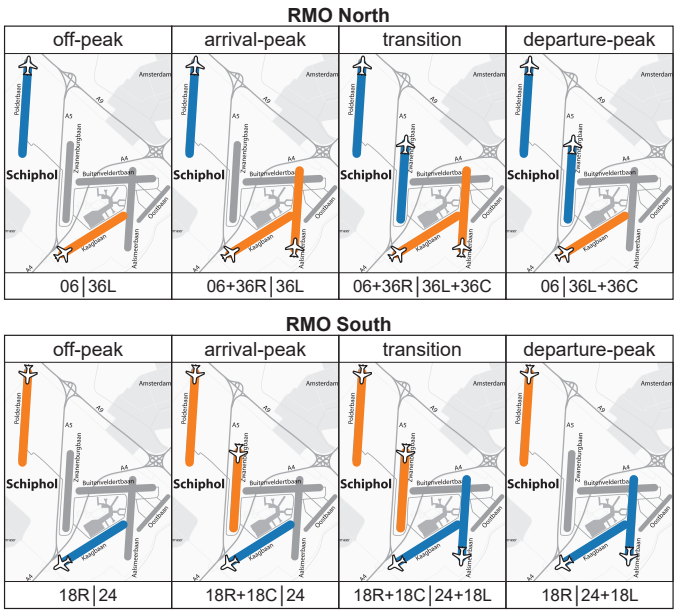

Figure 4. Runway mode of operations (RMO) at Schiphol: different phases of RMO North (top) and RMO South (bottom) with active runways for arrivals (orange) and departures (blue). Background map adapted from [22].

In the simulation, the same runways are activated according to those that were active in the historic 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 I lists the total number of flights, arrivals, departures, the main RMO, and the number of RMO phases for the two simulated days.

TABLE I. Overview of operational data

date	17-07-2019	18-07-2019
flights	1489	1492
arrivals	745	744
departures	744	748
RMO	RMO North	RMO South
RMO phases	19	19

As mentioned above, each aircraft is categorized as one of the six ICAO-types with an associated shape radius and wake turbulence category (WTC). Table II lists these parameters along with the count over the two days. The kinematic properties of the aircraft agents and the main algorithmic parameters are listed in Table III. As minimal safety distance between every pair of agents, we use their average radius in general. However, when an aircraft is trailing another agent, it has to keep a safety distance of at least 3-times the shape radius of the preceding aircraft, which resulted from expert interviews.

TABLE II. ICAO-TYPES: RADIUS AND WAKE TURBULENCE CATEGORY (WTC), AS WELL AS COUNT IN OPERATIONAL DATA

	parameters		count per day	
	shape [m]	WTC	17-07-2019	18-07-2019
ICAO-A	12	CAT-F	0	0
ICAO-B	25	CAT-E	22	20
ICAO-C	40	CAT-D	1195	1198
ICAO-D	54	CAT-C	37	43
ICAO-E	72	CAT-B	213	206
ICAO-F	80	CAT-A	22	25

TABLE III. KINEMATIC PROPERTIES OF AIRCRAFT AGENTS AND ALGORITHM PARAMETERS THAT ARE USED IN THE ROUTING ALGORITHM

parameter	value	unit
maximal velocity v_{max}	15	m/s
maximal turn velocity $v_{turn,max}$	5	m/s
minimal velocity v_{min}	1.5	m/s
acceleration acc	0.25	m/s^2
deceleration dec	-0.75	m/s^2
planning window w_{plng}	30	min
replanning period h_{plng}	15	min

We mainly use key performance indicators (KPIs) based on the taxi time as it is directly linked to the efficiency, predictability, and resulting emissions of airport surface movement operations. Besides the average, we also report the median and interquartile range of the taxi time, since it does not follow a normal distribution. Furthermore, as indication of the runway use, we provide the maximal hourly throughput and occupancy rate of any departure or arrival runway.

IV. SIMULATION: RESULTS

To give an overview of the traffic situation, Fig. 5 shows the hourly count of all flights for both the historic (black dotted line) as well as simulated operations (grey line) over the two days of 17th and 18th 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 figure visualizes 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.

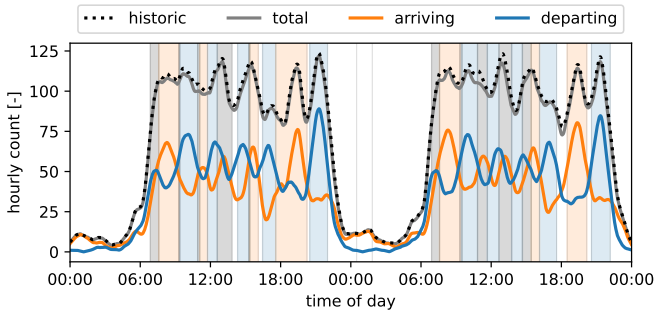

Figure 5. Hourly count of flights over the two days. Shades denote the RMO phase: off-peak (white), arrival-peak (orange), transition (grey), and departure-peak (blue)

Table IV lists the mean, median, and interquartile range (IQR) of the taxi times for inbound and outbound flights. For both days, the simulated operations show a significant decrease for all indicators: the taxi time of each flight is shorter on average, and varies less between flights. Moreover, for any runway in use, we report the maximal throughput and occupancy rate per hour. The occupancy rate is calculated as the relative time that the runway is blocked due to the minimal wake turbulence separation between two consecutive aircraft. For arrivals, the indicators are identical between the historic and simulated operations as we used the actual landing time (ALDT) as spawn-time in the simulation (cp. the assumptions in Section III). For departures, the maximal hourly throughput is similar between historic and simulated operations, while the maximal hourly occupancy rate has increased for the simulated operations.

TABLE IV. COMPARISON OF HISTORIC AND SIMULATED OPERATIONS

date		17-07-2019		18-07-2019	
operations		historic	simulated	historic	simulated
ARR	mean taxi time	05:00	03:34	10:29	08:40
	median taxi time	04:06	03:18	10:46	09:28
	IQR taxi time	03:10	01:53	06:06	05:24
	RWY throughput*	42		38	
	RWY occupancy*	68.7%		69.3%	
DEP	mean taxi time	14:30	12:35	10:21	07:43
	median taxi time	14:32	12:36	10:08	07:24
	IQR taxi time	06:28	05:01	04:16	02:45
	RWY throughput*	45	44	43	42
	RWY occupancy*	74.7%	76.1%	73.5%	76.6%

*: maximal hourly value for any runway

As defined in Section III, we do not use a departure sequence. Thus, the order of aircraft departing from a runway is different between the historic and simulated operations, as exemplary shown in Fig. 6. As emergent property of the MAS, whenever possible, flights are grouped together with minimal separation time between each takeoff, which is illustrated by the red shades in the figure. While the minimal separation of the historic order mostly matches those from RECAT-EU, in some cases, the actual separation was less than the minimum. We have to analyse the actual track data of the historic operations to determine the reason for this discrepancy.

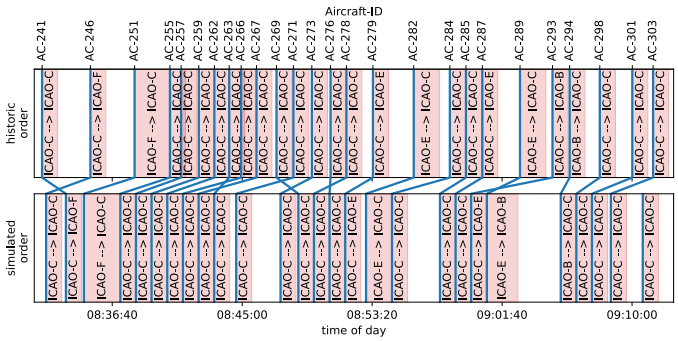

Figure 6. Comparison of exemplary takeoff order at runway 18R/36L between historic (top) and simulated operations (bottom), with actual takeoff time (blue lines) and minimal WTC separation (red shades)

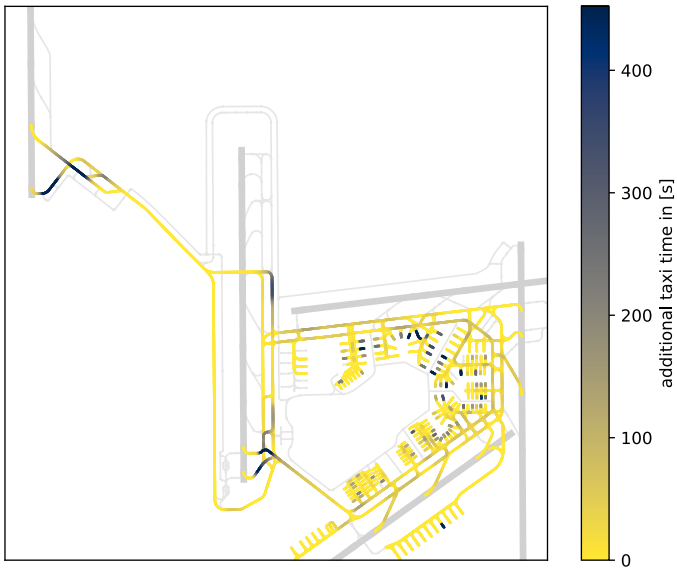

Figure 7. Additional taxi time per layout location on 17-07-2019

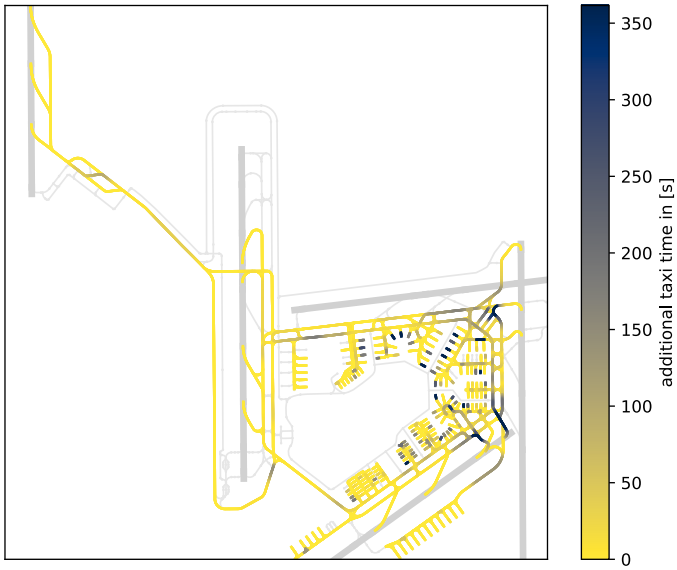

Figure 8. Additional taxi time per layout location on 18-07-2019

A. *Hotspots in the Taxiway Network*

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 minimal velocity of $1.5\,\mathrm{m/s}$, or the agent has to hold at the stand or engine-start location. The latter two create additional taxi time that we sum per $5\,\mathrm{m}$-segment of the graph underlying the airport layout. This yields the locations where most delays across all flights occur, as visualized in Figs. 7 and 8 for the two days, respectively.

For both days, hotspots form in front of the runway stopbars, the bay areas and stands, as well as on some taxiway segments. On the 17th July, one of the latter hotspots is located in front of the crossing of the second departure-runway 18C/36C, which is active only in departure-peaks and transition-phases (cp. Fig. 4). Aircraft slow down in front of the crossing to await 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 velocity until the minimal 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.

B. *Predictability of Taxi Time*

Accurately estimating the A-CDM milestones helps to synchronize the many different processes that take place at an airport. To this end, we assess the predictability of the taxi times within the simulated operations. Recall that per planning round, the MAS plans the routes for all aircraft that are scheduled to start within the $30\,\mathrm{min}$ planning window and those that are already taxiing. Conflicts with other agents that occur beyond the planning window are ignored. This yields a predicted taxi time that is updated in the subsequent planning round, which takes place latest after the $15\,\mathrm{min}$ replanning period has passed. Fig. 9 shows the accuracy of these predictions with respect to the remaining actual taxi time.

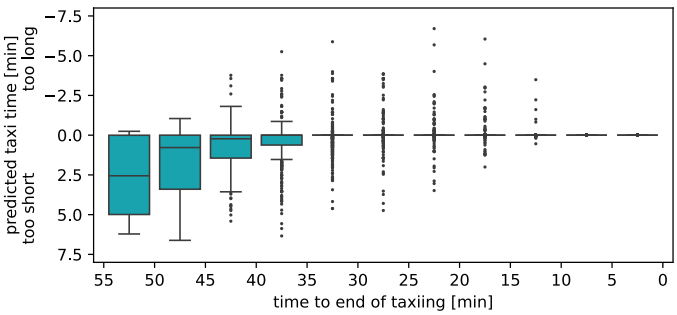

Figure 9. Predictability of taxi time

The first predictions (between $55\,\mathrm{min}$ and $35\,\mathrm{min}$ to the end of the respective actual taxi time) mostly underestimate the necessary taxiing duration. However, the following predictions are more accurate: from $30\,\mathrm{min}$ prior to reaching the taxi destination onward, the difference between predicted and

actual taxi time is negligible for most flights, with some outliers ranging between $\pm 5\,\mathrm{min}$. Below $10\,\mathrm{min}$, the prediction matches the actually remaining taxi time for all flights. Note that this may change when deviations to the planning arise during execution, which we did not model in this work.

C. Discussion on Accuracy of Taxi Time

In the following, we analyse the taxi time in more detail to better understand the differences between the historic and simulated operations. In Fig. 10, the taxi times for arrivals and departures are plotted per runway strip as box-and-whisker plots that show the median, first and third quartiles as box, as well as the outliers as whiskers and points. 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 engine-start time of $6\,\mathrm{min}$ for large aircraft (ICAO-D to ICAO-F) in comparison to $3\,\mathrm{min}$ for small aircraft, the taxi times of departing aircraft vary more than those of arriving aircraft in the simulated operations.

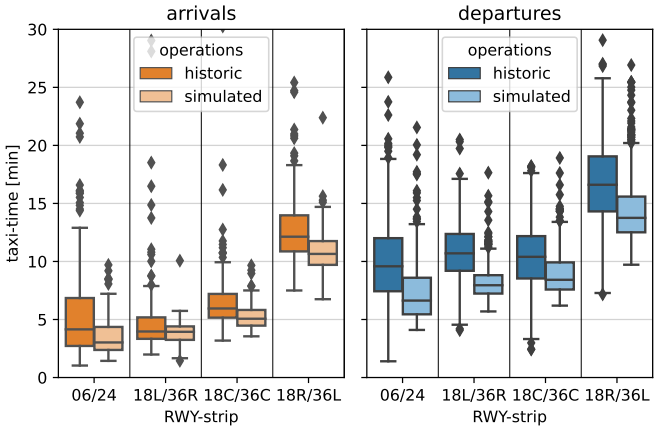

Figure 10. Box-and-whisker plot of historic and simulated taxi times for arrivals and departures per runway-strip

The above analysis of the taxi time points towards a significant increase in efficiency, decrease of emissions, and higher predictability of taxi times that could be achieved through autonomous taxiing operations. However, the above results may be influenced by factors that we did not model accurately enough. We discuss these in the following.

As mentioned in the list of assumptions in Section III, we neglected CTOT-slots as well as gate conflicts for aircraft that did not yet spawn in the simulation. Potential violations concern $3.9\,\%$ of all flights. However, when removing these from the plotted data of Fig. 10, the boxes that denote the central $50\,\%$ of each distribution do not change significantly. With this corrected dataset, the average taxi time reduces by approximately $2\,\mathrm{min}$ or $15\,\%$.

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 use the time point of entering the runway as the end of taxiing. This could introduce an offset to the taxi time of departing aircraft in comparison to the historic data. On the other hand, the accuracy of the historic A-CDM milestones is unknown. Furthermore, we do not know the actual engine-start duration of departing flights, and have used $15\,\mathrm{m/s}$ as maximally allowed velocity on straight taxiway segments as well as $5\,\mathrm{m/s}$ for curves with a radius below $100\,\mathrm{m}$. Thus, the taxi times of both the historic and simulated operations may be affected. For future work, we aim to get access to the historic trajectories of flights to be able to analyse these points.

D. 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 $1.80\,\mathrm{GHz}$ Intel Core i7-10610U CPU and $16\,\mathrm{GB}$ RAM. In the simulation, two parameters influence the duration of path planning the most: the number of vehicles to be routed concurrently, and the time window in which conflicts have to be resolved. However, in the presented work, we used a fixed planning window $w_{plng} = 30\,\mathrm{min}$, and replanning period $h_{plng} = 15\,\mathrm{min}$. In Fig. 11, the runtime of each planning round is plotted over the duration of the two days (top part) and the number of agents that had to be routed (bottom part). The exponential nature of the runtime in relation to the number of agents is clearly visible, while the changing RMO phases have a subordinate effect. Nonetheless, in most of the 206 planning rounds, the routing algorithm found a conflict-free solution in under $2\,\mathrm{min}$.

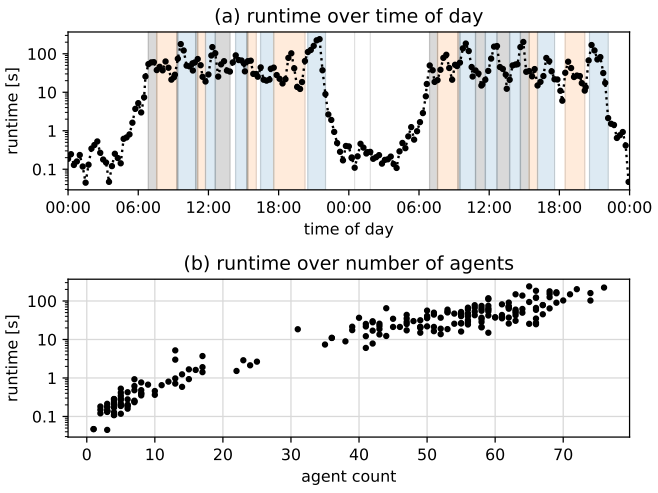

Figure 11. Runtime of routing algorithm. (a) shows the runtime of each planning round over the duration of the two days; the shades denote the RMO phase (off-peak: white, arrival-peak: orange, transition: grey, departure-peak: blue). (b) shows the runtimes per agent count.

E. Future Work

In this study, we considered a futuristic scenario of autonomous operations. However, before real-world airport sur-

face 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-automation teamwork that keeps controllers in the loop is needed.

As continuation of the research presented in this paper, we first plan to analyse how the engine-off taxiing techniques that were explored in AEON [7] affect the key performance indicators of autonomous taxiing operations. Furthermore, we plan to increase the model accuracy by including additional mechanisms to handle for instance the CTOT-slots and potential stand conflicts.

Besides this, aircraft 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, 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.

V. Conclusion

In this paper, we presented how autonomous taxiing operations at large airports could be facilitated by using a hierarchical multi-agent system (MAS) that coordinates and controls all movements on the airport surface. As centralized 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. Furthermore, pushback operations including engine-start as well as the adherence to a minimal safety distance during taxiing and minimal wake turbulence separation during takeoff were explicitly included in the model. The routing algorithm was found to be well suited for planning conflict-free trajectories of all aircraft.

We evaluated 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 autonomous operations controlled by the MAS reduced the average taxi time per flight by around $2\,\mathrm{min}$, or $15\,\%$. Moreover, within $30\,\mathrm{min}$ to the end of taxiing of most flights, the MAS was able to accurately predict the remaining taxi time. This underlines the potential of the MAS as control model for autonomous airport surface movement operations that are more efficient, predictable, and thus reduce the associated emissions.

Acknowledgment

This work has received funding from the SESAR Joint Undertaking under grant agreement No 892869 under European Union's Horizon 2020 research and innovation programme.

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] Eurocontrol, *European aviation in 2040 - challenges of growth*, Eurocontrol Statistics and Forecast Service, 2018.

[3] Eurocontrol, *Data snapshot #22 on lower summer taxi-out times*, 2021.

[4] 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*, ACM, 2017.

[5] "SESAR Joint Undertaking — MOTO - the Embodied Remote Tower." (2016), [Online]. Available: https://www.sesarju.eu/projects/moto (visited on 09/25/2023).

[6] "SESAR Joint Undertaking — TaCo - Take Control." (2017), [Online]. Available: https://www.sesarju.eu/projects/taco (visited on 09/25/2023).

[7] "AEON - Advanced Engine-Off Taxiing Operations," AEON. (2021), [Online]. Available: https://www.aeon-project.eu/ (visited on 09/25/2023).

[8] "SESAR Joint Undertaking — ASTAIR - Auto-Steer Taxi at Airport." (2023), [Online]. Available: https://www.sesarju.eu/projects/astair (visited on 09/25/2023).

[9] R. Lane, S. Dubuisson, M. Ellejmi, and E. Cerasi, *EUROCONTROL specification for A-SMGCS services*, Eurocontrol Specification, version 2.0, 2020, Released Issue.

[10] *AI roadmap 2.0: Human-centric approach to AI in aviation*, EASA, 2023.

[11] *Automation in air traffic management: Long-term vision and initial research roadmap*, SESAR JU, 2020.

[12] 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*, 2016.

[13] D. Helbing and S. Balietti, "How to do agent-based simulations in the future: From modeling social mechanisms to emergent phenomena and interactive systems design," 2015.

[14] R. Stern and N. R. Sturtevant, "Multi-agent pathfinding: Definitions, variants, and benchmarks," in *Proceedings of SoCS-19*, 2019.

[15] L. Cohen, T. Uras, T. K. S. Kumar, and S. Koenig, "Optimal and bounded-suboptimal multi-agent motion planning," in *Proceedings of SoCS-19*, 2019.

[16] 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*, 2019.

[17] M. Phillips and M. Likhachev, "SIPP: Safe Interval Path Planning for dynamic environments," in *Proceedings of ICRA-11*, 2011.

[18] ICAO, "Aerodrome design and operations," in *Annex 14 to the Convention on International Civil Aviation: Volume I*, ser. Aerodromes, 7th ed., ICAO, 2016.

[19] F. Rooseleer and V. Treve, *RECAT-EU: European wake turbulence categorisation and separation minima on approach and departure*, version 1.2, 2018, Released Issue.

[20] R. G. Sargent, "Verification and validation of simulation models," in *Proceedings of WSC-10*, IEEE, 2011.

[21] "Schiphol - standaard pushback per positie," Schiphol. (), [Online]. Available: https://www.schiphol.nl/en/operations/page/sleep-en-pushbackbewegingen/ (visited on 08/30/2022).

[22] NielsB. "Schiphol (Amsterdam Airport) overview map," Wikimedia Commons. (2007), [Online]. Available: https://commons.wikimedia.org/wiki/File:Schiphol-overview.png (visited on 09/28/2023).